A New Security Notion for PKC in the Standard Model: Weaker, Simpler, and Still Realizing Secure Channels

Wasilij Beskorovajnov[1], Roland Gröll[1], Jörn Müller-Quade[1,2,3], Astrid Ottenhues[2,3], and Rebecca Schwerdt[2,3]

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

Abstract. Encryption satisfying CCA2 security is commonly known to be unnecessarily strong for realizing secure channels. Moreover, CCA2 constructions in the standard model are far from being competitive practical alternatives to constructions via random oracle. A promising research area to alleviate this problem are weaker security notions—like IND-RCCA secure encryption or IND-atag-wCCA secure tag-based encryption—which are still able to facilitate secure message transfer (SMT) via authenticated channels.

In this paper we introduce the concept of sender-binding encryption (SBE), unifying prior approaches of SMT construction in the universal composability (UC) model. We furthermore develop the corresponding non-trivial security notion of IND-SB-CPA and formally prove that it suffices for realizing SMT in conjunction with authenticated channels. Our notion is the weakest so far in the sense that it generically implies the weakest prior notions—RCCA and atag-wCCA—without additional assumptions, while the reverse is not true. A direct consequence is that IND-stag-wCCA, which is strictly weaker than IND-atag-wCCA but stronger than our IND-SB-CPA, can be used to construct a secure channel. Finally, we give an efficient IND-SB-CPA secure construction in the standard model from IND-CPA secure double receiver encryption (DRE) based on McEliece. This shows that IND-SB-CPA security yields simpler and more efficient constructions in the standard model than the weakest prior notions, i.e., IND-atag-wCCA and IND-stag-wCCA.

Keywords: Secure Message Transfer · Authenticated Channel · Tag-based Encryption · IND-CPA · IND-CCA2 · CCA2 Relaxations · Universal Composability · McEliece.

1 Introduction

The construction of secure channels is one of the main goals of cryptography. Among the milestones that have been reached to this end are public-key cryptosystems by Diffie and Hellman [24], semantic security by Goldwasser and Micali [28] (today referred to as chosen plaintext attack (CPA)), and the stronger adaptive chosen ciphertext attack (CCA2) by Rackoff and Simon [42].

Nowadays, CCA2 secure public key encryption (PKE) is a cornerstone of many protocols realizing secure channels for our daily life applications. One of the most typical applications is the encryption of e-mails. This is usually realized by implementations of either the S/MIME [45] or OpenPGP [11] standard. Both standards utilize a public key infrastructure (PKI) and digital signatures to realize authenticated channels. Hence we see that widespread applications of secure message transfer (SMT) integrally use authenticated channels and a PKI in addition to encryption. secure message transfer (SMT) is an abstraction of authenticated and encrypted communication in the universal composability (UC) model. How secure message transfer (SMT) can be utilized in practical real world scenarios can be seen for example in [43].

It is widely known that CCA2 is unnecessarily strong to construct SMT when authenticated channels are already present [16]. In addition many concrete CCA2 constructions either lack efficiency to be considered practical constructions or were only proven secure within the random oracle model (ROM), which has inherent problems, e.g., that some constructions which can be proven secure in the ROM are insecure with any implementation of the random oracle [14]. We would like to point out that we do not question the usefulness of the ROM despite its shortcomings. However, we consider the exploration of alternatives just as important and therefore focus on constructions proven secure in the standard model in this work. Hence the following question arises:

> *What is the weakest security definition in order to establish a secure channel in the standard model if we assume existing authenticated channels?*

In an attempt to answer this question we find a non-trivial relaxation of the weakest prior notions of replayable chosen ciphertext attack (RCCA) from [16] and adaptive-tag weakly chosen ciphertext attack (atag-wCCA) from [36], which were both shown to be weaker than CCA2 and used to construct secure channels. While this work does not provide an ultimate answer to this question—i.e., we do not prove that our definition, labeled indistinguishability under sender-binding chosen plaintext attack (IND-SB-CPA), is the weakest possible and hence necessary—we show IND-SB-CPA to be sufficient in the sense that any encryption protocol satisfying this security can be used directly to UC-realize SMT using authenticated channels.

Although this is an interesting theoretic result, we argue that for more relevancy the previous question needs to be accompanied by the following:

> *Can weaker security notions lead to simpler and more efficient constructions of a secure channel in the standard model?*

In the current state of affairs, tag-based encryption (TBE) is an attractive choice for constructing efficient CCA2 secure PKE in the standard model as already the weakest established TBE security notion, indistinguishability under selective-tag weakly chosen ciphertext attack (IND-stag-wCCA), was shown by

Kiltz [30] to yield a transformation to CCA2 secure PKE by adding one-time signatures for example. We show that IND-stag-wCCA secure TBE does not actually require prior transformation to CCA2 secure PKE in order to construct secure channels: By deriving the new concept of sender-binding encryption (SBE) from TBE we are able to construct secure channels directly from IND-stag-wCCA secure encryption. The intuition behind SBE is to tie ciphertexts not only to the receiver as with classic PKE notions, but to the sending/encrypting party as well.

Somewhat surprisingly, via IND-SB-CPA secure SBE we are also able to construct secure channels from double receiver encryption (DRE) which only satisfies CPA security and soundness. CPA secure DRE was initially introduced by Diament et al. [23] to facilitate message transmission from one sender to two different receivers and allows for interesting applications such as security puzzles for denial of service countermeasures. Subsequently, Chow et al. [19] introduced the property of soundness for DRE, and proved it to be crucial for some applications such as plaintext awareness (PA). Our DRE-based protocol allows for a much simpler and more efficient encryption than IND-stag-wCCA secure TBE for constructing secure channels and hence allows us to answer the second question in the positive.

One caveat of the construction via DRE is that we require an extended PKI that realizes the *key registration with knowledge (KRK)* functionality. This guarantees that users of the PKI have knowledge of their private keys. While this is not a common functionality of PKIs in use today, there are first protocol drafts like OTRv4[4] which utilize deniable authenticated key exchange protocols that rely on the KRK functionality. In this case those are DAKEZ and XZDH due to Unger and Goldberg [47].

As discussed in the next section the two questions we raise have partially been considered in prior works. In this paper we make considerable headway towards answering both of them.

1.1 Related Work

In this section we firstly analyze the current scientific landscape of security notions for SMT construction with authenticated channels. We then discuss the most promising prior constructions to efficiently achieve these security notions.

A PKE satisfying CCA2 security was already shown by Canetti in [13] to realize SMT in the UC framework by communicating confidentially over authenticated channels. On the other hand CCA2 was also shown by Canetti et al. [16] to be unnecessarily strong for this purpose. Hence relaxations of CCA2 came into focus. Among these relaxations is indistinguishability under replayable chosen ciphertext attack (IND-RCCA), introduced by Canetti, Krawczyk and Nielsen in [16] where they show that IND-RCCA suffices to UC-realize SMT using authenticated channels. IND-RCCA differs from CCA2 in the characteristic that the ability to generate ciphertexts, which decrypt to the same plaintext as

[4] https://github.com/otrv4/otrv4/blob/master/otrv4.md

the test ciphertext, does not help the adversary to win the game. We provide the formal notions of IND-RCCA in Appendix B.1. Recently, Badertscher et al. [1] examined IND-RCCA and variations of it using the constructive cryptography framework to construct a confidential channel—a strictly weaker notion than SMT. They concluded that IND-RCCA is not sufficient to realize confidential channels when using the authenticated channel for public key transfer only. They introduce a stronger security definition to solve this problem whereas we, like the original IND-RCCA paper, assume authentication for every message transfer.

Another direction to achieve weaker security definitions is that of TBE which was introduced by MacKenzie, Reiter and Yang [36]. They introduced the notion of tag-based non-malleability, which is nowadays known as indistinguishability under adaptive-tag weakly chosen ciphertext attack (IND-atag-wCCA) security for TBE. The authors show that an IND-atag-wCCA secure TBE scheme is also sufficient to realize SMT when provided with authenticated channels. A relaxation, IND-stag-wCCA, has been shown to facilitate CCA2 constructions with the additional usage of a one-time signature scheme [7] or a message authentication code combined with a commitment scheme [9]. Both constructions are originally meant for identity based encryption (IBE), but Kiltz showed in [30] how to adapt these for the TBE setting. So far IND-stag-wCCA secure TBE has not been shown, however, to directly facilitate SMT.

Let us now look at how efficiently these security notions can be achieved without employing the ROM. The most efficient general construction paradigms nowadays are the lossy trapdoor functions by Peikert and Waters [41], the correlated products by Rosen and Segev [44] and the very similar k-repetition by Döttling et al. [25][5], the Cramer-Shoup-like constructions [20] and the adaptive trapdoor functions [32]. More efficient constructions of SMT can be built upon TBE. The—to the best of our knowledge—most efficient code-based TBE schemes nowadays are due to Kiltz [30], Kiltz, Masny and Pietrzak [31], Cheng et al. [18] and Yu et al. [48]. In their schemes, the notion of IND-stag-wCCA security for TBE is required, which can be used to construct CCA2 schemes by adding one-time signatures or message authentication codes and commitments as mentioned above.

Regarding both of our research questions we see that although some progress was made in previous works there is still a lot of room for improvement. In the following section we highlight this paper's contribution towards closing this gap.

1.2 Our Contribution

In this paper we develop the new security notion of IND-SB-CPA, which is the weakest so far to UC-realize SMT in conjunction with authenticated channels. We also give a concrete efficient construction of an IND-SB-CPA secure SBE scheme in the standard model. An overview of this five-part contribution is illustrated in Figure 1. The five contribution parts correspond to the Sections 2 to 6:

[5] In spite of being a generic paradigm this work was applied only to McEliece so far.

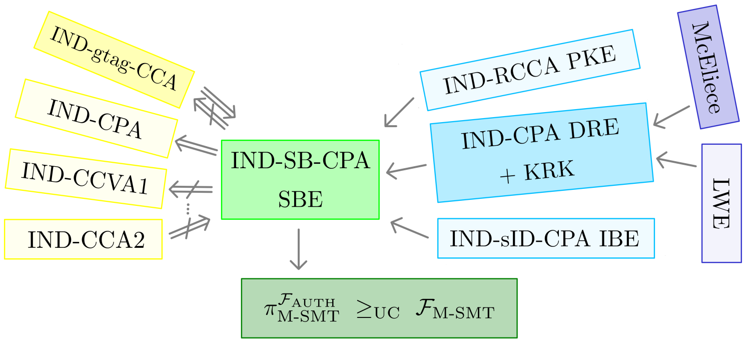

Fig. 1. Overview of Our Contribution

- In Section 2 we firstly provide the unifying definition of SBE, capturing all prior ways to construct SMT from authenticated channels and some form of encryption. A direct consequence is that all of the TBE notions, reformulated as SBE, directly construct SMT from authenticated channels. We then go on to develop the new game-based security notion of IND-SB-CPA. This is explicitly tailored to be as weak as possible while still only requiring authenticated channels to facilitate SMT. We achieve this by binding ciphertexts to sending parties.
- Section 3 presents a generic transformation from an indistinguishability under chosen plaintext attack (IND-CPA) secure DRE scheme with key registration to an IND-SB-CPA secure SBE scheme. To the extent of our knowledge it was not previously known how CPA secure DRE could be used to realize SMT. Appendix E presents further generic transformations based on IND-RCCA secure PKE and indistinguishability under selective identity chosen plaintext attack (IND-sID-CPA) secure IBE.
- In Section 4 we construct an IND-CPA secure and sound DRE scheme from a McEliece variant. In conjunction with Section 3 this can be used to implement SMT in a more efficient and simpler way than known so far. To the extent of our knowledge we are the first to construct a McEliece-based DRE with soundness. Moreover, we show an improvement of a factor 5 regarding the size of the public key, which is mostly due to the avoidance of relying solely on the (low-noise) learning parity with noise (LPN) assumption. Additionally, we provide another (2-repetition) McEliece construction and one from LWE-based binding encryption in Appendix F. All our constructions are proven secure in the standard model.
- In Section 5 we finally construct a protocol which combines IND-SB-CPA security with authenticated channels. This protocol is subsequently proven to UC-realize SMT under static corruption by a malicious adversary.
- Section 6 highlights the theoretical relation between IND-SB-CPA and TBE security notions—in particular that the new notion of IND-SB-CPA

is implied by the weakest known TBE security. Appendix G.2 expands on this theoretic classification by comparing IND-SB-CPA to classic PKE indistinguishability notions from CPA to CCA2.

1.3 Preliminaries

Firstly, let us note that all notations and abbreviations we use can be looked up in Appendix A. We talk about different *game-based security notions* for various types of encryption schemes throughout this paper. While we would expect the reader to be familiar with the standard definitions of IND-CPA/-CCA2 etc., we provide formal definitions of all notions for your convenience in Appendix B—in particular the more involved ones pertaining, e.g., to DRE, TBE and IBE schemes including security, correctness and soundness definitions.

In this work we use DRE as a building block for our construction. DRE encrypts a plaintext to two ciphertexts using two different public keys with the guarantee, that these ciphertexts decrypt to the same plaintext. Formally a DRE scheme consists of three probabilistic polynomial time (PPT) algorithms $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ and the function $\mathtt{f}_{\mathrm{Key}}$, which checks if the key pair (sk, pk) is well-formed.

$$\begin{aligned}
\mathtt{gen}: &\quad 1^\lambda \mapsto (sk, pk)\\
\mathtt{enc}: &\quad (pk_1, pk_2, m) \mapsto c\\
\mathtt{dec}: &\quad (sk_i, pk_1, pk_2, c) \mapsto m \text{ where } i \in \{1, 2\}\\
\mathtt{f}_{\mathrm{Key}}: &\quad (sk, pk) \mapsto \begin{cases}\mathsf{true}\\ \mathsf{false}.\end{cases}
\end{aligned}$$

TBE extends public key encryption by adding a tag to the encryption and decryption algorithms. This tag contains additional information and is a simple string. Formally a TBE scheme with message space $\mathbf{M}$ and tag space $\mathbf{T}$ consists of three PPT algorithms $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$.

$$\begin{aligned}
\mathtt{gen}: &\quad (1^\lambda) \mapsto (sk, pk)\\
\mathtt{enc}: &\quad (pk, t, m) \mapsto c\\
\mathtt{dec}: &\quad (sk, t, c) \mapsto m \in \mathbf{M} \cup \{\bot\}
\end{aligned}$$

The weakest security notion of TBE so far is IND-stag-wCCA introduced by Kiltz [30]. This and further definitions of TBE security can be found in Appendix B.2. The TBE notion IND-gtag-wCCA—which we start from to develop our notion of IND-SB-CPA security—is explicitly given in Section 2.

For readers who are not intimately familiar with the concept of *simulation-based security* or *universal composability* we also briefly recap the ideal/real-paradigm as well as UC in Appendix C. More detailed explanations can be found, for instance, in [12, 13]. As there have been conflicting definitions, we explicitly state formal definitions for the *ideal functionalities* of $\mathcal{F}_{\mathrm{AUTH}}$, $\mathcal{F}_{\mathrm{M\text{-}SMT}}$ and $\mathcal{F}_{\mathrm{KRK}}$. For $\mathcal{F}_{\mathrm{AUTH}}$ and $\mathcal{F}_{\mathrm{M\text{-}SMT}}$ these can be found in Section 5 and additionally with further discussion in Appendix D. Also, the definition for $\mathcal{F}_{\text{KRK}}$ can be found in Appendix D.

2 IND-SB-CPA Security

SMT is commonly realized by combining an IND-CCA2 secure PKE or an IND-atag-wCCA secure TBE scheme with authenticated channels. As highlighted in Section 1, however, both of those security notions seem to be unnecessarily strong and restrictive for this application. In this observation we are hardly the first (cp. Section 1.1) as there are previous efforts to relax security notions with the aim to facilitate SMT—like the RCCA relaxation of CCA2 and efforts to use IND-stag-wCCA secure TBE.

In this section we introduce the concept of SBE and our new security notion of IND-SB-CPA. It is even weaker than the IND-atag-wCCA relaxation IND-stag-wCCA but still captures the security needed for secure message transfer via authenticated channels. Although the term SBE has not previously been defined, all prior realizations of SMT via authenticated channels (based on CCA2, RCCA, atag-wCCA or selective-tag weakly chosen ciphertext attack (stag-wCCA)) work by constructing an SBE scheme from the underlying encryption scheme. We therefore regard this as a long overdue unifying definition which is central for the topic of SMT construction.

Definition 1 (Sender-binding encryption (SBE)). *The interface of an SBE scheme is given by a set of three PPT algorithms* $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$*:*

$$\begin{aligned} \mathtt{gen}: &\quad 1^\lambda \mapsto (sk, pk) \\ \mathtt{enc}: &\quad (pk, S, m) \mapsto c \\ \mathtt{dec}: &\quad (sk, S, c) \mapsto m. \end{aligned}$$

We expect an SBE scheme to fulfill the notion of correctness, i.e. that whenever $(sk, pk) \leftarrow \mathtt{gen}(1^\lambda)$*, then*

$$m = \mathtt{dec}(sk, S, \mathtt{enc}(pk, S, m)).$$

Some remarks are in order about this use case definition of SBE.

In addition to the inputs present in any common PKE scheme, encryption and decryption algorithms use the encrypting party's ID S[6] as well. The ID of a party represents the identification information used within the system. This might be the public key itself, the party's actual name, their e-mail address etc. This does not only bind a ciphertext to the receiving party who holds the secret key and is able to decrypt the ciphertext—as any PKE scheme does—but also to the party who created the encryption.

However, binding a ciphertext to the ID of a sending/encrypting party alone does not yet yield obvious benefits. Even if a specific party ID is specified by

[6] For the encryption mechanism we will sometimes omit the explicit input of the ID S if it is clear from the context which party S is conducting the encryption.

the protocol, party IDs are public knowledge and malicious parties can insert any ID they want. SBE starts to unfold its benefit when used in conjunction with IDs that are associated with authenticated channels. This channel reliably indicates the true sender S of a message. Checking this against the sender ID bound to the received ciphertext prevents (honest sender) replay attacks, i.e., that this message was just copied from another (unwitting) sender. The terminology "sender-binding" stems from the example application of SMT via authenticated channels where this is taken to be the encrypting/sending party. Of course there might be other use cases for SBE where the encrypting party does not constitute a "sender". But throughout this paper (whenever we talk about SBE) we use R and "receiver" to denote the party owning the keys $(sk_R, pk_R) := (sk, pk)$, and S and the term "sender" for the party whose ID is input on encryption and decryption.

Given the definition of an SBE scheme we still need to arrive at a meaningful corresponding security notion. The intuitive way to construct an SBE scheme is to use a TBE scheme where the tag space $\mathbf{T}$ is chosen to be the set of party IDs $\mathbf{P}$. Even a TBE scheme with arbitrary tag space $\mathbf{T}$ can easily be used for SBE as long as the tag space is as least as large as the set $\mathbf{P}$ of participating parties. To do so a public and injective function $\mathbf{P} \hookrightarrow \mathbf{T}$ is chosen to translate party IDs into tags. Hence to develop a security notion for SBE we start from the TBE notion indistinguishability under given-tag weakly chosen ciphertext attack (IND-gtag-wCCA). This is an intuitive weakening of the previously considered IND-stag-wCCA, with the only difference being that the adversary is not allowed to choose the challenge tag but is instead given a random tag by the challenger:

```
$\mathsf{Exp}^{\text{IND-gtag-wCCA}}_{\text{TBE},\mathcal{A}}$

(1) $t^* \xleftarrow{R} \mathbf{T}$
    $(sk, pk) \leftarrow \mathtt{gen}(1^\lambda)$
(2) $(st, m_0, m_1) \leftarrow \mathcal{A}^{\mathtt{dec}(sk,\cdot,\cdot)^a}(t^*, pk)$
(3) $b \xleftarrow{R} \{0,1\}$
    $c^* \leftarrow \mathtt{enc}(pk, t^*, m_b)$
(4) $b^* \leftarrow \mathcal{A}^{\mathtt{dec}(sk,\cdot,\cdot)^a}(st, c^*)$
(5) Return 1 if $b = b^*$, else return 0
```

[a] Decryption outputs $\perp$ for tags $t^* \in \{S, R\}$.

Fig. 2. The IND-gtag-wCCA TBE Game.

Definition 2 (IND-gtag-wCCA). *A TBE scheme* (gen, enc, dec) *satisfies IND-gtag-wCCA security, if and only if for any PPT adversary $\mathcal{A}_{gtag\text{-}CCA}$ the advantage to win the IND-gtag-wCCA game shown in Figure 2 is negligible in λ.*

Using party IDs as tags in TBE provides a special meaning to these tags. It is this additional meaning which induces the changes we make to IND-gtag-wCCA to arrive at our new notion of IND-SB-CPA for SBE: We now additionally have a connection between tags and key pairs, as any party ID (tag) is associated to the key pair of this party. Hence there is another ID/tag R corresponding to the key pair $(sk_R, pk_R) = (sk, pk)$ and another key pair (sk_S, pk_S) corresponding to the party $S = t^*$. As we are aiming towards the weakest possible notion from which to construct SMT we let both of those be chosen by the challenger instead of giving the adversary any more power. Depending on the underlying encryption scheme it is possible that keys may not be generated independently of the ID (think, e.g., of IBE schemes) or that public keys are used as IDs themselves. Hence we assume the challenger to randomly generate/draw keys and IDs in a consistent fashion. With the additional key pair (sk_S, pk_S) we also need to define how much decryption power the adversary gets for these keys in the two oracle phases. We choose this intuitively to be symmetric with the challenge keys (sk_R, pk_R). Because this gives a weaker notion and is still enough for SMT we restrict decryption not only for the challenge tag S but for R as well. All in all this adjustment of IND-gtag-wCCA to SBE yields the following definition:

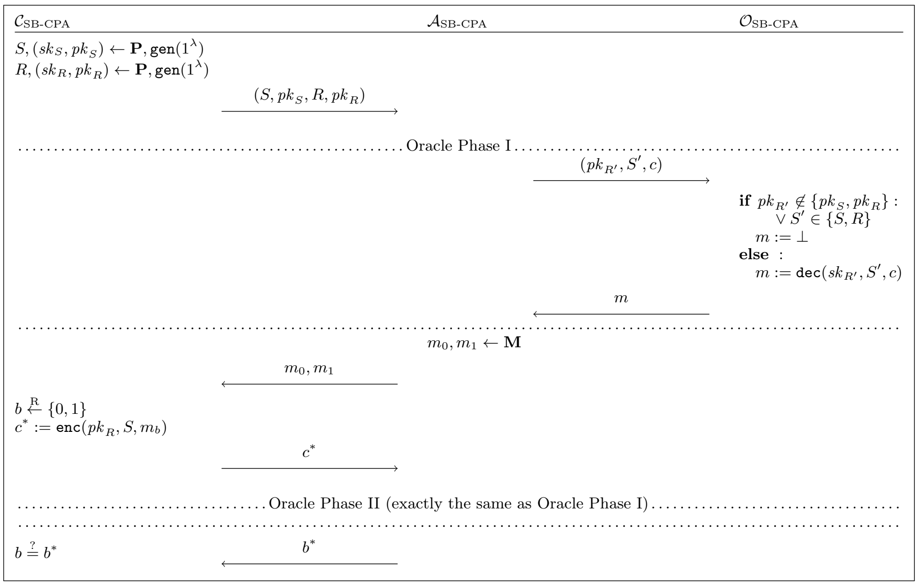
$\mathcal{C}_{\text{SB-CPA}}$
$\mathcal{A}_{\text{SB-CPA}}$
$\mathcal{O}_{\text{SB-CPA}}$
$S, (sk_S, pk_S) \leftarrow \mathbf{P}, \mathtt{gen}(1^\lambda)$
$R, (sk_R, pk_R) \leftarrow \mathbf{P}, \mathtt{gen}(1^\lambda)$
(S, pk_S, R, pk_R)
Oracle Phase I
$(pk_{R'}, S', c)$
if $pk_{R'} \notin \{pk_S, pk_R\}$:
$\vee S' \in \{S, R\}$
$m := \perp$
else :
$m := \mathtt{dec}(sk_{R'}, S', c)$
m
$m_0, m_1 \leftarrow \mathbf{M}$
m_0, m_1
$b \xleftarrow{\text{R}} \{0,1\}$
$c^* := \mathtt{enc}(pk_R, S, m_b)$
c^*
Oracle Phase II (exactly the same as Oracle Phase I)
$b \overset{?}{=} b^*$
b^*

Fig. 3. The IND-SB-CPA Game for SBE

Definition 3 (IND-SB-CPA). *An SBE scheme* $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ *satisfies IND-SB-CPA security, if and only if for any PPT adversary* $\mathcal{A}_{SB\text{-}CPA}$ *the advantage to win the IND-SB-CPA game shown in Figure 3 is negligible in* λ.

Within this context of SBE, the new security notion of IND-SB-CPA has a very straight forward intuition: If it was possible to alter a ciphertext $c \leftarrow \mathtt{enc}(pk, S, m)$ to some c' which successfully decrypted under another sender ID S' (i.e. $\mathtt{dec}(sk_R, S', c') \neq \perp$), replay attacks would be possible. Let us look at this in a bit more detail. From Figure 3 we see that the adversary is provided with perfect knowledge (via oracle or its own power) about any ciphertext which involves any other party than just S and R. About communication between S and R, on the other hand, the adversary learns nothing—with the natural exception that encryption only requires public knowledge and can therefore be conducted by the adversary as well. A directed version—where the adversary can additionally decrypt messages from R to S (but not from S to R)—would also naturally suggest itself. But as mentioned before our choice of a symmetric version is strictly weaker as well as sufficient for SMT construction. Having no decryption possibilities for the channel (S to R) along which the challenge ciphertext is sent justifies classifying IND-SB-CPA as some form of CPA security. For more thoughts on these classifications see Appendix G.3.

We thoroughly investigate the relationships between IND-SB-CPA and other game-based notions in Section 6 and Appendix G.2. In the next section we show that IND-SB-CPA is not merely of academic interest by giving a generic example construction for IND-SB-CPA secure SBE via DRE.

3 Transformation from DRE to SBE

In this section we generically construct an IND-SB-CPA secure SBE scheme from DRE. Further generic constructions as well as more involved discussions of this DRE construction—particular about the use of KRK—can be found in Appendix E.

Originally meant to encrypt a message to two receivers, we use DRE in such a way, that one of those ciphertexts is encrypted using the public key of the sender. This, together with the usage of PKIs using KRK results in an encryption where the sender is aware of the plaintext. Without KRK there is no guarantee that the sender has knowledge of the private key corresponding to his public key, so this awareness could not be guaranteed. A possible realization of the KRK functionality is that the PKI demands a zero-knowledge proof of knowledge about the secret key when registering the public key. While this is a possibly expensive operation it only needs to be done once while registering.

We require the underlying DRE scheme to be sound, IND-CPA secure and compatible with the key registration functionality $\mathcal{F}_{\mathrm{KRK}}$. For the definition of DRE, its soundness, and the definition of $\mathcal{F}_{\mathrm{KRK}}$ we refer the reader to Appendix B.4 and Appendix D respectively. This transformation will broaden our intuitive understanding of the new notion as well as provide a background for the concrete DRE construction we discuss in Section 4. We furthermore use the transformation in Section 6 to show that IND-SB-CPA does not in fact imply IND-gtag-wCCA but is a strictly weaker security notion.

Although DRE was initially devised to facilitate message transmission from one sender to two different receivers, choosing one of the receivers to be the sender itself provides a way to bind the ciphertext to the sender and to achieve an IND-SB-CPA secure SBE scheme.

One small caveat of using DRE is the need for key registration with knowledge: If we can not make sure the sender knows a key pair, ciphertexts encrypted under this key will not establish a reliable connection between ciphertext and sender. Hence we employ the ideal functionality $\mathcal{F}_{\text{KRK}}$. To do so, however, we need to make sure the underlying DRE scheme is compatible:

Remark 1. Throughout this section we will assume DRE schemes to permit efficiently computable boolean functions $\mathtt{f}_{\text{Key}}$. On input of a (possible) key pair (sk, pk) this function decides whether the keys "belong together", i.e., whether they could have been output by the encryption scheme's key generation algorithm or might just be an unrelated pair of values:

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

This is necessary for the scheme to be used in conjunction with the registration functionality $\mathcal{F}_{\text{KRK}}$. In Appendix D we discuss $\mathcal{F}_{\text{KRK}}$ a bit more and also see that we can easily dispose of the need for a function $\mathtt{f}_{\text{Key}}$ if we are happy for the registration functionality to (partially) generate the keys for the registering parties.

Let $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ be an IND-CPA secure DRE scheme which admits a function $\mathtt{f}_{\text{Key}}$. We define a new encryption scheme $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$:

$\mathtt{Gen}(1^\lambda)$ 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 $(SK, PK) := ((sk, pk), P)$.

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

$\hookrightarrow$ Return $c \leftarrow \mathtt{enc}(pk_R, pk_S, m)$.

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

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

Let us give some intuition about the construction before we move on to formalities. Choosing one of the receivers for DRE to be the sender itself and having them encrypt a message under its own key might seem counterintuitive at first, but has one crucial benefit: It guarantees to the other (actual) receiver that even if the sender might not have constructed the ciphertext themselves but rather copied it from somewhere else, they have knowledge about the plaintext since they are able to decrypt as well. This is guaranteed by the registration with

$\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}$ in conjunction with the soundness property of the underlying DRE scheme. In addition to showing that this construction does in fact satisfy IND-SB-CPA security, we provide a discussion in Appendix E on what properties exactly we need from DRE and how this is related to registration-based plaintext awareness (RPA).

Lemma 1. *In the $\mathcal{F}_{KRK}^{\mathtt{f}_{Key}}$ hybrid model* $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ *is an IND-SB-CPA secure SBE scheme.*

Proof. Assuming that $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ is a sound DRE scheme with key function $\mathtt{f}_{\text{Key}}$ and assuming we have an adversary $\mathcal{A}_{\text{SB-CPA}}$ who has non-negligible success probability in winning the IND-SB-CPA game with respect to $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$, we construct an adversary $\mathcal{A}_{\text{DRE-CPA}}$ with non-negligible success probability in winning the DRE IND-CPA game with respect to $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$. Note that in this case, $\mathcal{A}_{\text{DRE-CPA}}$ not only fields $\mathcal{A}_{\text{SB-CPA}}$'s queries to $\mathcal{O}_{\text{SB-CPA}}$ but also plays the role of $\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}$ and has therefore access to registered keys. In the reduction shown in Figure 4 we do not explicitly state this, but all interactions with $\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}$ are handled exactly as the functionality itself would. The only exceptions are that an instantaneous $\mathtt{ok}$ is assumed whenever the functionality would ask the adversary for some permission and that in the first phase the adversary $\mathcal{A}_{\text{DRE-CPA}}$ itself "registers" the keys pk_S and pk_R for S and R respectively without providing corresponding secret keys.

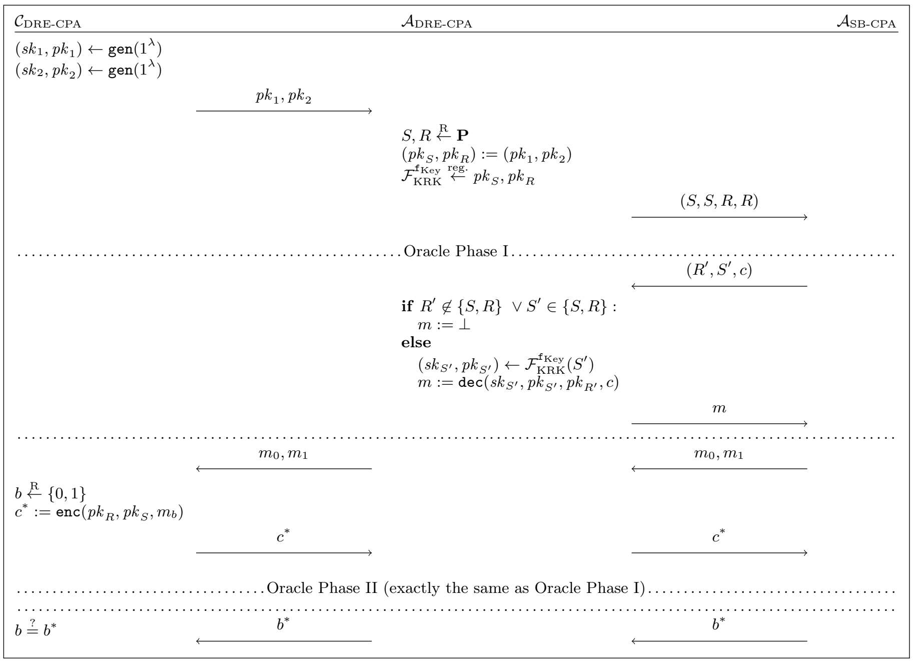
$\mathcal{C}_{\text{DRE-CPA}}$
$\mathcal{A}_{\text{DRE-CPA}}$
$\mathcal{A}_{\text{SB-CPA}}$
$(sk_1, pk_1) \leftarrow \mathtt{gen}(1^\lambda)$
$(sk_2, pk_2) \leftarrow \mathtt{gen}(1^\lambda)$
pk_1, pk_2
$S, R \overset{R}{\leftarrow} \mathbf{P}$
$(pk_S, pk_R) := (pk_1, pk_2)$
$\mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}} \overset{\text{reg.}}{\leftarrow} pk_S, pk_R$
(S, S, R, R)
Oracle Phase I
(R', S', c)
if $R' \notin \{S, R\} \vee S' \in \{S, R\}$:
$m := \perp$
else
$(sk_{S'}, pk_{S'}) \leftarrow \mathcal{F}_{\text{KRK}}^{\mathtt{f}_{\text{Key}}}(S')$
$m := \mathtt{dec}(sk_{S'}, pk_{S'}, pk_{R'}, c)$
m
m_0, m_1
m_0, m_1
$b \overset{R}{\leftarrow} \{0, 1\}$
$c^* := \mathtt{enc}(pk_R, pk_S, m_b)$
c^*
c^*
Oracle Phase II (exactly the same as Oracle Phase I)
$b \overset{?}{=} b^*$
b^*
b^*

Fig. 4. Reduction for DRE Construction

Since $\mathcal{A}_{\text{DRE-CPA}}$ has access to the internal state of $\mathcal{F}_{\text{KRK}}^{\mathbf{f}_{\text{Key}}}$, they can look up the keys $(sk_{S'}, pk_{S'})$ for any oracle query (R', S', c). If no such keys have been registered, decryption of the ciphertext would result in $\perp$. If keys have been registered, they can be used to correctly decrypt the ciphertext as the soundness of DRE (see Appendix B.4 for definition) guarantees

$$\mathtt{dec}(sk_{S'}, pk_{S'}, pk_{R'}, c) = \mathtt{dec}(sk_{R'}, pk_{R'}, pk_{S'}, c).$$

Hence it is no problem for $\mathcal{A}_{\text{DRE-CPA}}$ to respond with correct decryptions exactly as $\mathcal{O}_{\text{SB-CPA}}$ would. This gives $\mathcal{A}_{\text{DRE-CPA}}$ the same non-negligible success probability as $\mathcal{A}_{\text{SB-CPA}}$. □

This newfound utility for IND-CPA secure DRE schemes provides the motivational background for the next section, which in turn shows the relevance of our theoretical construction for the efficient construction of SMT in the standard model.

4 Efficient DRE Construction from McEliece and LPN

In this section we present an efficient way to construct an IND-CPA secure and sound DRE scheme from the McEliece and LPN assumptions and discuss how our construction improves the state of the art of SMT realizations in the standard model based on the McEliece and LPN assumptions. Moreover, to the extent of our knowledge we are the first to construct a DRE based on these assumptions. More details on our construction as well as further constructions via 2-repetition McEliece and learning with errors (LWE)-based binding encryption can be found in Appendix F.

Construction. Our DRE scheme can be seen as an augmentation of a construction from Kiltz et al. [31]. In this the authors propose a creative construction of a low-noise LPN-based TBE scheme, which they show to be IND-stag-wCCA secure. In the appendix of [31] the authors introduce a simplified variant of their IND-stag-wCCA secure construction, which is only IND-CPA secure. We use this simplified variant as a basis for our own construction. In order to establish the soundness property we add a second encryption of the randomness and exploit the randomness recovery to perform the consistency check. Moreover, we change the trapdoor mechanism to the one from the McEliece cryptosystem over Goppa codes. Hence we define our DRE scheme $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ as follows:

- $\mathtt{gen}$ Generate the McEliece secret key $sk := (\mathsf{S}, \mathsf{G}', \mathsf{P})$ and corresponding public key $pk := (\mathsf{G}, \mathsf{C})$ where $\mathsf{G} := \mathsf{S}\mathsf{G}'\mathsf{P}$ and C is a random binary matrix.
- $\mathtt{enc}$ Sample a fresh random vector s, fresh error vectors $\mathsf{e}, \mathsf{e}_R, \mathsf{e}_S$ and encrypt s for both sender S and receiver R, i.e., $\mathsf{c}_S := \mathsf{s} \cdot \mathsf{G}_S \oplus \mathsf{e}_S$ and $\mathsf{c}_R := \mathsf{s} \cdot \mathsf{G}_R \oplus \mathsf{e}_R$. Mask the encoded message m with the noisy product $\mathsf{s} \cdot \mathsf{C}_S \oplus \mathsf{e}$, i.e. $\mathsf{c}' = \mathsf{s} \cdot \mathsf{C}_S \oplus \mathsf{e} \oplus Encode(m)$ and output $c := (\mathsf{c}_R, \mathsf{c}_S, \mathsf{c}')$ as the ciphertext.

`dec` The receiver recovers the randomness s from c_R with textbook McEliece decryption, verifies the hamming weight $wgt(\mathsf{s} \cdot \mathsf{G}_S \oplus \mathsf{c}_S) < t$ and unmasks $Encode(m) \oplus \mathsf{e} = \mathsf{c}' \oplus \mathsf{s} \cdot \mathsf{C}_S$. Finally, the receiver decodes and outputs the message m.

For the encoding and decoding we propose to use a suitable Goppa code, which is fixed for all parties. More details can be found in Appendix F.

Theorem 1. *The DRE scheme* $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ *is IND-CPA 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-CPA adversary against the cryptosystem. Then there is a distinguisher $\mathcal{B}$ for Goppa codes and a distinguisher $\mathcal{D}$ for the LPNDP, such that for all $\lambda \in \mathbb{N}$*

$$\mathsf{Adv}_{\mathcal{A}}^{CPA}(\lambda) \leq \mathsf{Adv}_{\mathcal{D}}^{LPNDP_\theta(3n,l)}(\lambda) + 2 \times \mathsf{Adv}_{\mathcal{B}_R, G_R}^{IND}(\lambda).$$

Theorem 2. *The DRE scheme* $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ *satisfies DRE soundness.*

The proofs and formal definitions of assumptions and experiments can be found in Appendix F as well. Note also, that this DRE scheme admits an efficiently computable function $\mathtt{f}_{\mathrm{Key}}$ as required for the use with $\mathcal{F}_{\mathrm{KRK}}$ (cp. Section 3):

$$\mathtt{f}_{\mathrm{Key}} : ((\mathsf{S}, \mathsf{P}, \mathsf{G}'), (\mathsf{G}, \mathsf{C})) \mapsto \begin{cases} \mathsf{true}, & \mathsf{G} = \mathsf{S}\mathsf{G}'\mathsf{P} \\ \mathsf{false}, & \text{else.} \end{cases}$$

In conjunction with Theorems 1 and 2 our DRE scheme satisfies all requirements for the generic transformation to IND-SB-CPA given in Section 3. Hence we can use it to efficiently achieve SMT if combined with authenticated channels.

Discussion. Considering that one of the third round finalists of the post-quantum cryptography (PQC) standardization by the NIST[7] is a McEliece variant based on Goppa codes we expect this mechanism to have significantly better parameters than cryptosystems that are based solely on the (low noise) LPN assumption. We argue, however, that our construction may as well be realized with the sole (low noise) LPN assumption or the Niederreiter cryptosystem [39]. Also, a similar augmentation of the randomness recovering variant of the dual Regev [27] cryptosystem may yield a very similar construction of DRE based on LWE. Currently, the Niederreiter cryptosystem seems the most promising as it was already shown in [26] that the trapdoor function is one-way under k-correlated input. The tightness loss is expected to be a factor of 3 regarding the number of LPNDP samples and a factor of 2 regarding the indistinguishability assumption. Therefore, we expect our construction of DRE to have roughly the same parameters as their single receiver IND-CPA counterparts without the soundness. An algebraic comparison of the public keys and the ciphertext from

Construction	Public Key	Ciphertext
Kiltz et al. [31]	$(A, B_0, B_1, C) \in (\mathbb{Z}_2^{m \times n'})^3 \times \mathbb{Z}_2^{l' \times n'}$	$(c, c_0, c_1, c_2) \in (\mathbb{Z}_2^{m})^3 \times \mathbb{Z}_2^{l'}$
Yu et al. [48]	$(A, B_0, B_1, C) \in \mathbb{Z}_2^{\overline{n} \times \overline{n}} \times (\mathbb{Z}_2^{q \times \overline{n}})^2 \times \mathbb{Z}_2^{\overline{l} \times \overline{n}}$	$(c, c_0, c_1, c_2) \in (\mathbb{Z}_2^{\overline{n}}) \times (\mathbb{Z}_2^{q})^2 \times \mathbb{Z}_2^{\overline{l}}$
This Work	$(G, C) \in \mathbb{Z}_2^{l \times n} \times \mathbb{Z}_2^{l \times n}$	$(c_R, c_S, c') \in (\mathbb{Z}_2^{n})^3$

Table 1. Comparison of public keys and ciphertext between [31, 48] and this work.

our work and the current state of the art in [31] and [48] can be found in the table 1.

At this point some remarks are necessary to understand the comparisons more thoroughly. For the sake of simplicity we will give rough estimations of the respective public key sizes. Kiltz et al. [31] require for their dimensions that $m \geq 2n'$ and $l' \geq m$, where n' is the dimension of the low-noise LPN secret. Current estimations suggest that cryptosystems based on low-noise LPN to have rather large dimensions, e.g., [21] suggest for 80 bits of security $n' = 9000$ when the noise is $\mu = 0.0044$. Therefore, setting $n' = 9000$ leads to the smallest possible $m = 18000$ and $l' = 18000$ and results in a public key size of roughly 77 megabyte.

Yu et al. [48] improved the construction of [31] in such a way that it may be based on constant noise LPN assuming sub-exponential hardness. Current estimations of concrete constant noise LPN hardness suggest much smaller dimensions than in the low-noise variant, e.g., [6] suggest for 80 bits of security $\overline{n} = 1280$ and noise level of $\mu = 0.05$, which meets the restriction from [48] that $\mu \leq 0.1$. The crucial parameter is, however, the choice of an $\alpha > 0$ as this parameter controls the dimension $q = O(\overline{n}^{6 \cdot \alpha + 1})$, which means that minimizing α will minimize the size of the public key. In order to estimate α as small as possible we take the formula $\beta = \frac{1}{2} - \frac{1}{\overline{n}^{3 \cdot \alpha}}$, which controls the number $\beta \cdot q$ of bit flipping errors that a suitable error correcting code will correct. For the sake of simplicity we set $\alpha = 0.04$, which is almost the minimal possible α for an $\overline{n} = 1280$, and get approximately $q = 7127$. Finally, fixing the remaining dimension $\overline{l} = \overline{n}$ we get a public key size of roughly 2.5 megabyte, which is a substantial improvement compared to [31].

For classic McEliece constructions Bernstein et al. [5] suggests for 80 bits of security to utilize $[1632, 1269]$ Goppa codes. Setting $n = 1632$ and $l = 1269$ in this work leads to a public key size of roughly 505 kilobyte, which is roughly factor 5 smaller than previous works.

We would like to point out that constructions from [31] and [48] are not directly comparable to our construction because we rely on the additional indistinguishability assumption of Goppa codes from random linear codes. However, all three constructions are code-based and implement a secure channel such that (rough) estimations of concrete sizes regarding the same security level may help to understand the improvement.

[7] National Institute of Standards and Technology

5 Realizing $\mathcal{F}_{\textbf{M-SMT}}$ from IND-SB-CPA and $\mathcal{F}_{\textbf{AUTH}}$

In this section we show that IND-SB-CPA secure SBE suffices in conjunction with authenticated channels to realize SMT. We prove this in the universal composability (UC) model of Canetti [13] (which is explained in more detail in Appendix C) using *static corruptions* only. This means that the adversary chooses which parties to corrupt at the start of the protocol execution and not adaptively as the computation proceeds. We provide the formal definitions of the UC functionalities $\mathcal{F}_{\text{AUTH}}$ for authenticated channels and $\mathcal{F}_{\text{M-SMT}}$ for SMT to clarify which exact definitions we use. The latter deals with multiple receivers, multiple senders *and* multiple messages rather than working with a multi-session extension (cp. [17]) of a functionality $\mathcal{F}_{\text{SMT}}$ which only transmits a single message. Note that this is just a technical difference but essentially equivalent to the base of many arisen different definitions for SMT over the past. For more detailed discussions on these ideal functionalities see Appendix D.

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

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

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

$\mathcal{F}_{\textbf{M-SMT}}$

Provides:
Multi-receiver multi-message multi-sender secure message transfer with constant message size and polynomially many parties $P \in \mathbf{P}$.
State:
Function $p_{\text{Msg}} : \mathbf{SID} \times \mathbf{MID} \to \mathbf{M} \times \mathbf{P}^2$ of pending messages.
Behaviour:

- Upon receiving $(\texttt{send}, \mathit{sid}, R, m)$ from some party S, draw fresh mid, send $(\texttt{send}, \mathit{sid}, \mathit{mid}, S, R)$ to the adversary $\mathcal{A}$ and append $(\mathit{sid}, \mathit{mid}) \mapsto (m, S, R)$ to p_{Msg}.
- Upon receiving $(\texttt{send ok}, \mathit{sid}, \mathit{mid})$ from the adversary, look up $(m, S, R) := p_{\text{Msg}}(\mathit{sid}, \mathit{mid})$. If it exists, output $(\texttt{sent}, \mathit{sid}, S, m)$ to R.

We will proceed towards the goal of realizing SMT in three stages: Firstly, we define a candidate protocol $\pi^{\mathcal{F}_{\text{AUTH}}}_{\text{M-SMT}}$ in the $\mathcal{F}_{\text{AUTH}}$-hybrid model which utilizes an IND-SB-CPA secure SBE scheme. Secondly, we construct a simulator $\mathcal{S}_{\text{M-SMT}}$ aiming to provide indistinguishability between the candidate protocol and the SMT functionality $\mathcal{F}_{\text{M-SMT}}$. The last step is formally proving that in the $\mathcal{F}_{\text{AUTH}}$-hybrid model indistinguishability from $\mathcal{F}_{\text{M-SMT}}$ is actually achieved by $\pi^{\mathcal{F}_{\text{AUTH}}}_{\text{M-SMT}}$ in conjunction with $\mathcal{S}_{\text{M-SMT}}$.

Protocol $\pi^{\mathcal{F}_{\text{AUTH}}}_{\text{M-SMT}}$. Let $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ be an IND-SB-CPA secure SBE scheme. From this we define a secure message transfer protocol $\pi^{\mathcal{F}_{\text{AUTH}}}_{\text{M-SMT}}$ as follows: Whenever a party S wants to securely transmit a message m to some party R, they essentially send the encryption $c \leftarrow \texttt{enc}(pk_R, S, m)$ over an authenticated channel to R. When a party R receives a ciphertext c over an authenticated channel from some party S, they decrypt it via $m := \texttt{dec}(sk_R, S, c)$. Although this general principle is very simple, many details—e.g. regarding key generation—need to be taken into account. The formal definition looks as follows:

$\pi^{\mathcal{F}_{\text{AUTH}}}_{\text{M-SMT}}$

Realizes:
Multi-receiver multi-message multi-sender secure message transfer with constant message size.

Parameters:

- IND-SB-CPA secure SBE scheme $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ with message size l and ciphertext length l'.
- Functionality $\mathcal{F}_{\text{AUTH}}$.

State of Party P:

- Function $p_{\text{Cred}} : \mathbf{SID} \to \mathbf{SK} \times \mathbf{PK}$ of own credentials.
- Function $p_{\text{Pk}} : \mathbf{SID} \times \mathbf{P} \to \mathbf{PK}$ of known public keys.
- Function $p_{\text{Send}} : \mathbf{SID} \times \mathbf{P} \to \mathbf{M}^*$ of pending messages.

Behaviour of Party P:

\\ Being asked to initialize

- Upon receiving output $(\texttt{sent}, sid_{\text{AUTH}}, S, P, (\texttt{init}, sid))$ from $\mathcal{F}_{\text{AUTH}}$, if there is no entry $p_{\text{Cred}}(sid)$ yet:
  (1) $(sk, pk) \leftarrow \texttt{gen}(1^\lambda)$.
  (2) Append $sid \mapsto (sk, pk)$ to p_{Cred}.
  (3) For each party $P' \neq P$: Draw fresh sid'_{AUTH} and call $\mathcal{F}_{\text{AUTH}}$ with input $(\texttt{send}, sid'_{\text{AUTH}}, P', (\texttt{inited}, sid, pk))$.

\\ Receiving keys and sending stored messages

- Upon receiving output $(\texttt{sent}, sid_{\text{AUTH}}, P', P, (\texttt{inited}, sid, pk_{P'}))$ from $\mathcal{F}_{\text{AUTH}}$, if there is no entry $p_{\text{Pk}}(sid, P')$ yet:
  (1) Append $(sid, P') \mapsto pk_{P'}$ to p_{Pk}.
  (2) For any $m \in p_{\text{Send}}(sid, P')$:

```
        (1) Remove $m$ from $p_{\mathrm{Send}}(sid, P')$.
        (2) $c \leftarrow \mathtt{enc}(pk_{P'}, P, m)$.
        (3) Draw fresh $sid_{\mathrm{AUTH}}$.
        (4) Call $\mathcal{F}_{\mathrm{AUTH}}$ with input $(\mathtt{send}, sid_{\mathrm{AUTH}}, P', (sid, c))$.

\\ Sending messages

  • Upon receiving input $(\mathtt{send}, sid, R, m)$ with $m \in \{0,1\}^l$ from environment $\mathcal{Z}$:
      ◦ If $R = P$ report output $(\mathtt{sent}, sid, P, m)$ to the environment.
      ◦ Else if no entry $p_{\mathrm{Pk}}(sid, R)$ exists yet:
        (1) Append $m$ to $p_{\mathrm{Send}}(sid, R)$.
        (2) Draw fresh $sid_{\mathrm{AUTH}}$.
        (3) Call $\mathcal{F}_{\mathrm{AUTH}}$ with input $(\mathtt{send}, sid_{\mathrm{AUTH}}, R, (\mathtt{init}, sid))$.
      ◦ Else:
        (1) $pk_R := p_{\mathrm{Pk}}(sid, R)$.
        (2) $c \leftarrow \mathtt{enc}(pk_R, P, m)$.
        (3) Draw fresh $sid_{\mathrm{AUTH}}$.
        (4) Call $\mathcal{F}_{\mathrm{AUTH}}$ with input $(\mathtt{send}, sid_{\mathrm{AUTH}}, R, (sid, c))$.

\\ Receiving messages

  • Upon receiving output $(\mathtt{sent}, sid_{\mathrm{AUTH}}, S, R, (sid, c))$ from $\mathcal{F}_{\mathrm{AUTH}}$:
    (1) Look up $pk_S := p_{\mathrm{Pk}}(sid, S)$. If this does not exist, abort.
    (2) $m \leftarrow \mathtt{dec}(sk, S, c)$.
    (3) Report output $(\mathtt{sent}, sid, S, m)$ to the environment $\mathcal{Z}$.
```

Simulator $\mathcal{S}_{\text{M-SMT}}$. According to the real/ideal paradigm explained in Appendix C, our protocol $\pi_{\text{M-SMT}}^{\mathcal{F}_{\mathrm{AUTH}}}$ realizes secure message transfer if and only if for any (dummy) adversary $\mathcal{A}$ interacting with the real protocol, there exists a simulator $\mathcal{S}$ interacting with the ideal functionality $\mathcal{F}_{\text{M-SMT}}$ such that no environment $\mathcal{Z}$ can distinguish between executions in the real and ideal world. We now construct such a simulator $\mathcal{S}_{\text{M-SMT}}$ which we will later show to achieve indistinguishability for $\pi_{\text{M-SMT}}^{\mathcal{F}_{\mathrm{AUTH}}}$ and $\mathcal{F}_{\text{M-SMT}}$.

The main idea of the simulator $\mathcal{S}_{\text{M-SMT}}$ is that it simulates the protocol behaviour of all parties and the hybrid functionality $\mathcal{F}_{\mathrm{AUTH}}$ in its head. It takes inputs to and reports messages and outputs from these in-the-head parties to $\mathcal{Z}$ on the one hand and uses them on the other hand to interface with the ideal functionality $\mathcal{F}_{\text{M-SMT}}$. The only case in which the simulator does not have sufficient knowledge to perfectly simulate the protocol in their head is when an honest party S sends a message m to another honest party R: The simulator has no way of knowing the actual message m. In this case $\mathcal{S}_{\text{M-SMT}}$ reports an encryption $c \leftarrow \mathtt{enc}(pk_R, S, 0)$ of zero to have been send instead.

The overall construction of $\mathcal{S}_{\text{M-SMT}}$ is shown in Figure 5. Again there are some more details to keep track of (especially regarding the box labeled “Behaviour” in Figure 5) so we provide a more formal definition as well:

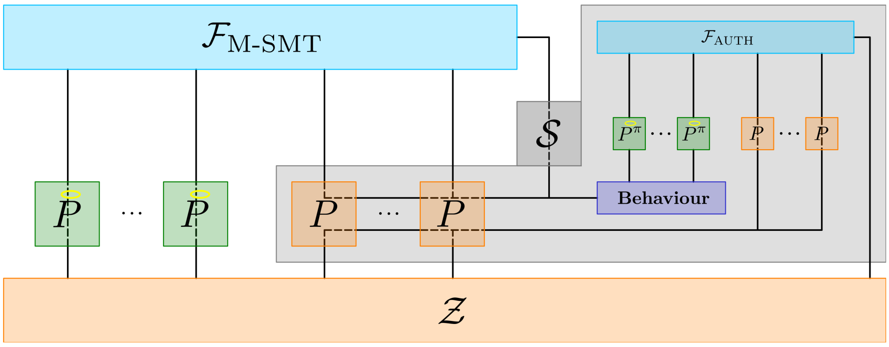

Fig. 5. Overview of Simulator $\mathcal{S}$

$\mathcal{S}_{\textbf{M-SMT}}$

Realizes:
Multi-receiver multi-message multi-sender secure message transfer with constant message size.

Parameters:

- Security parameter λ.
- IND-SB-CPA secure SBE scheme $(\texttt{gen}, \texttt{enc}, \texttt{dec})$.

In-the-head Parties:

- Functionality $\mathcal{F}_{\text{AUTH}}$. 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 the protocol $\pi_{\text{M-SMT}}^{\mathcal{F}_{\text{AUTH}}}$, 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).
- Dummy corrupted parties. Whenever the simulator is asked by the environment to call the 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 store in their states.

Behaviour:

\\ Self-communication

- Upon receiving $(\texttt{send}, \mathit{sid}, \mathit{mid}, P, P)$ from $\mathcal{F}_{\text{M-SMT}}$ to $\mathcal{A}$ for honest party P, call $\mathcal{F}_{\text{M-SMT}}$ with input $(\texttt{send ok}, \mathit{sid}, \mathit{mid})$.

\\ Message from honest to honest party

- Upon receiving $(\texttt{send}, \mathit{sid}, \mathit{mid}, S, R)$ from $\mathcal{F}_{\text{M-SMT}}$ to $\mathcal{A}$ for honest parties $S \neq R$:

- ○ Start in-the-head party S^{π} with input $(\texttt{send}, \mathit{sid}, R, 0)$ from the environment $\mathcal{Z}$.
- ○ If in-the-head party R^{π} at some point reports output $(\texttt{sent}, \mathit{sid}, S, 0)$, call $\mathcal{F}_{\text{M-SMT}}$ with input $(\texttt{send ok}, \mathit{sid}, \mathit{mid})$.[8]

\\\\ Message from honest to corrupted party

- Upon receiving $(\texttt{send}, \mathit{sid}, \mathit{mid}, S, R)$ from $\mathcal{F}_{\text{M-SMT}}$ to $\mathcal{A}$ for honest party S and corrupted party R:
  (1) Call $\mathcal{F}_{\text{M-SMT}}$ with input $(\texttt{send ok}, \mathit{sid}, \mathit{mid})$.
  (2) Receive output $(\texttt{sent}, \mathit{sid}, S, m)$ from $\mathcal{F}_{\text{M-SMT}}$ to R.
  (3) Start in-the-head party S^{π} with input $(\texttt{send}, \mathit{sid}, R, m)$ from the environment $\mathcal{Z}$.

\\\\ Message from corrupted to honest party

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

Security Theorem and Proof. The last thing left to do is to prove that under static corruption the simulator $\mathcal{S}_{\text{M-SMT}}$ does in fact achieve indistinguishability between $\pi_{\text{M-SMT}}^{\mathcal{F}_{\text{AUTH}}}$ and $\mathcal{F}_{\text{M-SMT}}$ in the $\mathcal{F}_{\text{AUTH}}$-hybrid model. To do so we will reduce this indistinguishability to the IND-SB-CPA security of the underlying SBE scheme. I.e. assuming there is an environment $\mathcal{Z}$ which can efficiently distinguish a real execution of $\pi_{\text{M-SMT}}^{\mathcal{F}_{\text{AUTH}}}$ from an ideal experiment with $\mathcal{F}_{\text{M-SMT}}$ and $\mathcal{S}_{\text{M-SMT}}$ (with non-negligible probability) we construct an adversary $\mathcal{A}_{\text{SB-CPA}}$ who can win the IND-SB-CPA game with non-negligible probability.

To achieve this let us first take a closer look at what a successfully distinguishing environment needs to do:

Remark 2. From the definition of the simulator $\mathcal{S}_{\text{M-SMT}}$ we immediately see that if an environment $\mathcal{Z}$ is able to distinguish executions of $\mathcal{F}_{\text{M-SMT}}$ and $\pi_{\text{M-SMT}}^{\mathcal{F}_{\text{AUTH}}}$, it can only do so by messages between honest parties $S \neq R$. In this case the simulator prompts its in-the-head sender S^{π} to send a message 0 to R instead of the actual message m (which the simulator does not know). The environment will therefore receive from $\mathcal{F}_{\text{AUTH}}$ (played by $\mathcal{S}_{\text{M-SMT}}$) a message

$$\Big(\texttt{send}, \mathit{sid}_{\text{AUTH}}, S, R, \big(\mathit{sid}, \texttt{enc}(pk_R, S, 0)\big)\Big)$$

[8] At this point we assume the simulator to track the protocol executions in their head so they know which mid to use. For readability purposes we refrained from introducing notation to explicitly store this.

in the ideal execution, while it receives in the protocol execution the message

$$\Big(\texttt{send}, sid_{\text{AUTH}}, S, R, \big(sid, \texttt{enc}(pk_R, S, m)\big)\Big).$$

In all other cases the simulator can perfectly mimic the protocol execution by playing the relevant parties and functionalities in its head.[9]

Let us restrict the distinguishing possibilities even more by introducing a sequence of hybrid games and showing that we only need to consider distinguishability of two consecutive hybrids:

Definition 4 (Hybrids H_k). *Let $k \in \mathbb{N}_0$ be a natural number. The hybrid H_k represents the execution set-up where almost all interactions are handled as in the real world execution of $\pi^{\mathcal{F}_{AUTH}}_{M\text{-}SMT}$. Note that Remark 2 guarantees that these are the same as in the ideal world, apart from encryptions of messages between honest parties. Now the only difference between an execution of $\pi^{\mathcal{F}_{AUTH}}_{M\text{-}SMT}$ and H_k is the following: For the first k messages m_i ($i \leq k$) between two honest parties $R_i \neq S_i$, the output from $\mathcal{F}_{AUTH}$ to the environment $\mathcal{Z}$*

$$\Big(\texttt{send}, sid_{AUTH}, S_i, R_i, \big(sid, \texttt{enc}(pk_{R_i}, S_i, 0)\big)\Big)$$

contains an encryption of zeros—as it would in the ideal execution with simulator $\mathcal{S}_{M\text{-}SMT}$—instead of an encryption of the real message m_i.
Note that H_0 is equal to the real world execution of $\pi^{\mathcal{F}_{AUTH}}_{M\text{-}SMT}$ and H_∞ (where encryptions of zeros are used for all messages m_i, $i \in \mathbb{N}$) is equal to the ideal world execution of $\mathcal{F}_{M\text{-}SMT}$ with $\mathcal{S}_{M\text{-}SMT}$.

Lemma 2. *Let there be an environment $\mathcal{Z}$ which distinguishes real and ideal world. Then there is a $\kappa \in \mathbb{N}$ and an environment $\mathcal{Z}_\kappa$ which distinguishes hybrids $H_{\kappa-1}$ and H_κ.*

Proof. By definition $\mathcal{Z}$ distinguishes executions in hybrids H_0 and H_∞. Since $\mathcal{Z}$ is PPT, there is a polynomial $p_{\mathcal{Z}}$ which bounds its runtime, i.e. $\mathcal{Z}$ takes at most $p_{\mathcal{Z}}(\lambda)$ steps. In particular $\mathcal{Z}$ can request no more than $p_{\mathcal{Z}}(\lambda)$ messages to be sent between honest parties, and hence executions of $\mathcal{Z}$ in H_∞ and H_k are the same for all $k > p_{\mathcal{Z}}(\lambda)$. Hence by transitivity of indistinguishability (here we require the chain from H_0 to H_∞ to actually be finite by the argument before), there is an $\kappa \in \mathbb{N}$ such that H_κ and $H_{\kappa-1}$ are not indistinguishable. □

With this preparatory work, we are finally ready to prove that our protocol $\pi^{\mathcal{F}_{\text{AUTH}}}_{\text{M-SMT}}$ does in fact realize secure message transfer:

Theorem 3. *Under static corruption, $\pi^{\mathcal{F}_{AUTH}}_{M\text{-}SMT}$ is a UC-realization of $\mathcal{F}_{M\text{-}SMT}$ in the $\mathcal{F}_{AUTH}$-hybrid model, if the underlying SBE scheme satisfies IND-SB-CPA security. I.e.*

$$\pi^{\mathcal{F}_{AUTH}}_{M\text{-}SMT} \geq_{UC} \mathcal{F}_{M\text{-}SMT}.$$

[9] Please convince yourself from the definition of the simulator $\mathcal{S}_{\text{M-SMT}}$ that it has all the knowledge required for simulation and that activations/outputs of $\mathcal{F}_{\text{M-SMT}}$ will actually occur at the right times.

Proof. Assume there is an environment $\mathcal{Z}$ which distinguishes between executions of $\pi_{\text{M-SMT}}^{\mathcal{F}_{\text{AUTH}}}$ and $\mathcal{F}_{\text{M-SMT}}$. By Lemma 2 there is a $\kappa \in \mathbb{N}$ such that $\mathcal{Z}$ distinguishes hybrids $H_{\kappa-1}$ and H_κ with non-negligible probability. We now construct and adversary $\mathcal{A}_{\text{SB-CPA}}$ from $\mathcal{Z}$ which has non-negligible probability to win the IND-SB-CPA game. First $\mathcal{A}_{\text{SB-CPA}}$ receives (S, pk_S, R, pk_R) from $\mathcal{C}_{\text{SB-CPA}}$. Then it starts $\mathcal{Z}$ in it's head, playing all other parties. Again by Remark 2, $\mathcal{Z}$ needs to register at least two honest parties (and send a message between them) to distinguish. For the two honest parties R and S (randomly chosen by the challenger), $\mathcal{A}_{\text{SB-CPA}}$ does not generate fresh credentials as the honest parties would do, but rather uses pk_S and pk_R from $\mathcal{C}_{\text{SB-CPA}}$.

It is no problem that $\mathcal{A}_{\text{SB-CPA}}$ does not know sk_R, sk_S. The only case they are used is when a corrupted party sends a message to R or S, i.e. when one of them receives output $(\texttt{sent}, sid_{\text{AUTH}}, P, R/S, (sid, c))$ for some corrupted party P from the functionality $\mathcal{F}_{\text{AUTH}}$. In this case $\mathcal{A}_{\text{SB-CPA}}$ promts the oracle $\mathcal{O}_{\text{SB-CPA}}$ with input (pk_S, P, c). Note that it is $P \notin \{S, R\}$. Hence $\mathcal{O}_{\text{SB-CPA}}$ by definition responds with the decryption $m := \texttt{dec}(sk_{S/R}, P, c)$ and $\mathcal{A}_{\text{SB-CPA}}$ can let the simulator call $\mathcal{F}_{\text{M-SMT}}$ with input $(\texttt{send}, sid, S/R, m)$ in the name of P as usual.

For the first $\kappa - 1$ messages which are sent between two honest parties, we report encryptions of 0 instead, when $\mathcal{Z}$ asks the adversary to see the content of the communication channel. When $\mathcal{Z}$ asks for the κ-th message m_κ to be sent, $\mathcal{A}_{\text{SB-CPA}}$ does the following:

- If m_κ is not a message from S to R, give up.
- If m_κ is to be sent from S to R, hand messages 0 and m_κ to $\mathcal{C}_{\text{SB-CPA}}$ and receive challenge c^*. Report c^* as communication channel content to $\mathcal{Z}$.

From now on, when a message m is sent between two honest parties, always report an encryption of m as channel content instead of 0 as before. When $\mathcal{Z}$ stops and reports it has run in the hybrid H_κ, report bit $b = 0$ to $\mathcal{C}_{\text{SB-CPA}}$, if $\mathcal{Z}$ decides on $H_{\kappa-1}$, report $b = 1$. □

6 Relation between IND-SB-CPA and TBE Notions

We have presented the new notion of IND-SB-CPA for SBE in Section 2, given some intuition on what this notion implies and broadened the intuitive understanding by a generic example construction in Section 3. What is still missing from the picture is a formal classification of how this notion directly relates to other security notions. To fill this gap we firstly examine the connection between IND-SB-CPA and TBE security notions in this section.

In Appendix G.2 we also look at the implications between IND-SB-CPA and classical PKE IND notions ranging from CPA to CCA2.

First note that although the notion of IND-gtag-wCCA has not been defined prior to this work it is an obvious relaxation of IND-stag-wCCA security—which was the weakest TBE notion considered so far. The proofs for the (non-)implications between IND-gtag-wCCA and IND-stag-wCCA can be found in Appendix G.1.

In this section we concentrate on the relationship between IND-SB-CPA and IND-gtag-wCCA. To compare the two notions we assume the tag space $\mathbf{T}$ considered for IND-gtag-wCCA to be equal to a set $\mathbf{P}$ of party IDs. Of course a bijection between the two is sufficient as well, but we compare the notions for tag and ID spaces of the same size. An overview is shown in Figure 6.

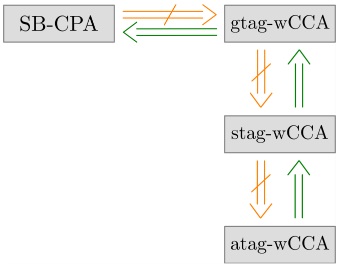

Fig. 6. Relationship to TBE Notions

Lemma 3. *IND-SB-CPA* $\Leftarrow$ *IND-gtag-wCCA.*

Proof. Let $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ be a TBE scheme. Under assumption of an efficient adversary $\mathcal{A}_{\text{SB-CPA}}$ with non-negligible probability to win the IND-SB-CPA security game, we will construct an efficient adversary $\mathcal{A}_{\text{gtag-wCCA}}$ who has the same success probability in the IND-gtag-wCCA game. An overview of the construction can be found in Figure 7.

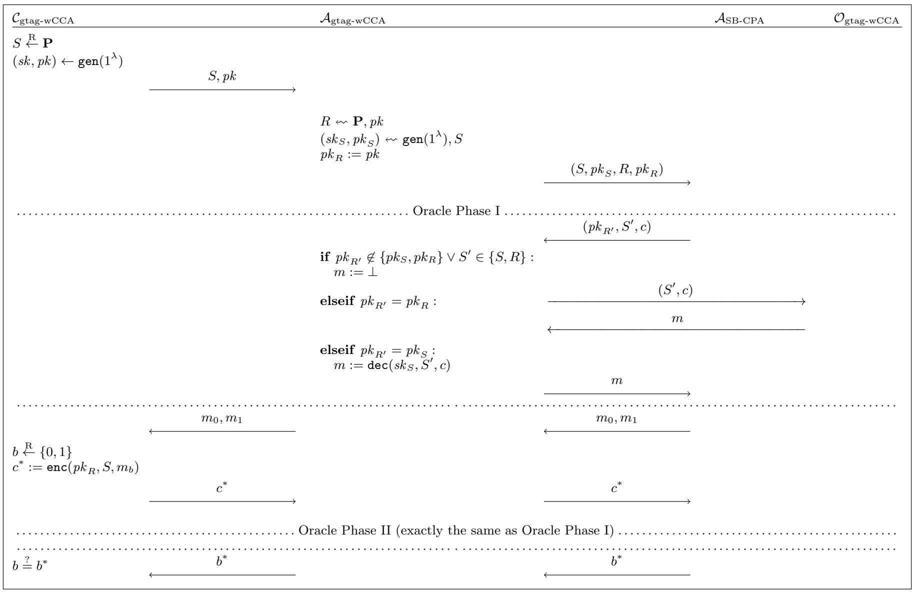
$\mathcal{C}_{\text{gtag-wCCA}}$
$\mathcal{A}_{\text{gtag-wCCA}}$
$\mathcal{A}_{\text{SB-CPA}}$
$\mathcal{O}_{\text{gtag-wCCA}}$
$S \overset{R}{\leftarrow} \mathbf{P}$
$(sk, pk) \leftarrow \texttt{gen}(1^\lambda)$
S, pk
$R \leftsquigarrow \mathbf{P}, pk$
$(sk_S, pk_S) \leftsquigarrow \texttt{gen}(1^\lambda), S$
$pk_R := pk$
(S, pk_S, R, pk_R)
Oracle Phase I
$(pk_{R'}, S', c)$
if $pk_{R'} \notin \{pk_S, pk_R\} \vee S' \in \{S, R\}$:
$m := \perp$
elseif $pk_{R'} = pk_R$:
(S', c)
m
elseif $pk_{R'} = pk_S$:
$m := \texttt{dec}(sk_S, S', c)$
m
m_0, m_1
m_0, m_1
$b \overset{R}{\leftarrow} \{0, 1\}$
$c^* := \texttt{enc}(pk_R, S, m_b)$
c^*
c^*
Oracle Phase II (exactly the same as Oracle Phase I)
$b \overset{?}{=} b^*$
b^*
b^*

Fig. 7. Reduction for IND-SB-CPA $\Leftarrow$ IND-gtag-wCCA

After being handed an ID S as the challenge tag and a public key pk, the adversary $\mathcal{A}_{\text{gtag-wCCA}}$ determines an ID R matching the public key $pk = pk_R$ and generates a key pair (sk_S, pk_S) matching the ID S. Depending on the specific scheme, these might, e.g., involve some key registration or be completely independent of one another. The IDs and public keys (S, pk_S, R, pk_R) are handed on to $\mathcal{A}_{\text{SB-CPA}}$. Any valid oracle queries $(pk_{R'}, S', c)$ from $\mathcal{A}_{\text{SB-CPA}}$ (i.e., those with $S' \notin \{S, R\}$ and $pk_{R'} \in \{pk_S, pk_R\}$) are answered in one of two ways: If pk'_R is equal to the challenge key pk_R, the query (S', c) is forwarded to $\mathcal{A}_{\text{gtag-wCCA}}$'s own oracle $\mathcal{O}_{\text{gtag-wCCA}}$. Otherwise, $\mathcal{A}_{\text{gtag-wCCA}}$ uses it's secret key sk_S to perform the decryption itself. In both cases the challenge is answered exactly like an oracle $\mathcal{O}_{\text{SB-CPA}}$ would. After forwarding the messages m_0, m_1 and the challenge ciphertext c^* between $\mathcal{A}_{\text{SB-CPA}}$ and $\mathcal{C}_{\text{gtag-wCCA}}$, the oracle phase is repeated exactly as before. Finally, the bit b^* which $\mathcal{A}_{\text{SB-CPA}}$ outputs is forwarded as well. If the adversary $\mathcal{A}_{\text{SB-CPA}}$ wins, so will $\mathcal{A}_{\text{gtag-wCCA}}$. □

Lemma 4. *IND-SB-CPA $\not\Rightarrow$ IND-gtag-wCCA.*

Proof. Let us consider the DRE-based example $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ from section 3 again. In Lemma 1 we have already shown that this scheme is IND-SB-CPA secure. To prove our current claim it remains to be shown that $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ does not satisfy IND-gtag-wCCA security. We do so by constructing an efficient adversary $\mathcal{A}_{\text{gtag-wCCA}}$ which has non-negligible probability of winning the IND-gtag-wCCA security game. Firstly the challenger $\mathcal{C}_{\text{gtag-wCCA}}$ chooses a random party ID $S \in \mathbf{P}$, generates the challenge key pair (SK_R, PK_R) and registers it for some party R. On input of S, PK_R, the adversary $\mathcal{A}_{\text{gtag-wCCA}}$ generates a fresh key pair (SK_S, PK_S), and register this key pair with $\mathcal{F}_{\text{KRK}}$ in the name of S. Now the adversary chooses random messages $m_0 \neq m_1$ for the challenge and receives $c^* = \texttt{Enc}(PK_R, S, m_b)$. Due to DRE soundness the adversary can now decrypt the challenge as $m_b = \texttt{Dec}(SK_S, R, c^*)$ and win the IND-gtag-wCCA game with probability one. □

Although this proof is instructing for the intuitive understanding of SBE schemes since it relies on the fact that there is a connection between tags and party keys, it also relies on the party whose ID is randomly chosen as the challenge tag to be corruptible by the adversary. I.e. the adversary needs to be able to register keys for this party. Due to this caveat let us give a second proof of the lemma:

Proof (Alternative version). Let $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ be an IND-SB-CPA secure SBE scheme. We use this to construct an SBE/TBE scheme $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ which is still IND-SB-CPA secure but does not satisfy IND-gtag-wCCA security:

$$\begin{aligned}\texttt{Gen} &:= \texttt{gen}\\ \texttt{Enc} &:= \texttt{enc}\\ \texttt{Dec}(sk, S, c) &:= \begin{cases}\texttt{dec}(sk, S, c)||sk & , sk = sk_S\\ \texttt{dec}(sk, S, c)||0\cdots 0 & , \text{ else.}\end{cases}\end{aligned}$$

It is obvious that this modified scheme does still satisfy IND-SB-CPA security, as we have $(\texttt{Gen}, \texttt{Enc}) = (\texttt{gen}, \texttt{enc})$ everywhere and $\texttt{Dec} = \texttt{dec}$ on the domain where $\mathcal{O}_{\text{SB-CPA}}$ answers queries. It is not, however, IND-gtag-wCCA secure, as any adversary can query $\mathcal{O}_{\text{gtag-wCCA}}$ with input (R, c) where R is the party ID corresponding to challenge key pk_R and c is an arbitrary ciphertext. The oracle will hand back sk_R which can be used to decrypt the challenge ciphertext c^* and win the security game every time. □

7 Conclusion

In this work we have introduced the concept of sender-binding encryption and developed the corresponding new security notion of IND-SB-CPA. We showed IND-SB-CPA security to be sufficient for UC-realizing secure message transfer (SMT) when combined with authenticated channels. Furthermore the direct implication from Section 6 and generic transformations from Appendix E show that it is currently the weakest known notion with this property. Additionally we provided a generic transformation for IND-SB-CPA via IND-CPA secure double receiver encryption (DRE) in conjunction with key registration with knowledge. In particular this construction from DRE yields an efficient practical instantiation based on McEliece in the standard model.

For future work we see several directions to further this line of research. Although we know IND-SB-CPA to be weaker than prior notions which realize SMT via authenticated channels, it remains to be shown whether it constitutes the weakest possible notion to do so. It is also far from obvious that our current practical constructions are the most efficient to satisfy IND-SB-CPA security. More effort in this direction might prove fruitful as well.

Acknowledgments. We thank Björn Kaidel for fruitful initial discussions. We thank the PKC 2022 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. Badertscher, C., Maurer, U., Portmann, C., and Rito, G.: Revisiting (R)CCA Security and Replay Protection. IACR Cryptol. ePrint Arch. **2020**, 177 (2020). https://eprint.iacr.org/2020/177
2. Barak, B., Canetti, R., Nielsen, J.B., and Pass, R.: Universally Composable Protocols with Relaxed Set-Up Assumptions. In: focs04ed (ed.) focs04name, pp. 186–195. focspub, focs04addr (2004). DOI: 10.1109/FOCS.2004.71
3. Bellare, M., and Palacio, A.: Towards Plaintext-Aware Public-Key Encryption without Random Oracles. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, pp. 48–62. Springer, Heidelberg (2004). DOI: 10.1007/978-3-540-30539-2_4
4. Bellare, M., and Rogaway, P.: Optimal Asymmetric Encryption. In: Santis, A.D. (ed.) EUROCRYPT'94. LNCS, pp. 92–111. Springer, Heidelberg (1995). DOI: 10.1007/BFb0053428

5. Bernstein, D.J., Lange, T., and Peters, C.: Attacking and Defending the McEliece Cryptosystem. In: Buchmann, J., and Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 31–46. Springer, Heidelberg (2008). DOI: 10.1007/978-3-540-88403-3_3. https://doi.org/10.1007/978-3-540-88403-3%5C_3
6. Bogos, S., Tramèr, F., and Vaudenay, S.: On Solving Lpn using BKW and Variants. IACR Cryptol. ePrint Arch. (2015). http://eprint.iacr.org/2015/049
7. Boneh, D., Canetti, R., Halevi, S., and Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. SIAM Journal on Computing **36**(5), 1301–1328 (2007)
8. Boneh, D., and Franklin, M.K.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, pp. 213–229. Springer, Heidelberg (2001). DOI: 10.1007/3-540-44647-8_13
9. Boneh, D., and Katz, J.: Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, pp. 87–103. Springer, Heidelberg (2005). DOI: 10.1007/978-3-540-30574-3_8
10. Boyen, X., Izabachène, M., and Li, Q.: A Simple and Efficient CCA-Secure Lattice KEM in the Standard Model. In: Galdi, C., and Kolesnikov, V. (eds.) SCN 20. LNCS, pp. 321–337. Springer, Heidelberg (2020). DOI: 10.1007/978-3-030-57990-6_16
11. Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and Thayer, R.: OpenPGP Message Format. RFC **4880**, 1–90 (2007). DOI: 10.17487/RFC4880. https://doi.org/10.17487/RFC4880
12. Canetti, R.: *Security and Composition of Multi-party Cryptographic Protocols*, Cryptology ePrint Archive, Report 1998/018 (1998). https://eprint.iacr.org/1998/018. 1998
13. Canetti, R.: Universally Composable Security: A New Paradigm for Cryptographic Protocols. In: 42nd FOCS, pp. 136–145. IEEE Computer Society Press (2001). DOI: 10.1109/SFCS.2001.959888
14. Canetti, R., Goldreich, O., and Halevi, S.: *The Random Oracle Methodology, Revisited*, Cryptology ePrint Archive, Report 1998/011 (1998). https://eprint.iacr.org/1998/011. 1998
15. Canetti, R., Halevi, S., and Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., and Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, pp. 207–222. Springer, Heidelberg (2004). DOI: 10.1007/978-3-540-24676-3_13
16. Canetti, R., Krawczyk, H., and Nielsen, J.B.: Relaxing Chosen-Ciphertext Security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, pp. 565–582. Springer, Heidelberg (2003). DOI: 10.1007/978-3-540-45146-4_33
17. Canetti, R., and Rabin, T.: Universal Composition with Joint State. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, pp. 265–281. Springer, Heidelberg (2003). DOI: 10.1007/978-3-540-45146-4_16
18. Cheng, H., Li, X., Qian, H., and Yan, D.: Simpler CCA Secure PKE from LPN Problem Without Double-Trapdoor. In: Naccache, D., Xu, S., Qing, S., Samarati, P., Blanc, G., Lu, R., Zhang, Z., and Meddahi, A. (eds.) ICICS 18. LNCS, pp. 756–766. Springer, Heidelberg (2018). DOI: 10.1007/978-3-030-01950-1_46
19. Chow, S.S.M., Franklin, M.K., and Zhang, H.: Practical Dual-Receiver Encryption - Soundness, Complete Non-malleability, and Applications. In: Benaloh, J. (ed.) CT-RSA 2014. LNCS, pp. 85–105. Springer, Heidelberg (2014). DOI: 10.1007/978-3-319-04852-9_5
20. 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**(1), 167–226 (2003)

21. Damgård, I., and Park, S.: Is Public-Key Encryption Based on LPN Practical? IACR Cryptol. ePrint Arch. (2012). http://eprint.iacr.org/2012/699
22. Das, A., Dutta, S., and Adhikari, A.: Indistinguishability against Chosen Ciphertext Verification Attack Revisited: The Complete Picture. In: Susilo, W., and Reyhanitabar, R. (eds.) ProvSec 2013. LNCS, pp. 104–120. Springer, Heidelberg (2013). DOI: 10.1007/978-3-642-41227-1_6
23. Diament, T., Lee, H.K., Keromytis, A.D., and Yung, M.: The Dual Receiver Cryptosystem and Its Applications. In: Atluri, V., Pfitzmann, B., and McDaniel, P. (eds.) ACM CCS 2004, pp. 330–343. ACM Press (2004). DOI: 10.1145/1030083.1030128
24. Diffie, W., and Hellman, M.E.: New Directions in Cryptography. IEEE Transactions on Information Theory **22**(6), 644–654 (1976)
25. Döttling, N., Dowsley, R., Müller-Quade, J., and Nascimento, A.C.A.: *A CCA2 Secure Variant of the McEliece Cryptosystem*, Cryptology ePrint Archive, Report 2008/468 (2008). https://eprint.iacr.org/2008/468. 2008
26. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., and Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: Nguyen, P.Q., and Pointcheval, D. (eds.) PKC 2010. LNCS, pp. 279–295. Springer, Heidelberg (2010). DOI: 10.1007/978-3-642-13013-7_17
27. Gentry, C., Peikert, C., and Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., and Dwork, C. (eds.) 40th ACM STOC, pp. 197–206. ACM Press (2008). DOI: 10.1145/1374376.1374407
28. Goldwasser, S., and Micali, S.: Probabilistic Encryption. Journal of Computer and System Sciences **28**(2), 270–299 (1984)
29. Herzog, J., Liskov, M., and Micali, S.: Plaintext Awareness via Key Registration. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, pp. 548–564. Springer, Heidelberg (2003). DOI: 10.1007/978-3-540-45146-4_32
30. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., and Rabin, T. (eds.) TCC 2006. LNCS, pp. 581–600. Springer, Heidelberg (2006). DOI: 10.1007/11681878_30
31. 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, Heidelberg (2014). DOI: 10.1007/978-3-642-54631-0_1
32. Kiltz, E., Mohassel, P., and O'Neill, A.: Adaptive Trapdoor Functions and Chosen-Ciphertext Security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, pp. 673–692. Springer, Heidelberg (2010). DOI: 10.1007/978-3-642-13190-5_34
33. Knorr, J.-M.: Abstreitbare Nachrichtenauthentifikation mit Post-Quanten-Kryptographie (in German). Karlsruhe Institute of Technology (2016)
34. Liu, Y., Zhang, D., Deng, Y., and Li, B.: (Identity-based) dual receiver encryption from lattice-based programmable hash functions with high min-entropy. Cybersecur. **2**(1), 18 (2019). DOI: 10.1186/s42400-019-0034-y. https://doi.org/10.1186/s42400-019-0034-y
35. Liu, Y., Wang, L., Shen, X., and Li, L.: New Constructions of Identity-Based Dual Receiver Encryption from Lattices. Entropy **22**(6), 599 (2020). DOI: 10.3390/e22060599. https://doi.org/10.3390/e22060599
36. 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, Heidelberg (2004). DOI: 10.1007/978-3-540-24638-1_10

37. Micciancio, D., and Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., and Johansson, T. (eds.) EUROCRYPT 2012. LNCS, pp. 700–718. Springer, Heidelberg (2012). DOI: 10.1007/978-3-642-29011-4_41
38. Noh, G., Hong, D., Kwon, J.O., and Jeong, I.R.: A Strong Binding Encryption Scheme from Lattices for Secret Broadcast. IEEE Commun. Lett. **16**(6), 781–784 (2012). DOI: 10.1109/LCOMM.2012.041112.112495. https://doi.org/10.1109/LCOMM.2012.041112.112495
39. Nojima, R., Imai, H., Kobara, K., and Morozov, K.: Semantic security for the McEliece cryptosystem without random oracles. Des. Codes Cryptogr. **49**(1-3), 289–305 (2008). DOI: 10.1007/s10623-008-9175-9. https://doi.org/10.1007/s10623-008-9175-9
40. Pandey, S.K., Sarkar, S., and Jhanwar, M.P.: Relaxing IND-CCA: Indistinguishability against Chosen Ciphertext Verification Attack. In: Bogdanov, A., and Sanadhya, S.K. (eds.) SPACE 2012. LNCS, vol. 7644, pp. 63–76. Springer, Heidelberg (2012). DOI: 10.1007/978-3-642-34416-9_5. https://doi.org/10.1007/978-3-642-34416-9%5C_5
41. Peikert, C., and Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., and Dwork, C. (eds.) 40th ACM STOC, pp. 187–196. ACM Press (2008). DOI: 10.1145/1374376.1374406
42. Rackoff, C., and Simon, D.R.: Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO'91. LNCS, pp. 433–444. Springer, Heidelberg (1992). DOI: 10.1007/3-540-46766-1_35
43. Rill, J.: Towards Applying Cryptographic Security Models to Real-World Systems. Karlsruhe Institute of Technology, Germany (2020)
44. Rosen, A., and Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC 2009. LNCS, pp. 419–436. Springer, Heidelberg (2009). DOI: 10.1007/978-3-642-00457-5_25
45. Schaad, J., Ramsdell, B., and Turner, S.: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification. RFC **8551**, 1–63 (2019). DOI: 10.17487/RFC8551. https://doi.org/10.17487/RFC8551
46. Shoup, V.: *A Proposal for an ISO Standard for Public Key Encryption*, Cryptology ePrint Archive, Report 2001/112 (2001). https://eprint.iacr.org/2001/112. 2001
47. Unger, N., and Goldberg, I.: Improved Strongly Deniable Authenticated Key Exchanges for Secure Messaging. PoPETs **2018**(1), 21–66 (2018)
48. Yu, Y., and Zhang, J.: Cryptography with Auxiliary Input and Trapdoor from Constant-Noise LPN. In: Robshaw, M., and Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 214–243. Springer, Heidelberg (2016). DOI: 10.1007/978-3-662-53018-4_9. https://doi.org/10.1007/978-3-662-53018-4%5C_9
49. Zhang, D., Zhang, K., Li, B., Lu, X., Xue, H., and Li, J.: Lattice-Based Dual Receiver Encryption and More. In: Susilo, W., and Yang, G. (eds.) ACISP 18. LNCS, pp. 520–538. Springer, Heidelberg (2018). DOI: 10.1007/978-3-319-93638-3_30

A Notations and Abbreviations

This appendix can be used to look up all notations and abbreviations employed throughout this paper.

A.1 Notations

$\xleftarrow{R}$	Uniformly randomly drawn from
$\hookrightarrow$	Output
$\geq_{\mathrm{UC}}$	Securely UC-realizes
$\perp$	Invalid/failed
$\mathcal{A}$	Adversary
Adv	Advantage
aux	Auxiliary input/output
b	Bit from $\{0, 1\}$
$\mathcal{C}$	Challenger
c	Ciphertext
c^*	Challenge ciphertext
c	Vectors
$\mathtt{dec}/\mathtt{Dec}$	Decryption algorithm
$\mathcal{E}$	Encryption scheme
$\mathtt{enc}/\mathtt{Enc}$	Encryption algorithm
Exp	Experiment
$\mathtt{ext}$	Key extraction algorithm
$\mathcal{F}$	Ideal functionality
$\mathtt{f}_{\mathrm{ID}}/\mathtt{F}_{\mathrm{ID}}$	ID function
$\mathtt{f}_{\mathrm{Key}}/\mathtt{F}_{\mathrm{Key}}$	Boolean key function
$\mathtt{f}_{\mathbf{PK}}$	Key function $sk \mapsto pk$
G	Matrices
$\mathtt{gen}/\mathtt{Gen}$	Key generation algorithm
goal	Goal of the adversary
id/ID	Protocol party ID
$\mathbf{ID}$	Set of all IDs
$\mathtt{init}$	Asking to initialize
$\mathtt{inited}$	Initialization done
k	Binary key length
λ	Security parameter
l	Message length
l'	Ciphertext length
m	Message
$\mathbf{M}$	Message space
$\mathtt{message}$	$\mathcal{F}$ message variable
mid	Message ID
$\mathbf{MID}$	Set of all message IDs
mpk	IBE master public key
msk	IBE master secret key
n	Security parameter for McEliece
$\mathcal{O}$	Oracle
π	Protocol
$\pi^{\mathcal{F}_{\mathrm{AUTH}}}_{\text{M-SMT}}$	M-SMT protocol
P	Party
$\mathbf{P}$	Set of all parties
$\mathbb{P}$	Probability
pk/PK	Public key
$\mathbf{PK}$	Set of all public keys
pow	Power of the adversary
pr	Boolean prefix function
R	Receiver
$\mathbf{R}$	Set of all registered parties
$\mathtt{receiver}$	Message *receiver*
$\mathtt{register}$	Asking to be registered
$\mathtt{register\ ok}$	Registration allowed
$\mathtt{registered}$	Registration done
$\mathtt{retrieve}$	Asking to retrieve credentials
$\mathtt{retrieve\ ok}$	Retrieval allowed
$\mathtt{retrieved}$	Retrieval done
$resp$	Oracle response
S	Sender
$\mathcal{S}$	Simulator
$\mathcal{S}_{\text{M-SMT}}$	Simulator for $\pi^{\mathcal{F}_{\mathrm{AUTH}}}_{\text{M-SMT}}$
scp	Scope of adversary's power
$\mathtt{send}$	Asking to send message
$\mathtt{send\ ok}$	Transmission allowed
$\mathtt{sent}$	Message sent
set	Setting of security game
sid	Session ID
$\mathbf{SID}$	Set of all session IDs
sk/SK	Secret key
$\mathbf{SK}$	Set of all secret keys
$\mathtt{stray}$	Message *stray*
$test$	Special response of $\mathcal{O}_{\mathrm{RCCA}}$
usk	User secret key
$\mathcal{Z}$	Environment

A.2 Abbreviations

CCA chosen ciphertext attack
CCA1 non-adaptive chosen ciphertext attack
CCA1.5 chosen ciphertext decryption/verification attack
CCA2 adaptive chosen ciphertext attack
CCVA chosen ciphertext verification attack
CCVA1 non-adaptive chosen ciphertext verification attack
CCVA2 adaptive chosen ciphertext verification attack
CPA chosen plaintext attack
DAKEZ Deniable authenticated key exchange with zero-knowledge
DRE double receiver encryption
IBE identity based encryption
IF ideal functionality
IND indistinguishability
IND-CCA2 indistinguishability under adaptive chosen ciphertext attack
IND-CCVA1 indistinguishability under non-adaptive chosen ciphertext verification attack
IND-CPA indistinguishability under chosen plaintext attack
IND-gtag-wCCA indistinguishability under given-tag weakly chosen ciphertext attack
IND-stag-wCCA indistinguishability under selective-tag weakly chosen ciphertext attack
stag-wCCA selective-tag weakly chosen ciphertext attack
IND-RCCA indistinguishability under replayable chosen ciphertext attack
IND-sID-CPA indistinguishability under selective identity chosen plaintext attack
IND-SB-CPA indistinguishability under sender-binding chosen plaintext attack
IND-atag-wCCA indistinguishability under adaptive-tag weakly chosen ciphertext attack
atag-wCCA adaptive-tag weakly chosen ciphertext attack
KRK key registration with knowledge
LPN learning parity with noise
LPNDP learning parity with noise decisional problem
LWE learning with errors
OTR Off-the-Record
PA plaintext awareness
PKE public key encryption
PKI public key infrastructure
PPT probabilistic polynomial time
PQC post-quantum cryptography
RCCA replayable chosen ciphertext attack
ROM random oracle model
RPA registration-based plaintext awareness
SBE sender-binding encryption
SID session ID
SMT secure message transfer
TBE tag-based encryption
TM turing machine
UC universal composability
XZDH Extended Zero-knowledge Diffie-Hellman

B Encryption Schemes and Game-Based Security Notions

In this appendix we recapitulate different types of encryption schemes and the regarding game-based security notions. We start with the lesser known chosen ciphertext verification attack (CCVA) and RCCA notions for traditional PKE schemes before we go on to notions for special PKE schemes like TBE, IBE, and DRE.

B.1 Classic PKE Notions

CCVA and CCA1.5 The notion of CCVA has been introduced in [40] and further differentiated in [22]. The following definitions are based on the latter: A verification oracle decides on input of a ciphertext and public key whether or not the ciphertext is valid, i.e.,

$$\mathcal{O}_{\text{CCVA}}(pk, c) = \begin{cases} \mathsf{false}, & \mathtt{dec}(sk, c) = \perp \\ \mathsf{true}, & \text{otherwise.} \end{cases}$$

Analogous to chosen ciphertext attack (CCA), non-adaptive chosen ciphertext verification attack (CCVA1) describes security notions where access to the oracle $\mathcal{O}_{\text{CCVA}}$ is only permitted in the first oracle phase while with adaptive chosen ciphertext verification attack (CCVA2) access is granted in both phases. Note that no other oracle access (e.g. decryption) is given and that contrary to CCA, puncturing is not necessary with CCVA as the challenge ciphertext will always result in true and this does not provide the adversary with any new information.

CCA1.5 is defined as a mix of non-adaptive chosen ciphertext attack (CCA1) and CCVA2: In the first phase the adversary is provided with a decryption oracle while in the second phase only verification queries are permitted.

RCCA Notions Canetti, Krawczyk, and Nielsen introduce in [16] the security definition of IND-RCCA with the following game.

Let $\mathcal{E} = (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ be an encryption scheme with message space $\mathbf{M}$. Let $\mathcal{A}$ be an adversary and λ the security parameter.

Key generation: Run $(pk, sk) \leftarrow \mathtt{gen}(1^\lambda)$, and give pk to $\mathcal{A}$.

First decryption stage: When $\mathcal{A}$ queries $(\mathtt{ciphertext}, c)$, compute $m = \mathtt{dec}(c, sk)$ and give m to $\mathcal{A}$.

Encryption stage: When $\mathcal{A}$ queries $(\mathtt{test\ message}, m_0, m_1)$ with $m_0, m_1 \in \mathbf{M}$, and $m_0 \neq m_1$, compute $c^* = \mathtt{enc}(m_b, pk)$ where $b \overset{\text{R}}{\leftarrow} \{0, 1\}$, and give c^* to $\mathcal{A}$. (This step is performed only once.)

Second decryption stage: When $\mathcal{A}$ queries $(\mathtt{ciphertext}, c)$ after c^* is defined, compute $m = \mathtt{dec}(c, sk)$. If $m \in \{m_0, m_1\}$ then give $\mathtt{test}$ to $\mathcal{A}$. Otherwise, give m to $\mathcal{A}$.

Guessing stage: When $\mathcal{A}$ outputs $(\texttt{guess}, b')$, the outcome of the game is determined as follows. If $b' = b$ then $\mathcal{A}$ wins the game. Otherwise, $\mathcal{A}$ loses the game.

Definition 5 (IND-RCCA). *An encryption scheme $\mathcal{E}$ is said to be IND-RCCA secure if any PPT adversary $\mathcal{A}$ wins the IND-RCCA game above with probability that is at most negligibly more than one half.*

B.2 Tag-based encryption (TBE)

A TBE scheme $\mathcal{E}$ with message space $\mathbf{M}$ and tag space $\mathbf{T}$ consists of three PPT algorithms $(\texttt{gen}, \texttt{enc}, \texttt{dec})$. Let λ be the security parameter.

$$\begin{aligned} \texttt{gen} &: (1^\lambda) \mapsto (sk, pk) \\ \texttt{enc} &: (pk, t, m) \mapsto c \\ \texttt{dec} &: (sk, t, c) \mapsto m \in \mathbf{M} \cup \{\perp\} \end{aligned}$$

On input of the security parameter λ, the key generation algorithm $\texttt{gen}$ generates a pair of secret and public key sk and pk. The encryption algorithm $\texttt{enc}$ produces a ciphertext c by input of a public key pk, a tag t, and a message m. The decryption algorithm $\texttt{dec}$ takes a secret key sk, a tag t and a ciphertext c as input, outputting a message m (or possibly $\perp$ if decryption fails). The set of these algorithms corresponds to the classical PKE interface with the sole difference that a tag is required to encrypt messages as well as decrypt ciphertexts. Correspondingly, a TBE scheme is expected to fulfill the notion of correctness, i.e. that whenever $(sk, pk) \leftarrow \texttt{gen}(1^\lambda)$, then for all $t \in \mathbf{T}$

$$m = \texttt{dec}(sk, t, \texttt{enc}(pk, t, m)).$$

One special security notion for TBE schemes is selective-tag weakly chosen ciphertext attack (stag-wCCA) security introduced by Kiltz [30].

Definition 6 (stag-wCCA TBE security [30]). *Consider the experiment* $\mathsf{Exp}^{\text{stag-wCCA}}_{\text{TBE},\mathcal{A}}$ *from Figure 8 for a TBE scheme* $\mathcal{E} = (\texttt{gen}, \texttt{enc}, \texttt{dec})$ *and an adversary $\mathcal{A}$. In this experiment, $\mathcal{A}$ is not allowed to query the oracle on inputs (t^*, c) including the challenge tag in step (4). Furthermore, $\mathcal{A}$ must output m_0, m_1 of equal length. Let*

$$\mathsf{Adv}^{\text{stag-wCCA}}_{\mathcal{E},\mathcal{A}}(\lambda) := \mathbb{P}\big[\mathsf{Exp}^{\text{stag-wCCA}}_{\mathcal{E},\mathcal{A}} = 1\big] - \frac{1}{2}.$$

We say that $\mathcal{E} = (\texttt{gen}, \texttt{enc}, \texttt{dec})$ selective-tag weakly secure against chosen ciphertext attacks, *e.g. is stag-wCCA secure, if* $\mathsf{Adv}^{\text{stag-wCCA}}_{\mathcal{E},\mathcal{A}}(\lambda)$ *is negligible for all PPT $\mathcal{A}$.*

A more general security notion for TBE schemes has been introduced by MacKenzie, Reiter, and Yang, which can be classified as adaptive-tag weakly chosen ciphertext attack (atag-wCCA) security.

```
$\mathsf{Exp}^{\text{stag-wCCA}}_{\text{TBE},\mathcal{A}}$

(1) $(t^*, st_0) \leftarrow \mathcal{A}(1^\lambda, init)$
    $(sk, pk) \leftarrow \mathtt{gen}(1^\lambda)$
(2) $(st, m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{dec}(\cdot,\cdot)}(find, st_0, pk)$
(3) $b \overset{R}{\leftarrow} \{0,1\}$
    $c^* \leftarrow \mathtt{enc}(pk, t^*, m_b)$
(4) $b^* \leftarrow \mathcal{A}^{\mathsf{dec}(\cdot,\cdot)}(guess, st, c^*)$
(5) Return 1 if $b = b^*$, else return 0
```

Fig. 8. The stag-wCCA TBE Game.

Definition 7 (atag-wCCA TBE security [30, 36]). *Consider the experiment* $\mathsf{Exp}^{\text{atag-wCCA}}_{\text{TBE},\mathcal{A}}$ *from Figure 9 for a TBE scheme* $\mathcal{E} = (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ *and an adversary* $\mathcal{A}$*. In this experiment,* $\mathcal{A}$ *is not allowed to query the oracle on inputs* (t^*, c) *including the challenge tag in step (4). Furthermore,* $\mathcal{A}$ *must output* m_0, m_1 *of equal length. Let*

$$\mathsf{Adv}^{\text{atag-wCCA}}_{\mathcal{E},\mathcal{A}}(\lambda) := \mathbb{P}\big[\mathsf{Exp}^{\text{atag-wCCA}}_{\mathcal{E},\mathcal{A}} = 1\big] - \frac{1}{2}.$$

We say that $\mathcal{E} = (\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ adaptive-tag weakly secure against chosen ciphertext attacks*, e.g. is atag-wCCA secure, if* $\mathsf{Adv}^{\text{atag-wCCA}}_{\mathcal{E},\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$*.*

```
$\mathsf{Exp}^{\text{atag-wCCA}}_{\text{TBE},\mathcal{A}}$

(1) $(sk, pk) \leftarrow \mathtt{gen}(1^\lambda)$
(2) $(t^*, st, m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{dec}(\cdot,\cdot)}(find, pk)$
(3) $b \overset{R}{\leftarrow} \{0,1\}$
    $c^* \leftarrow \mathtt{enc}(pk, t^*, m_b)$
(4) $b^* \leftarrow \mathcal{A}^{\mathsf{dec}(\cdot,\cdot)}(guess, st, c^*)$
(5) Return 1 if $b = b^*$, else return 0
```

Fig. 9. The atag-wCCA TBE Game.

The most general security notion in this line is based on [46], which can be declared as *full* CCA security for TBE schemes. We achieve *full* CCA from atag-wCCA by giving the adversary the possibility to query any tuple (t, c), even including $t = t^*$, as long as $c \neq c^*$.

The line of this three security notions form a hierachy, where *full CCA* is the strongest notion and stag-wCCA represents the weakest notion. All of these security notions are still stronger than our security notion of gtag-wCCA security for TBE. We remark, that gtag-wCCA is streakly weaker than the prior mentioned security notions as shown in Appendix G.1.

B.3 Identity based encryption (IBE)

The first practical IBE scheme was proposed by Boneh and Franklin in [8]. An IBE scheme $\mathcal{E}$ with message space $\mathbf{M}$ and identity space $\mathbf{ID}$ consists of four PPT algorithms $(\texttt{gen}, \texttt{ext}, \texttt{enc}, \texttt{dec})$. Let λ be the security parameter.

$$\begin{aligned}
\texttt{gen} &: (1^\lambda) \mapsto (\mathit{mpk}, \mathit{msk}) \\
\texttt{ext} &: (\mathit{msk}, \mathit{id}) \mapsto \mathit{usk}_{\mathit{id}} \\
\texttt{enc} &: (\mathit{mpk}, \mathit{id}, m) \mapsto c \\
\texttt{dec} &: (\mathit{usk}_{\mathit{id}}, \mathit{id}, c) \mapsto m \in \mathbf{M} \cup \{\bot\}
\end{aligned}$$

For correctness, we require that for all $(\mathit{mpk}, \mathit{msk}) \leftarrow \texttt{gen}(1^\lambda)$, all $\mathit{id} \in \mathbf{ID}$, all $m \in \mathbf{M}$, all $c \leftarrow \texttt{enc}(\mathit{mpk}, \mathit{id}, m)$, and all $\mathit{usk}_{\mathit{id}} \leftarrow \texttt{ext}(\mathit{msk}, \mathit{id})$, we always have $\texttt{dec}(\mathit{usk}_{\mathit{id}}, \mathit{id}, c) = m$.

Definition 8 (IND-ID-CPA). *Consider the experiment* $\mathsf{Exp}^{\mathit{ind\text{-}id\text{-}cpa}}_{\mathcal{E},\mathcal{A}}$ *from Figure 10 for an IBE scheme* $\mathcal{E} = (\texttt{gen}, \texttt{ext}, \texttt{enc}, \texttt{dec})$ *and an adversary* $\mathcal{A}$*. In this experiment,* $\mathcal{A}$ *is not allowed to output an identity id that it has queried to its* $\texttt{ext}$ *oracle, or to later query id to* $\texttt{ext}$*. Furthermore,* $\mathcal{A}$ *must output* m_0, m_1 *of equal length. Let*

$$\mathsf{Adv}^{\mathit{ind\text{-}id\text{-}cpa}}_{\mathcal{E},\mathcal{A}}(\lambda) := \mathbb{P}\big[\mathsf{Exp}^{\mathit{ind\text{-}id\text{-}cpa}}_{\mathcal{E},\mathcal{A}} = 1\big] - \frac{1}{2}.$$

```
$\mathsf{Exp}^{\text{ind-id-cpa}}_{\mathcal{E},\mathcal{A}}$

$(\mathit{mpk}, \mathit{msk}) \leftarrow \texttt{gen}(1^\lambda)$
$(\mathit{st}, \mathit{id}, m_0, m_1) \leftarrow \mathcal{A}^{\texttt{ext}(\mathit{msk},\cdot)}(\mathit{mpk})$
$b \overset{R}{\leftarrow} \{0, 1\}$
$c^* \leftarrow \texttt{enc}(\mathit{mpk}, \mathit{id}, m_b)$
$b' \leftarrow \mathcal{A}^{\texttt{ext}(\mathit{msk},\cdot)}(\mathit{st}, c^*)$
Return 1 if $b' = b$, else return 0
```

Fig. 10. The IND-ID-CPA Game.

We say that $\mathcal{E} = (\mathtt{gen}, \mathtt{ext}, \mathtt{enc}, \mathtt{dec})$ *has indistinguishable ciphertexts under chosen-plaintext attacks, e.g. is IND-ID-CPA secure, if* $\mathsf{Adv}^{\textit{ind-id-cpa}}_{\mathcal{E},\mathcal{A}}$ *is negligible for all PPT* $\mathcal{A}$*.*

We further consider a weaker security notion introduced in [15], where the adversary has to specify the identity they want to attack at the beginning of the experiment.

Definition 9 (IND-sID-CPA). *Consider the experiment* $\mathsf{Exp}^{\textit{ind-sid-cpa}}_{\mathcal{E},\mathcal{A}}$ *from Figure 11 for an IBE scheme* $\mathcal{E} = (\mathtt{gen}, \mathtt{ext}, \mathtt{enc}, \mathtt{dec})$ *and a PPT algorithm* $\mathcal{A}$*. In this experiment,* $\mathcal{A}$ *is not allowed to query id to* $\mathtt{ext}$ *and has to output* m_0, m_1 *of equal length. Let*

```
                    $\mathsf{Exp}^{\text{ind-sid-cpa}}_{\mathcal{E},\mathcal{A}}$

(1) $(st, id) \leftarrow \mathcal{A}(1^\lambda)$
(2) $(mpk, msk) \leftarrow \mathtt{gen}(1^\lambda)$
(3) $(st', m_0, m_1) \leftarrow \mathcal{A}^{\mathtt{ext}(msk,\cdot)}(st, mpk)$
(4) $b \xleftarrow{R} \{0,1\}$
(5) $c^* \leftarrow \mathtt{enc}(mpk, id, m_b)$
(6) $b' \leftarrow \mathcal{A}^{\mathtt{ext}(msk,\cdot)}(st', c^*)$
(7) Return 1 if $b' = b$, else return 0
```

Fig. 11. The IND-sID-CPA Game.

$$\mathsf{Adv}^{\textit{ind-sid-cpa}}_{\mathcal{E},\mathcal{A}}(\lambda) := \mathbb{P}\big[\mathsf{Exp}^{\textit{ind-sid-cpa}}_{\mathcal{E},\mathcal{A}} = 1\big] - \frac{1}{2}.$$

We say that the IBE scheme $\mathcal{E} = (\mathtt{gen}, \mathtt{ext}, \mathtt{enc}, \mathtt{dec})$ *has indistinguishable ciphertexts under selective identity chosen-plaintext attacks, e.g. is IND-sID-CPA secure, if* $\mathsf{Adv}^{\textit{ind-sid-cpa}}_{\mathcal{E},\mathcal{A}}$ *is negligible for all PPT algorithms* $\mathcal{A}$*.*

B.4 Double receiver encryption (DRE)

A DRE scheme consists of three PPT algorithms $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ and the function $\mathtt{f}_{\text{Key}}$, which checks if the key pair (sk, pk) is well-formed. Let λ be the security parameter.

$$\begin{aligned}
\texttt{gen} &: 1^\lambda \mapsto (sk, pk)\\
\texttt{enc} &: (pk_1, pk_2, m) \mapsto c\\
\texttt{dec} &: (sk_i, pk_1, pk_2, c) \mapsto m \text{ where } i \in \{1, 2\}\\
\texttt{f}_{\text{Key}} &: (sk, pk) \mapsto \begin{cases}\mathsf{true}\\ \mathsf{false}.\end{cases}
\end{aligned}$$

Definition 10 (IND-CPA DRE). *A DRE scheme is said to be indistinguishable under chosen plaintext attack, e.g. is IND-CPA secure, if any PPT algorithm $\mathcal{A}$ wins the IND-CPA DRE game in Figure 12 with probability that is at most negligibly more than one half.*

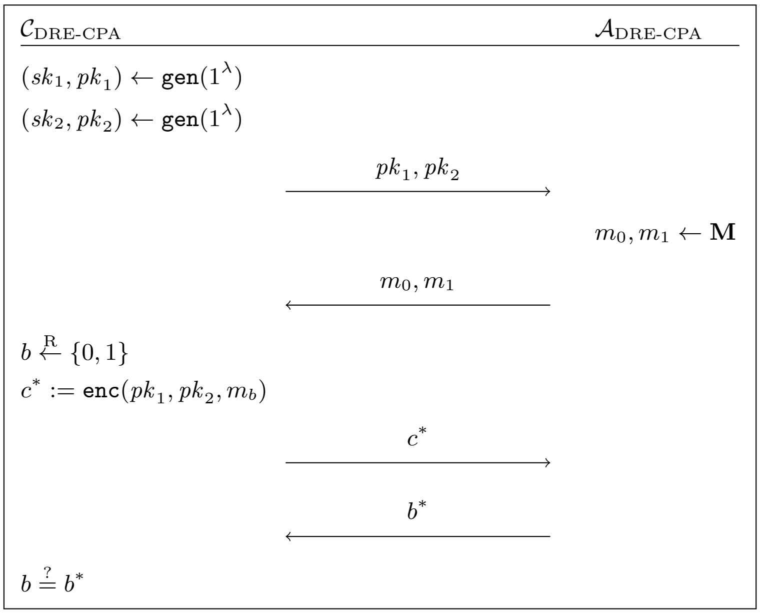
$\mathcal{C}_{\text{DRE-CPA}}$
$\mathcal{A}_{\text{DRE-CPA}}$
$(sk_1, pk_1) \leftarrow \texttt{gen}(1^\lambda)$
$(sk_2, pk_2) \leftarrow \texttt{gen}(1^\lambda)$
pk_1, pk_2
$m_0, m_1 \leftarrow \mathbf{M}$
m_0, m_1
$b \overset{R}{\leftarrow} \{0, 1\}$
$c^* := \texttt{enc}(pk_1, pk_2, m_b)$
c^*
b^*
$b \overset{?}{=} b^*$

Fig. 12. The IND-CPA DRE Game.

Definition 11 (Soundness for DRE [19]). *Consider the experiment $\mathsf{Exp}^{dre\text{-}sound}_{\mathcal{E},\mathcal{A}}$ from Figure 13 for a DRE scheme $\mathcal{E}$ and a PPT algorithm $\mathcal{A}$. The advantage of $\mathcal{A}$ is*

$$\mathsf{Adv}^{dre\text{-}sound}_{\mathcal{E},\mathcal{A}}(\lambda) := \mathbb{P}[\mathsf{Exp}^{dre\text{-}sound}_{\mathcal{E},\mathcal{A}} = 1] - \frac{1}{2}.$$

$\mathcal{E}$ satisfies soundness if for any $\mathcal{A}$, we have that $\mathsf{Adv}^{dre\text{-}sound}_{\mathcal{E},\mathcal{A}}$ is negligible in λ.

```
$\mathsf{Exp}^{\text{dre-sound}}_{\mathcal{E},\mathcal{A}}$

(1) $(sk_S, pk_S) \leftarrow \mathtt{gen}(1^\lambda); (sk_R, pk_R) \leftarrow \mathtt{gen}(1^\lambda)$
(2) $c \leftarrow \mathcal{A}(1^\lambda, sk_S, pk_S, sk_R, pk_R)$
(3) Return 1 if $\mathtt{dec}(sk_R, pk_S, pk_R, c) \neq \mathtt{dec}(sk_S, pk_S, pk_R, c)$, else return 0.
```

Fig. 13. The DRE Soundness Game.

C The Real/Ideal-Paradigm and Universal Composability

For readers who are not intimately familiar with the concept of simulation-based security we will briefly recap the ideal/real-paradigm as well as UC. More detailed explanations can be found, for instance, in [12, 13].

There is a fundamental difference between traditionally employed list- or game-based security notions and simulation-based security: Game-based notions capture individual properties, mostly indicating that a specific form of attack can not be successful. For each application scenario a list of desired properties is compiled. Protocols which fulfill all these properties—i.e. are secure against all these forms of attack—are considered secure. This, however, inherently assumes that all possible problems have been foreseen and are covered by the listed properties.

Simulation-based security, cf. [12], captures tasks in a more holistic fashion. An ideal functionality $\mathcal{F}$ is defined which acts as an incorruptible third party. Thus it can trivially solve the task in an ideal fashion: $\mathcal{F}$ takes all necessary inputs from the various parties, computes the desired function on it and returns the different outputs back to the parties. All the involved protocol parties as well as functionalities, adversaries etc. are modeled as interactive PPT turing machine (TM)s. Different instances of the same functionality are identified via unique session ID (SID)s. Note that virtually all tasks we solve with cryptographic protocols can be cast as a multi-party function evaluation, although we might not naturally think of them this way. Message transfer, for instance, is just a two-party function where the sender inputs the message and has no output while the receiver has no input but receives the message as output. Now the main principle behind simulation-based security is the ideal/real- paradigm: A protocol π securely realizes $\mathcal{F}$—write "$\pi \geq \mathcal{F}$"—if for any "real" adversary $\mathcal{A}$ and execution of the real protocol π with this adversary there is an ideal adversary $\mathcal{S}$, called "simulator", which can interact with the ideal functionality $\mathcal{F}$ in such a way that the real execution $\mathrm{EXEC}_{\pi,\mathcal{A}}$ and ideal execution $\mathrm{IDEAL}_{\mathcal{F},\mathcal{S}}$ are computationally indistinguishable. This kind of indistinguishability, however, can only provide stand-alone security—meaning it can not necessarily guarantee anything when protocols are concurrently composed with other secure protocols.

To guarantee security under general composition a slightly stronger notion is needed: Simulation security in the UC framework, introduced in [13]. The significant difference in this framework is that the distinguisher $\mathcal{Z}$, called "environment", is interactively involved in the executions. It adaptively chooses inputs, requiring the simulator to provide protocol messages and outputs *in time* and await the environments response. This implies that standard techniques such as *rewinding* do not work in UC. More formally:

Definition 12 (UC Security). *Let $\mathcal{F}$ be an ideal functionality and π a protocol. We say that π* emulates *or* securely realizes *the ideal functionality $\mathcal{F}$, if for any PPT adversary $\mathcal{A}$ there is a PPT simulator $\mathcal{S}$ such that no PPT environment $\mathcal{Z}$ can distinguish $EXEC_{\pi,\mathcal{A},\mathcal{Z}}$ from $IDEAL_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ with more than negligible probability. In this case we write*

$$\pi \geq_{UC} \mathcal{F}.$$

In [13] it was furthermore shown that the setting of a distinguishing environment and real adversary $\mathcal{A}$ can equivalently be replaced by an adversarial environment $\mathcal{Z}$ and dummy adversary $\mathcal{A}$. This is the setting we will consider in the proofs of this paper.

Note that with all the benefits of simulation-based security, it is often less cumbersome to prove game-based security notions than indistinguishability from the desired ideal functionality. Hence generic constructions—like our proof that an IND-SB-CPA secure SBE scheme suffices for SMT if used in conjunction with authenticated channels—are particularly valuable: Newly designed protocols only require game-based proofs to gain the benefits of simulation-based security.

D Ideal Functionalities

We encounter ideal functionalities in the DRE transformation in Section 3 and—even more importantly—when we show that our new notion of an IND-SB-CPA secure SBE scheme suffices in conjuction with authenticated channels to facilitate secure message transfer: We will consider both authenticated channel and SMT as ideal functionalities for this purpose. Since there have been conflicting definitions for all of the mentioned functionalities, we make the effort to explicitly recapitulate detailed formal descriptions of the versions we work with. The definitions are (sometimes loosely) based on [13] and [2] and simplified to be used with *static corruption only.*

The Ideal Functionality $\mathcal{F}_{\mathbf{AUTH}}$ The ideal functionality of authenticated channel is rather simple. It takes a receiver R and message m from some sending party S, reports this in full detail to the adversary asking for permission to deliver the message and—if permission is given—outputs message, sender and receiver information to the receiving party R. Note in particular, that $\mathcal{F}_{\text{AUTH}}$ only deals with the transmission of one single message. Multiple messages require multiple instances (each with a separate *sid*) of the functionality. More formally:

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

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

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

This definition is simplified to be used with *static corruption only* and is based on [13, Fig. 12].

The Ideal Functionality $\mathcal{F}_{\textbf{M-SMT}}$ For SMT there have been a lot of different definitions around over the years. Most of them (cp. the seven different versions in the history of [13]) deal—just like $\mathcal{F}_{\text{AUTH}}$—with transmission of a single message. A functionality for transmission of multiple messages is given in [16]. This however is based on an obsolete version of [13]. Furthermore it deals with multiple messages and senders, but still only allows for one receiver per instance of the functionality. To provide some ease of notation with the more holistic PKI approach which our new notion of IND-SB-CPA security suggests, we will give a definition of $\mathcal{F}_{\text{M-SMT}}$ which deals with multiple receivers, multiple senders *and* multiple messages rather than working with a multi-session extension (cp. [17]) of a functionality $\mathcal{F}_{\text{SMT}}$ which only transmits a single message. Note that this is just a technical difference but essentially equivalent. We also handle the commonly present backdoor messages which ask the adversary for permissions (e.g. to transmit something over a channel which the adversary can block) more explicitly. To uniquely identify these messages and their responses we utilise message IDs *mid*.

Furthermore we again give a simplified version restricted to *static corruption* and assume messages of fixed length only so there is no need for the functionality to disclose the (publicly known) lengths of messages to the adversary:

$\mathcal{F}_{\textbf{M-SMT}}$

Provides:
Multi-receiver multi-message multi-sender secure message transfer with constant message size and polynomially many parties $P \in \mathbf{P}$.

State:
Function $p_{\text{Msg}} : \mathbf{SID} \times \mathbf{MID} \to \mathbf{M} \times \mathbf{P}^2$ of pending messages.
Behaviour:

- Upon receiving $(\texttt{send}, \mathit{sid}, R, m)$ from some party S, draw fresh mid, send $(\texttt{send}, \mathit{sid}, \mathit{mid}, S, R)$ to the adversary $\mathcal{A}$ and append $(\mathit{sid}, \mathit{mid}) \mapsto (m, S, R)$ to p_{Msg}.
- Upon receiving $(\texttt{send ok}, \mathit{sid}, \mathit{mid})$ from the adversary, look up $(m, S, R) := p_{\text{Msg}}(\mathit{sid}, \mathit{mid})$. If it exists, output $(\texttt{sent}, \mathit{sid}, S, m)$ to R.

The Ideal Functionality $\mathcal{F}_{\mathbf{KRK}}$ Key registration with knowledge (KRK) does not only facilitate a simple PKI, it additionally guarantees that parties have knowledge of a secret key which corresponds to their public credentials. There are (at least) three different possibilities for an ideal functionality to provide this, which differ in who ultimately chooses the public and secret keys upon registration:

(1) The registering party asks the functionality for keys.
(2) The registering party provides the functionality with a secret key. The functionality then determines a corresponding public key.
(3) The registering party provides the functionality with a secret and public key pair. The functionality determines whether this is a well-formed key pair.

We think the third option is not only the one which gives the least power to the ideal functionality, but also the most realistic. We will therefore employ this version but keep in mind that possibilities (1) and (2) would serve us equally well. The other versions may in particular be preferable in cases where the PKI scheme do not permit a way for the functionality to efficiently decide whether a key pair is valid or not. We model this by parameterizing the ideal functionality with an efficiently computable boolean function $\mathtt{f}_{\text{Key}} : \mathbf{SK} \times \mathbf{PK} \to \{\mathsf{true}, \mathsf{false}\}$. On input of a key pair $(\mathit{sk}, \mathit{pk})$ this function outputs true for any well-formed keys and false otherwise. Note that if in version (2) there is a deterministic function $\mathtt{f}_{\mathbf{PK}}$ which on input sk outputs a corresponding public key pk, this can be transformed to the third case via $\mathtt{f}_{\text{Key}}(\mathit{sk}, \mathit{pk}) := (\mathtt{f}_{\mathbf{PK}}(\mathit{sk}) = \mathit{pk})$.

The following formal definition is loosely based on [2] but more well-defined and slightly different:

$$\mathcal{F}_{\mathbf{KRK}}^{\mathtt{f}_{\mathbf{Key}}}$$

Provides:
Key registration with knowledge.
Parameters:

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

State:

- Function $p_{\text{Reg}} : mid \mapsto (P, sk, pk)$ of pending registrations.
- Function $p_{\text{Ret}} : mid \mapsto (P_i, P_j)$ 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 $(P, sk, pk) := p_{\text{Reg}}(mid)$, check
  - ○ $\mathtt{f}_{\text{Key}}(sk, pk) = \mathsf{true}$
  - ○ $\nexists\ sk', pk' : (P, sk', pk') \in \mathbf{R}$
  - ○ $\nexists\ P', sk' : (P', sk', pk) \in \mathbf{R}$

  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 $(P_i, P_j) := p_{\text{Ret}}(mid)$ 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.

The three checks $\mathcal{F}_{\text{KRK}}$ performs before finally registering a parties credentials guarantee that only valid key pairs may be registered, that each party registers at most one key pair and that no to parties can share the same public key.

E Generic Transformations to SBE

In this section we generically construct IND-SB-CPA secure SBE schemes from various security notions like IND-RCCA and IND-CPA secure IBE and DRE. This will broaden our intuitive understanding of the new notion as well as provide a background for the concrete efficient constructions we discuss in Appendix F.

E.1 Transformation from DRE to SBE

The actual generic transformation DRE to SBE is explained in Section 3. At this point we provide additional explanations and discussions about the construction.

Let us begin with several remarks on the use of key registration via $\mathcal{F}_{\text{KRK}}$ within the construction from Section 3. Afterwards we discuss the possibility to weaken our assumption of soundness in the underlying DRE scheme and the connection between our construction an the concept of plaintext awareness.

Remark 3 (Use of $\mathcal{F}_{KRK}$).

(1) Note firstly, that the secret key *sk* is handed to the ideal functionality $\mathcal{F}_{\text{KRK}}^{\mathbf{f}_{\text{Key}}}$, but is not accessible to any other party. Realizations of $\mathcal{F}_{\text{KRK}}^{\mathbf{f}_{\text{Key}}}$ without a trusted party would probably use the public key in conjunction with a zero knowledge proof of knowledge of the secret key instead of the key itself. Secret keys are *not* registered to be handed out to other parties.

(2) For any communication partner P, it is sufficient to retrieve the key pk_P once from $\mathcal{F}_{\text{KRK}}^{\mathbf{f}_{\text{Key}}}$ and then store it internally. There is no need to communicate with $\mathcal{F}_{\text{KRK}}^{\mathbf{f}_{\text{Key}}}$ again for any communication with P.

(3) The senders public key pk_S which is retrieved during encryption can also be stored internally after generation to make its retrieval from $\mathcal{F}_{\text{KRK}}^{\mathbf{f}_{\text{Key}}}$ superfluous.

(4) We would like to point out that the realization of authenticated channels is usually constructed from a PKI. The same PKI can be augmented with zero-knowledge proofs to guarantee KRK and the possibly expensive operation for registering a public key and proving the knowledge of the according secret key has to be performed only once at the beginning.

Remark 4 (Partially Sound DRE). Using DRE in the asymmetric fashion of our construction—where one of the DRE receivers is the sender itself—we notice that the soundness of an DRE scheme is more than we require: We would be perfectly happy for soundness to hold *only* when the first receiver is able to successfully decrypt. I.e. if the first receiver outputs $\perp$ on decryption of c, we do not care whether the second receiver (who in above SBE construction is the sender) outputs $\perp$ as well or decrypts the ciphertext c to some valid message m.

We call this weaker notion of DRE soundness *partial soundness*. Note that (particularly) in this case, decryption algorithms between the first and second receiver might differ, i.e. a partially sound DRE scheme might be given by PPT algorithms $(\texttt{gen}, \texttt{enc}, \texttt{dec}, \texttt{dec}')$. It is easy to see that with the above construction of an SBE from a DRE scheme, we already achieve IND-SB-CPA security from an IND-CCVA2 secure and partially sound DRE.

Such an IND-CCVA2 secure partially sound DRE scheme can be constructed by taking an IND-CCA2 secure PKE scheme $(\texttt{gen}, \texttt{enc}, \texttt{dec})$, where $\texttt{enc}$ is deterministic with the last argument representing the randomness used. From this we define a DRE scheme $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec}, \texttt{Dec}')$ via:

$$\begin{aligned}
\texttt{Gen} &:= \texttt{gen} \\
\texttt{Enc} &: (pk, pk', m) \mapsto (\texttt{enc}(pk, (m, s); r), \texttt{enc}(pk', m; s)) \\
\texttt{Dec} &: (sk, pk, pk', (c, c')) \mapsto \begin{cases} \perp, & c' \neq \texttt{enc}(pk', m; s) \\ m, & \text{otherwise} \end{cases} \\
&\qquad \text{where } (m, s) := \texttt{dec}(sk, c) \\
\texttt{Dec}' &: (sk', pk, pk', (c, c')) \mapsto \texttt{dec}(sk', c').
\end{aligned}$$

Right now, we are not aware of any IND-CCVA2 secure partially sound DRE construction which does not employ an IND-CCA2 secure PKE scheme. Since there are easier ways to construct an IND-SB-CPA secure SBE scheme from CCA2 (e.g. Appendix E.2) we will not go into this weaker DRE notion any further.

Remark 5 (Connection to PA). The way we use DRE here is somewhat related to the concept of PA (cf. [4, 3]). The (non-equivalent) notions from these two papers do not intrinsically satisfy IND-SB-CPA however. They focus on the property that knowledge about all oracle queries made by the adversary, or respectively the adversary's input and randomness, is enough to permit the existence of an efficient plaintext extractor. Which is not sufficient to prevent replay attacks or satisfy IND-SB-CPA security. The key difference is that an adversary effectively has knowledge about the plaintext in any valid ciphertext it can *construct* from only the public key of the receiver.exhaustive For IND-SB-CPA on the other hand, the adversary would have to be plaintext-aware about any ciphertext that a party under their control could validly send to the receiver—regardless of who constructed it or where it came from. Note that from two of the PA definitions in [4, 3], IND-SB-CPA can be constructed anyway (cf Appendix E.2) as they imply CCA2 which in turn implies RCCA.

The notion of RPA introduced by [29] on the other hand is a lot closer to the construction we presented. It states that if an adversary can successfully construct a ciphertext c from some party P to R, then the adversary's knowledge about the key registration of P is enough (together with public keys) to decrypt c. This does not only prohibit replay attacks and implies IND-SB-CPA in case the oracle has the same knowledge about key registration that the adversary has (an in above DRE construction). It also implies another property which is irrelevant for IND-SB-CPA: If the adversary has no knowledge about the key registration of a party (e.g. for the party S in the SB-CPA game), then they can either not construct a valid ciphertext from this party to R, or the ciphertext they constructed can already be decrypted with public keys only. IND-SB-CPA security does not require this.

E.2 SB-CPA via RCCA

As explained in Section 1.1, RCCA variants are one of the previously weakest security notions used in conjunction with authenticated channels to realize secure message transfer. For this appendix we take "IND-RCCA" to be the weakest among the various RCCA definitions from [1] and show that this is sufficient to easily construct an IND-SB-CPA secure SBE scheme. This means in particular that all stronger definitions of RCCA as well as CCA2 (which implies RCCA) can be used to achieve SB-CPA security via this construction.

So let $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ be an IND-RCCA secure PKE scheme. We define a new SBE scheme $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ via:

$$\begin{aligned}
&\mathtt{Gen} := \mathtt{gen}\\
&\mathtt{Enc} : (pk_R, S, m) \mapsto \mathtt{enc}(pk_R, m\|S)\\
&\mathtt{Dec} : (sk_R, S, c) \mapsto \begin{cases} \bot, \nexists m : \mathtt{dec}(sk_R, c) = m\|S \\ m, \mathtt{dec}(sk_R, c) = m\|S \end{cases}
\end{aligned}$$

By encrypting the sender's ID together with the message, the receiver can check on decryption whether this corresponds to the sender they expected or not. Of course any malicious sender can insert any party ID they want into the ciphertext, but intuitively this construction guarantees that encrypted messages can not just be copied and used by another party as if they had encrypted it themselves. This is exactly what we need for IND-SB-CPA security.

Lemma 5. *The SBE scheme* $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ *is IND-SB-CPA secure.*

Proof. Assume that $\mathcal{A}_{\text{SB-CPA}}$ is an adversary which has non-negligible success probability in winning the SB-CPA game with respect to $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$. With this we construct an adversary $\mathcal{A}_{\text{RCCA}}$ for the IND-RCCA game with respect to $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ as shown in Figure 14.

We still need to show, that the responses from $\mathcal{A}_{\text{RCCA}}$ to $\mathcal{A}_{\text{SB-CPA}}$ are indistinguishable from the responses $\mathcal{O}_{\text{SB-CPA}}$ would give. For the first oracle phase, this is easy to see as $\mathcal{A}_{\text{RCCA}}$ does exactly the same as $\mathcal{O}_{\text{SB-CPA}}$ would. The only difference is that messages to R get decrypted with the help of $\mathcal{O}_{\text{RCCA}}$. Nevertheless this decryption exactly amounts to $m := \mathtt{Dec}(sk_{R'}, S', c)$.

Although Oracle Phase II is identical to Oracle Phase I in the behaviour of $\mathcal{A}_{\text{RCCA}}$, there is actually a difference in the responses of $\mathcal{O}_{\text{RCCA}}$ in case $pk_{R'} = pk_R$: if c is an encryption of one of the challenge messages $(m_0\|S)$, $(m_1\|S)$, the oracle $\mathcal{O}_{\text{RCCA}}(c)$ will reply with *test* rather than a decrypted message. Note, however, that since $S' \neq S$ in this case, a ciphertext c containing $(m_b\|S)$ would yield $\mathtt{Dec}(sk_{R'}, S', c) = \bot$. This is exactly the answer $\mathcal{A}_{\text{RCCA}}$ gives to $\mathcal{A}_{\text{SB-CPA}}$. Therefore, there is no difference in the output of the first or second oracle phase to $\mathcal{O}_{\text{SB-CPA}}$ in the view of $\mathcal{A}_{\text{SB-CPA}}$.

Please note that $\mathcal{A}_{\text{RCCA}}$ has exactly the same success probability as $\mathcal{A}_{\text{SB-CPA}}$, which is non-negligible by assumption. □

E.3 SB-CPA via IBE

Let $(\mathtt{gen}, \mathtt{ext}, \mathtt{enc}, \mathtt{dec})$ be an IND-sID-CPA secure IBE scheme. Furthermore let $\mathtt{f}_{\text{ID}}$ be the (publicly known) function of the parties' IBE IDs. We assume this function to be efficiently computable and injective (i.e. no two parties share the same ID). We define a new SBE scheme $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ by:

$$\begin{aligned}
&\mathtt{Gen} := \mathtt{gen} \text{ (i.e. } (SK, PK) := (msk, mpk))\\
&\mathtt{Enc} : (PK_R, S, m) = (mpk_R, S, m) \mapsto \mathtt{enc}(mpk_R, \mathtt{f}_{\text{ID}}(S), m)\\
&\mathtt{Dec} : (SK_R, S, c) = (msk_R, S, c) \mapsto \mathtt{dec}(\mathtt{ext}(msk_R, \mathtt{f}_{\text{ID}}(S)), c).
\end{aligned}$$

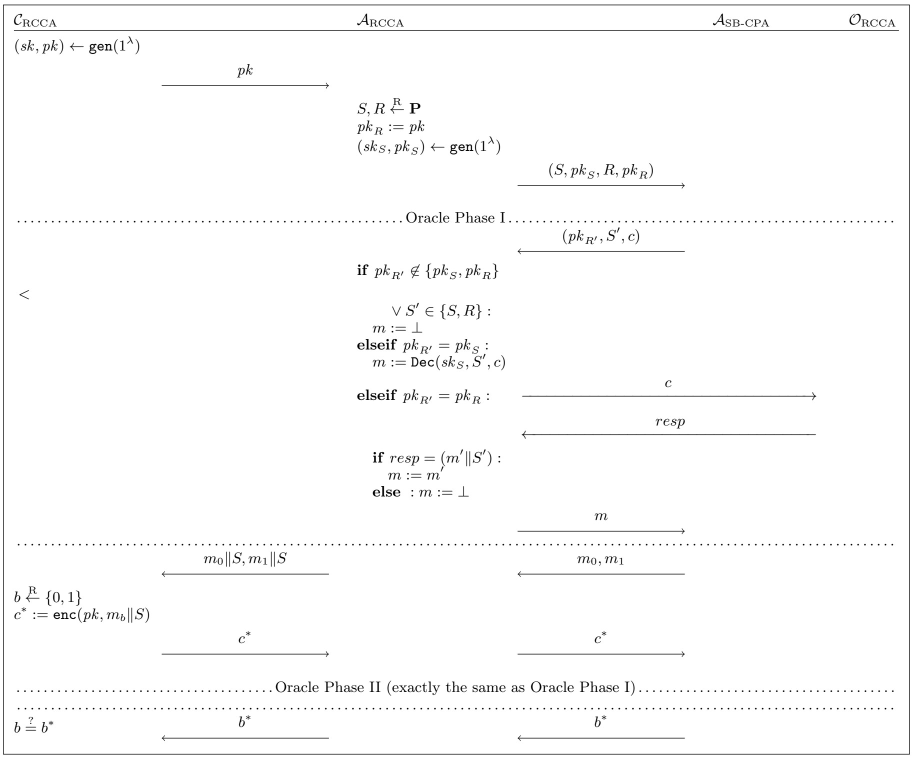
$\mathcal{C}_{\text{RCCA}}$
$\mathcal{A}_{\text{RCCA}}$
$\mathcal{A}_{\text{SB-CPA}}$
$\mathcal{O}_{\text{RCCA}}$
$(sk, pk) \leftarrow \texttt{gen}(1^\lambda)$
pk
$S, R \overset{R}{\leftarrow} \mathbf{P}$
$pk_R := pk$
$(sk_S, pk_S) \leftarrow \texttt{gen}(1^\lambda)$
(S, pk_S, R, pk_R)
Oracle Phase I
$(pk_{R'}, S', c)$
$<$
if $pk_{R'} \notin \{pk_S, pk_R\}$
$\vee S' \in \{S, R\}$:
$m := \perp$
elseif $pk_{R'} = pk_S$:
$m := \texttt{Dec}(sk_S, S', c)$
elseif $pk_{R'} = pk_R$:
c
$resp$
if $resp = (m' \| S')$:
$m := m'$
else : $m := \perp$
m
$m_0 \| S, m_1 \| S$
m_0, m_1
$b \overset{R}{\leftarrow} \{0, 1\}$
$c^* := \texttt{enc}(pk, m_b \| S)$
c^*
c^*
Oracle Phase II (exactly the same as Oracle Phase I)
$b \overset{?}{=} b^*$
b^*
b^*

Fig. 14. Reduction for RCCA Construction

In this construction we use the underlying IBE scheme "the wrong way round", generating one instance of the scheme for each receiver. This intuitively corresponds to the simpler and less efficient idea of having the receiver use a different key pair for each sender (which would also satisfy SB-CPA security). An interesting question would be whether we could construct a kind of double IBE scheme, where each ID can be used to extract a master secret key for a complete IBE scheme (i.e. $(msk_R, mpk_R) = (\texttt{ext}_{\text{master}}(\texttt{f}_{\text{ID}}(R)), \texttt{f}_{\text{ID}}(R)))$ as well as to extract individual user secret keys (i.e. $\texttt{ext}(msk_R, \texttt{f}_{\text{ID}}(S)))$. Unfortunately this goes beyond the scope of this paper right now, so let us first satisfy ourselves that we actually achieve SB-CPA security this way:

Lemma 6. *The SBE scheme* $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ *is IND-SB-CPA secure.*

Proof. Assuming an adversary $\mathcal{A}_{\text{SB-CPA}}$ who has non-negligible probability in winning the IND-SB-CPA game with respect to $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$, we construct an adversary $\mathcal{A}_{\text{sID-CPA}}$ with non-negligible success probability in winning the IND-sID-CPA game with respect to $(\texttt{gen}, \texttt{f}_{\text{ID}}, \texttt{ext}, \texttt{enc}, \texttt{dec})$ as shown in Figure 15.

In this example the behaviour in both oracle phases is again identical and corresponds to the same steps $\mathcal{O}_{\text{SB-CPA}}$ would take on input $(mpk_{R'}, S', c)$. To

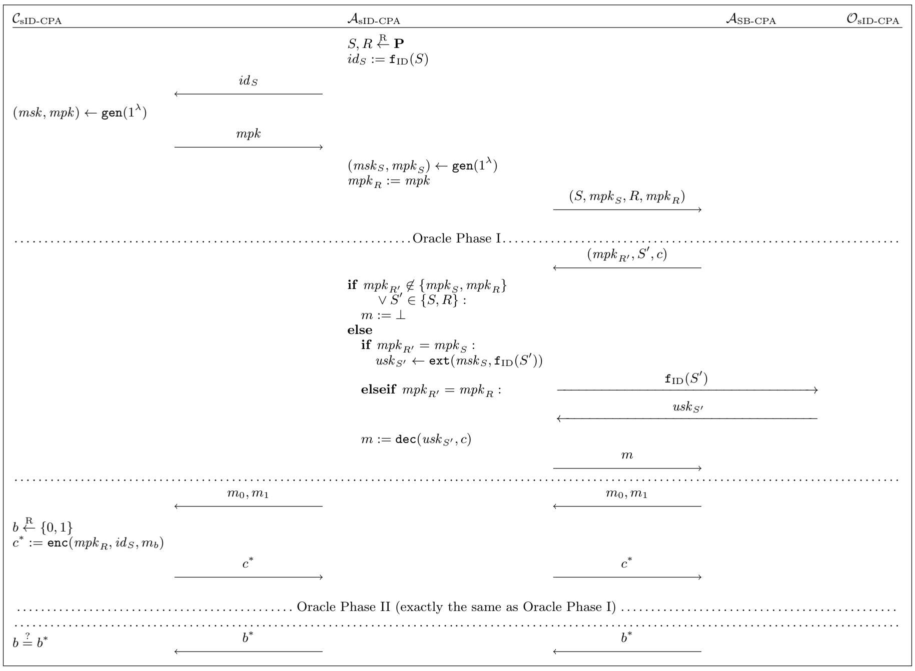
$\mathcal{C}_{\text{sID-CPA}}$
$\mathcal{A}_{\text{sID-CPA}}$
$\mathcal{A}_{\text{SB-CPA}}$
$\mathcal{O}_{\text{sID-CPA}}$
$S, R \overset{R}{\leftarrow} \mathbf{P}$
$id_S := \mathtt{f}_{\text{ID}}(S)$
id_S
$(msk, mpk) \leftarrow \mathtt{gen}(1^\lambda)$
mpk
$(msk_S, mpk_S) \leftarrow \mathtt{gen}(1^\lambda)$
$mpk_R := mpk$
(S, mpk_S, R, mpk_R)
Oracle Phase I
$(mpk_{R'}, S', c)$
if $mpk_{R'} \notin \{mpk_S, mpk_R\}$
$\vee S' \in \{S, R\}$:
$m := \bot$
else
if $mpk_{R'} = mpk_S$:
$usk_{S'} \leftarrow \mathtt{ext}(msk_S, \mathtt{f}_{\text{ID}}(S'))$
elseif $mpk_{R'} = mpk_R$:
$\mathtt{f}_{\text{ID}}(S')$
$usk_{S'}$
$m := \mathtt{dec}(usk_{S'}, c)$
m
m_0, m_1
m_0, m_1
$b \overset{R}{\leftarrow} \{0, 1\}$
$c^* := \mathtt{enc}(mpk_R, id_S, m_b)$
c^*
c^*
Oracle Phase II (exactly the same as Oracle Phase I)
$b \overset{?}{=} b^*$
b^*
b^*

Fig. 15. Reduction for IBE Construction

decrypt messages sent to R, $\mathcal{A}_{\text{sID-CPA}}$ enlists the help of $\mathcal{O}_{\text{sID-CPA}}$ instead of extracting $usk_{S'}$ itself. $\mathcal{O}_{\text{sID-CPA}}$ will always provide $\mathcal{A}_{\text{sID-CPA}}$ with the user secret key $usk_{S'}$, as it was checked earlier that $S' \notin \{S, R\}$ and therefore $\mathtt{f}_{\text{ID}}(S') \neq id_S$.

Note that $\mathcal{A}_{\text{sID-CPA}}$ has exactly the same success probability as $\mathcal{A}_{\text{SB-CPA}}$—which is non-negligible by assumption—because messages, challenge and the bit b^* are all just forwarded. □

Remark 6 (Weaker IBE). Looking closely at above reduction we notice that a weaker notion than IND-sID-CPA security would suffice: The ability for $\mathcal{A}_{\text{sID-CPA}}$ to choose the sender's IBE ID id_S is completely unnecessary.[10] If the sender ID was chosen by the challenger $\mathcal{C}_{\text{IBE}}$ together with the parameters (msk, mpk) at the start of the game, the reduction would work equally well. This is not only because the credentials (msk_S, mpk_S) are independent of the sender's ID and could still be chosen by $\mathcal{A}_{\text{IBE}}$. The main reason is that we only need $\mathcal{A}_{\text{IBE}}$ to have as much power over choosing the sender and its credentials as $\mathcal{A}_{\text{SB-CPA}}$ has in the IND-SB-CPA game. Digressing to some of the notation we introduce in Appendix G.3: From any IBE scheme which is IND-set-CPA for choosing mode $\mathsf{set} = (\mathcal{P}$, time, *Receiver*$)$, we get an IND-$\mathsf{set}'$-SB-CPA secure SBE scheme

[10] Note that "sender" in this case denotes the *receiving* party in the classic IBE scheme. For this argument we will continue to call this party "sender" and "S" to hopefully increase readability and avoid unnecessary ambiguity.

with choosing mode $\mathsf{set}' = (\mathcal{P}, \text{time}, \mathit{Sender})$ from the above construction. In particular, an IND-($\mathcal{C}$, Start, $((\mathit{msk}, \mathit{mpk})$, Receiver R, $(\mathit{msk}_R, \mathit{mpk}_R)))$-CPA secure IBE scheme with the respective extraction oracles suffices to construct IND-SB-CPA secure SBE.

F McEliece and LWE Constructions of SB-CPA in the Standard Model

The following definitions are used for Theorems 1, 4 and 6.

Definition 13 (Indistinguishability Assumption for Goppa Codes). *Let D be a probabilistic algorithm. For every $n \in \mathbb{N}$, we define*

$$\begin{aligned}\mathsf{Adv}^{ind}_{D,\mathsf{G}}(n) = \mathbb{P}\Big[((\mathsf{G}, t), sk) \leftarrow \mathsf{gen}(1^n) | D(\mathsf{G}, t) = 1\Big] \\ -\mathbb{P}\Big[\mathsf{U} \overset{\mathrm{R}}{\leftarrow} \mathcal{U}_{l\times n} | D(\mathsf{U}, t) = 1\Big]\end{aligned}$$

Also we define the advantage function of the problem as follows. For any ω,

$$\mathsf{Adv}^{ind}_{\mathsf{G}}(n, \omega) = \underset{D}{max}\Big\{\mathsf{Adv}^{ind}_{D,\mathsf{G}}(n)\Big\} \tag{1}$$

where the maximum is over all D with time-complexity ω. We say G is indistinguishable if, for every poly bounded ω and every sufficiently large n, $\mathsf{Adv}^{ind}_{\mathsf{G}}(n, \omega)$ is negligible.

Definition 14 (LPN Search Problem (LPNSP)). *Let s be a random binary string of length l. We consider the Bernoulli distribution $\mathcal{B}_\theta$ with parameter $\theta \in (0, \frac{1}{2})$. Let $\mathcal{Q}_{s,\theta}$ be the following distribution:*

$$\{(\mathsf{a}, \langle \mathsf{s}, \mathsf{a}\rangle \oplus \mathsf{e}) | \mathsf{a} \overset{\mathrm{R}}{\leftarrow} \{0, 1\}^l, \mathsf{e} \leftarrow \mathcal{B}_\theta\}$$

For an adversary $\mathcal{A}$ trying to discover the random string s, we define its advantage as

$$\mathsf{Adv}^{LPN_\theta}_{\mathcal{A}}(l) = \mathbb{P}\left[\mathcal{A}^{\mathcal{Q}_{s,\theta}} = \mathsf{s} | \mathsf{s} \overset{\mathrm{R}}{\leftarrow} \{0, 1\}^l\right]$$

The LPN_θ is hard if the advantage of all PPT adversaries $\mathcal{A}$ that make a **polynomial number of oracle ($\mathcal{Q}_{s,\theta}$) queries** is negligible.

Definition 15 (LPN Distinguishing Problem (LPNDP)). *Let s and $\mathcal{Q}_{s,\theta}$ be as in 14. Let $\mathcal{A}$ be a PPT adversary, whose distinguishing advantage between $\mathcal{Q}_{s,\theta}$ and the uniform distribution $\mathcal{U}_{l+1}$ after issuing at most q queries is defined as follows*

$$\mathsf{Adv}^{LPNDP_\theta}_{\mathcal{A}}(q, l) = \Big|\mathbb{P}[\mathcal{A}^{\mathcal{Q}_{\mathbf{s},\theta}} = 1 | \mathsf{s} \overset{\mathrm{R}}{\leftarrow} \{0, 1\}^l] - \mathbb{P}[\mathcal{A}^{\mathcal{U}_{l+1}} = 1]\Big|$$

F.1 IND-CPA DRE via McEliece with Soundness

The conceptually similar construction of Kiltz et al. [31] is a low-noise LPN construction that is a stag-wCCA secure TBE.

Parameters:

- Let $\mathcal{G}_{n,t}$ be the family of irreducible binary Goppa-codes of length n, which can correct up to t errors with a code dimension l.
- Let $\theta = \frac{t}{n} + \epsilon$ be the Bernoulli parameter of the error for some $\epsilon > 0$.
- Let $G_2 \in \{0,1\}^{l \times n}$ be the publicly known generator matrix of a code from $\mathcal{G}_{n,t}$, where *Correct* is the according error-correcting algorithm.
- Let $M = \{0,1\}^l$ be the message space.

Cryptosystem: We define $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ as follows.

- The key generation algorithm $\mathtt{gen}(1^n)$ works as follows:
  - ○ Sample a random Matrix $C \in \{0,1\}^{l \times n}$
  - ○ Sample a generator matrix $G' \in \{0,1\}^{l \times n}$ for a code from $\mathcal{G}_{n,t}$.
  - ○ Sample a random non-singular matrix $S \in \{0,1\}^{l \times l}$.
  - ○ Sample a random permutation matrix $P \in \{0,1\}^{n \times n}$.
  - ○ Set $G := SG'P$.
  - ↪ Return $pk = (G, C, t)$ and $sk = (S, G', P)$
- The encryption algorithm $\mathtt{enc}(pk_R, pk_S, m)$ works as follows:
  - ○ Parse pk_R as (G_R, C_R, t) and pk_S as (G_S, C_S, t)
  - ○ Sample $s \overset{R}{\leftarrow} \{0,1\}^l$
  - ○ $e_R, e_S, e \leftarrow \mathcal{B}_\theta$
  - ○ $c_R = s \cdot G_R \oplus e_R$
  - ○ $c_S = s \cdot G_S \oplus e_S$
  - ○ $c' = s \cdot C_S \oplus e \oplus m \cdot G_2$
  - ↪ Return $c = (c_R, c_S, c')$.
- The decryption algorithm $\mathtt{dec}(sk_R, pk_S, c)$ works as follows:
  - ○ Parse c as (c_R, c_S, c') and sk_R as (S_R, G'_R, P_R)
  - ○ Compute $\hat{y}_R = c_R \cdot P_R^{-1} = (s \cdot S_R) \cdot G'_R \oplus e_R \cdot P_R^{-1}$
  - ○ Compute $s \cdot S_R = Correct(\hat{y}_R)$
  - ○ Compute $s = (s \cdot S_R)S_R^{-1}$
  - ○ Compute $c'_S = s \cdot G_S$
  - ○ Set the verification bit b as follows
    - ∗ Set $b = 1$ if the hamming weight of $c'_S \oplus c_S$ is smaller than t.
    - ∗ Set $b = 0$ otherwise.
  - ↪ If $b = 0$ return $\perp$, otherwise:
    - ∗ Compute $c' = s \cdot C_S$
    - ∗ Correct the error from $m = Correct(c \oplus c')$, where $c \oplus c' = (m \cdot G_2 \oplus e)$.
    - ↪ Return m.

Theorem 4. *The DRE scheme* $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ *is IND-CPA secure, given that both the McEliece assumption and the LPNDP hold. In particular, let* $\mathcal{A}$ *be an IND-CPA adversary against the cryptosystem. Then there is a distinguisher* $\mathcal{B}$ *for Goppa codes and a distinguisher* $\mathcal{D}$ *for the LPNDP, such that for all* $\lambda \in \mathbb{N}$

$$\mathsf{Adv}_{\mathcal{A}}^{CPA}(\lambda) \leq \mathsf{Adv}_{\mathcal{D}}^{LPNDP_\theta(3n,l)}(\lambda) + 2 \times \mathsf{Adv}_{\mathcal{B}_R,G_R}^{ind}(\lambda).$$

Proof. **Game 1** This is the DRE IND-CPA game. The challenge ciphertext will be of the form:

$$c^* = (\mathsf{s} \cdot \mathsf{G}_S \oplus \mathsf{e}_S, \mathsf{s} \cdot \mathsf{G}_R \oplus \mathsf{e}_R, \mathsf{s} \cdot \mathsf{C}_S \oplus \mathsf{e} \oplus m_b \cdot \mathsf{G}_2)$$

Let $\mathsf{G}^* := (\mathsf{G}_S|\mathsf{G}_R|\mathsf{C}_S) \in \{0,1\}^{l\times(3\cdot n)}$ and $\mathsf{e}^* := (\mathsf{e}_S|\mathsf{e}_R|\mathsf{e}) \in \{0,1\}^{3\cdot n}$. In order to simplify the understanding of the transitions to the next games we rewrite c^* into the following form, where $\mathit{0}$ has dimension $2 \cdot n$.

$$c^* = (\mathsf{s} \cdot \mathsf{G}^* \oplus \mathsf{e}^* \oplus (\mathbf{0}, m_b \cdot \mathsf{G}_2))$$

Game 2 Same as Game 1, except that the generator matrix G_R within the public key is replaced by uniformly random matrix $\mathsf{U}_R \in \{0,1\}^{l\times n}$. Therefore, the receiver public key in Game 2 is $pk_R := (\mathsf{U}_R, \mathsf{C}_R, t)$.

Any distinguisher $\mathcal{A}_R$ distinguishing between Game 1 and Game 2 yields a distinguisher $\mathcal{B}_R$ for a random irreducible Goppa code from a random linear code. Therefore,

$$\mathsf{Adv}_{\mathcal{A}}^{CPA} \leq \mathsf{Adv}_{\mathcal{A},\text{Game } 2}^{CPA} + \mathsf{Adv}_{\mathcal{B}_R,\mathsf{G}_R}^{ind}(\lambda)$$

Game 3 Same as Game 2, except that the generator matrix G_S within the public key is replaced by uniformly random matrix $\mathsf{U}_S \in \{0,1\}^{l\times n}$. Therefore, the sender public key in Game 3 is $pk_S := (\mathsf{U}_S, \mathsf{C}_S, t)$.

Any distinguisher $\mathcal{A}_S$ distinguishing between Game 2 and Game 3 yields a distinguisher $\mathcal{B}_S$ for a random irreducible Goppa code from a random linear code. Therefore, w.l.o.g

$$\mathsf{Adv}_{\mathcal{A}}^{CPA} \leq \mathsf{Adv}_{\mathcal{A},\text{Game } 3}^{CPA} + 2 \times \mathsf{Adv}_{\mathcal{B}_R,\mathsf{G}_R}^{ind}(\lambda) \tag{2}$$

Game 4 Instead of computing the challenge ciphertext as

$$c^* = (\mathsf{s} \cdot \mathsf{G}^* \oplus \mathsf{e}^* \oplus (\mathbf{0}, m_b \cdot \mathsf{G}_2))$$

the challenger chooses $c^* \overset{\text{R}}{\leftarrow} \mathcal{U}_{3\cdot n}$ instead. We justify this replacement by observing that $(\mathsf{s} \cdot \mathsf{G}^* \oplus \mathsf{e}^*)$ is an instance of the LPNDP and therefore can be replaced by a random value $\mathsf{u} \overset{\text{R}}{\leftarrow} \mathcal{U}_{3\cdot n}$. The random vector u acts as a One-Time Pad s.t. the ciphertext is transformed into a uniformly distributed random value:

$$c^* = (\mathsf{u} \oplus (\mathit{0}, m_b \cdot \mathsf{G}_2))$$

This is the challenge ciphertext used in Game 4. The advantage of the original DRE IND-CPA adversary $\mathcal{A}$ is now 0, as the succeeding probability is $\frac{1}{2}$. The indistinguishability follows from the hardness of the LPNDP.

If the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has non-negligibly different succeeding probabilities in Games 3, 4 then we can use this adversary to solve any LPNDP. To this end we can use the following distinguisher D for a given LPNDP oracle $\mathcal{O}$, which is either $\mathcal{Q}_{\mathsf{s},\theta}$ with $\mathsf{s} \in \{0,1\}^l$ or $\mathcal{U}_{l+1}$ by issuing $(3 \cdot n)$ number of queries.

(1) Generate the public keys $\mathsf{U}_R, \mathsf{U}_S, \mathsf{C}_S$ for $\mathcal{A}$.
- Query the LPNDP oracle: $(\mathsf{a}_1, b_1), \cdots, (\mathsf{a}_{3\cdot n}, b_{3\cdot n}) \leftarrow \mathcal{O}$
- Set $\mathsf{U}_R = (\mathsf{a}_1^t, \cdots, \mathsf{a}_n^t)$, $\mathsf{U}_S = (\mathsf{a}_{n+1}^t, \cdots, \mathsf{a}_{2n}^t)$ and $\mathsf{C}_S = (\mathsf{a}_{2n+1}^t, \cdots, \mathsf{a}_{3n}^t)$.

(2) $(m_0, m_1) \leftarrow \mathcal{A}_1(\mathsf{U}_S, \mathsf{C}_S, \mathsf{U}_R)$

(3) $b \overset{\text{R}}{\leftarrow} \{0,1\}$

(4) Set the challenge ciphertext to

$$c^* = ((b_1, \cdots, b_{3\cdot n}) \oplus (\mathbf{0}, m_b \cdot \mathsf{G}_2))$$

(5) $b' \leftarrow \mathcal{A}_2(\mathsf{U}_S, \mathsf{C}_S, \mathsf{U}_R, c^*)$

(6) If $b' = b$ then return 1, else return 0.

If $\mathcal{O} = \mathcal{Q}_{\mathsf{s},\theta}$, then we have the same situation as in Game 3, else $\mathcal{O} = \mathcal{U}_{l+1}$ and we have the same situation as in Game 4. Therefore:

$$\mathsf{Adv}_{\mathcal{A},Game3}^{CPA} \leq \mathsf{Adv}_{\mathcal{A},\text{Game 4}}^{CPA} + \mathsf{Adv}_{\mathcal{D}}^{LPNDP}(\lambda) = \mathsf{Adv}_{\mathcal{D}}^{LPNDP_\theta(3n,l)}(\lambda)$$

This concludes that the overall advantage is

$$\mathsf{Adv}_{\mathcal{A}}^{CPA} \leq \mathsf{Adv}_{\mathcal{D}}^{LPNDP_\theta(3n,l)}(\lambda) + 2 \times \mathsf{Adv}_{\mathcal{B}_R,\mathsf{G}_R}^{ind}(\lambda)$$

□

Theorem 5. *The encryption scheme* $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ *satisfies DRE-soundness.*

The definition of the soundness property of DRE can be found in Appendix B.4.

Proof. If the sender and the receiver are able to extract the same randomness s, then they will extract the same message m from the ciphertext due to the determinism of the decryption.

Now, consider the case $\mathtt{dec}(pk_S, sk_R, c) = \perp$ and $\mathtt{dec}(pk_R, sk_S, c) = m$. We will prove by contradiction that this case never happens. Parse c as $(\mathsf{c}_R, \mathsf{c}_S, \mathsf{c}')$ and $pk_R = (\mathsf{G}_R, \mathsf{C}_R)$ and $pk_S = (\mathsf{G}_S, \mathsf{C}_S)$, where the first two parts of the ciphertext have the following form due to being textbook McEliece ciphertexts.

$$\mathsf{c}_R = \mathsf{s}' \cdot \mathsf{G}_R \oplus \mathsf{e}_R$$
$$\mathsf{c}_S = \mathsf{s} \cdot \mathsf{G}_S \oplus \mathsf{e}_S$$

From $\mathtt{dec}(pk_S, sk_R, c) = \perp$ it follows that the verification step has failed. This means that after recovering the randomness s' from c_R by $recover(sk_R, \mathsf{c}_R) = \mathsf{s}'$, where *recover* is the textbook McEliece decryption, the hamming distance of $\mathsf{s}' \cdot \mathsf{G}_S$ has to be greater or equal than t to c_S. Considering that $\mathsf{s}' \cdot \mathsf{G}_S \oplus \mathsf{c}_S = \mathsf{s}'\mathsf{G}_S \oplus \mathsf{s}\mathsf{G}_S \oplus \mathsf{e}_S$ we get

$$wgt(\mathsf{s}'\mathsf{G}_S \oplus \mathsf{s}\mathsf{G}_S \oplus \mathsf{e}_S) \geq t$$

From this it follows that $\mathsf{s}' \neq \mathsf{s}$ due to e_S being guaranteed to have the hamming weight $wgt(\mathsf{e}_S) < t$ by the syndrome decoding algorithm within the textbook McEliece decryption.
However, from $\mathtt{dec}(pk_R, sk_S, c) = m$ it follows that

$$wgt(\mathsf{s}\mathsf{G}_R \oplus \mathsf{s}'\mathsf{G}_R \oplus \mathsf{e}_R) < t$$

Now, $\mathsf{s}\mathsf{G}_R$ and $\mathsf{s}'\mathsf{G}_R$ are codewords for $\mathsf{s} \neq \mathsf{s}'$ and therefore are guaranteed to have hamming distance $d(\mathsf{s}\mathsf{G}_R, \mathsf{s}'\mathsf{G}_R) \geq 2t + 1$. This contradicts with $wgt(\mathsf{s}\mathsf{G}_R \oplus \mathsf{s}'\mathsf{G}_R \oplus \mathsf{e}_R) < t$ as $wgt(\mathsf{e}_R) < t$. Therefore, this case is not possible.
Similar considerations will yield that the case $\mathtt{dec}(pk_S, sk_R, c) = m_R$ and $\mathtt{dec}(pk_R, sk_S, c) = m_S$ with $m_R \neq m_S$ is impossible.
Conclusively, $\mathbb{P}[\mathsf{Exp}^{\text{sound}}_{\mathcal{A},\Pi} = 1] = 0$. □

F.2 IND-CPA DRE via 2-repetition McEliece

Let $PKE_{\text{McE},2} = (\mathtt{gen}_{\text{McE},2}, \mathtt{enc}_{\text{McE},2}, \mathtt{dec}_{\text{McE},2})$ be a verifiable 2-repetition encryption scheme, which is a variant ($k = 2$) from [25] based on the McEliece cryptosystem, and $M = \{0,1\}^l$, where $l = l_1 + l_2$ (as in [39]). We define the cryptosystem as follows.

- The key generation algorithm $\mathtt{gen}_{\text{McE},2}(1^n)$ works as follows:
  - ○ Sample a generator matrix $\mathsf{G}' \in \{0,1\}^{l \times n}$ of an irreducible binary Goppa code, which can correct up to t errors with a code dimension l.
  - ○ Sample a random non-singular matrix $\mathsf{S} \in \{0,1\}^{l \times l}$.
  - ○ Sample a random permutation matrix $\mathsf{P} \in \{0,1\}^{n \times n}$.
  - ○ Set $\mathsf{G} := \mathsf{S}\mathsf{G}'\mathsf{P}$.
  - ↪ Return $pk = (\mathsf{G}, t)$ and $sk = (\mathsf{S}, \mathsf{G}', \mathsf{P})$
- The encryption algorithm $\mathtt{enc}_{\text{McE},2}(pk_R, pk_S, m)$ works as follows:
  - ○ Parse pk_R as (G_R, t) and pk_S as (G_S, t)
  - ○ Sample $\mathsf{s} \xleftarrow{R} \{0,1\}^{l_1}$, where $l_1 \in \Omega(n)$.
  - ○ $\mathsf{e}_R, \mathsf{e}_S \leftarrow \mathcal{B}_\theta$, where $\mathcal{B}_\theta$ is the Bernoulli distribution with $\theta = \frac{t}{n} - \varepsilon$ for some $\varepsilon > 0$.
  - ○ $\mathsf{c}_R = [\mathsf{s}|m] \cdot \mathsf{G}_R \oplus \mathsf{e}_R$
  - ○ $\mathsf{c}_S = [\mathsf{s}|m] \cdot \mathsf{G}_S \oplus \mathsf{e}_S$
  - ↪ Return $c = (\mathsf{c}_R, \mathsf{c}_S)$.
- The decryption algorithm $\mathtt{dec}_{\text{McE},2}(sk_R, pk_S, c)$ works as follows:
  - ○ Parse c as $(\mathsf{c}_R, \mathsf{c}_S)$ and sk_R as $(\mathsf{S}_R, \mathsf{G}'_R, \mathsf{P}_R)$
  - ○ Compute $\hat{\mathsf{y}}_R = \mathsf{c}_R \cdot \mathsf{P}_R^{-1} = ([\mathsf{s}|m]\mathsf{S}_R) \cdot \mathsf{G}'_R \oplus \mathsf{e}_R \cdot \mathsf{P}_R^{-1}$
  - ○ Compute $[\mathsf{s}|m] \cdot \mathsf{S}_R = Correct(\hat{\mathsf{y}}_R)$
  - ○ Compute $[\mathsf{s}|m] = ([\mathsf{s}|m]\mathsf{S}_R)\mathsf{S}_R^{-1}$

- ∘ Compute $c'_S = [s|m] \cdot G_S$
- ∘ Set the verification bit b as follows
  - ∗ Set $b = 1$ if the hamming weight of $c'_S \oplus c_S$ is smaller than t.
  - ∗ Set $b = 0$ otherwise.

$\hookrightarrow$ If $b = 1$ return m, else $\perp$

Theorem 6. *The encryption scheme $PKE_{McE,2}$ is IND-CPA secure, given that both the McEliece assumption and the LPNDP hold. In particular, let $\mathcal{A}$ be an IND-CPA adversary against $PKE_{McE,2}$. Then there is a distinguisher $\mathcal{B}$ for Goppa codes and a distinguisher $\mathcal{D}$ for the LPNDP, such that for all $\lambda \in \mathbb{N}$*

$$\mathsf{Adv}_{\mathcal{A}}^{CPA}(\lambda) \leq \mathsf{Adv}_{\mathcal{D}}^{LPNDP_\theta(2n,l)}(\lambda) + 2 \times \mathsf{Adv}_{\mathcal{B}_R,G_R}^{ind}(\lambda).$$

The original proof can be found in [25]. However, the authors did not explicitly state the advantage of the adversary.

Proof. **Game 1** This is the DRE IND-CPA game.

Game 2 Same as Game 1, except that the generator matrix G_R within the public key is replaced by uniformly random matrix $U_R \in \{0,1\}^{l \times n}$. Therefore, the receiver public key in Game 2 is $pk_R := (U_R, t)$.

Any distinguisher $\mathcal{A}_R$ distinguishing between Game 1 and Game 2 yields a distinguisher $\mathcal{B}_R$ for a random irreducible Goppa code from a random linear code. Therefore,

$$\mathsf{Adv}_{\mathcal{A}}^{CPA} \leq \mathsf{Adv}_{\mathcal{A},\text{Game 2}}^{CPA} + \mathsf{Adv}_{\mathcal{B}_R,G_R}^{ind}(\lambda)$$

Game 3 Same as Game 1, except that the generator matrix G_S within the public key is replaced by uniformly random matrix $U_S \in \{0,1\}^{l \times n}$. Therefore, the sender public key in Game 3 is $pk_S := (U_S, t)$.

Any distinguisher $\mathcal{A}_S$ distinguishing between Game 2 and Game 3 yields a distinguisher $\mathcal{B}_S$ for a random irreducible Goppa code from a random linear code. Therefore, w.l.o.g

$$\mathsf{Adv}_{\mathcal{A}}^{CPA} \leq \mathsf{Adv}_{\mathcal{A},\text{Game 3}}^{CPA} + 2 \times \mathsf{Adv}_{\mathcal{B}_R,G_R}^{ind}(\lambda) \tag{3}$$

Game 4 Instead of computing the challenge ciphertext as

$$c^* = ([s|m_b]U_S \oplus e_S, [s|m_b]U_R \oplus e_R)$$

the challenger chooses $c^* = (c_1, c_2)$ with $c_1, c_2 \overset{R}{\leftarrow} \mathcal{U}_n$ instead.

The indistinguishability of Game 4 from Game 3 is shown as follows.

- Observe $\forall i \in \{R, S\}$ that $U_i^T = \left(U_{i,1}^T | U_{i,2}^T\right)$ with $U_{i,1}^T \in \{0,1\}^{l_1 \times n}$ and $U_{i,2}^T \in \{0,1\}^{l_2 \times n}$ s.t. $l_1 + l_2 = l$.
- Then $\forall i \in \{R, S\}$ the ciphertext can be transformed as $[s|m_b]U_i \oplus e_i = (s \cdot U_{i,1} \oplus e_i) \oplus m \cdot U_{i,2}$

Thus, the ciphertext can be transformed into:

$$\begin{aligned} c^* &= ([s|m_b]U_S \oplus e_S, [s|m_b]U_R \oplus e_R) \\ &= \Big((s \cdot U_{S,1} \oplus e_S) \oplus m_b \cdot U_{S,2}, (s \cdot U_{R,1} \oplus e_R) \oplus m_b \cdot U_{R,2}\Big) \end{aligned}$$

Firstly, set the matrices $U_1 = (U_{S,1}|U_{R,1}) \in \{0,1\}^{l_1 \times 2n}$ and $U_2 = (U_{S,2}|U_{R,2}) \in \{0,1\}^{l_2 \times 2n}$. Secondly, summarize the error vectors into one, i.e. $e = (e_s|e_R) \in \{0,1\}^{2n}$ The ciphertext is now:

$$c^* = ((s \cdot U_1 \oplus e) \oplus m_b \cdot U_2)$$

Finally, we can interpret $(s \cdot U_1 \oplus e)$ as an instance of the LPNDP and replace by a random value $u \overset{R}{\leftarrow} \mathcal{U}_{2n}$. The random vector u acts as a OTP s.t. the ciphertext is transformed into a uniformly distributed random vector:

$$c^* = (u \oplus m_b \cdot U_2)$$

This is the challenge ciphertext used in Game 4. The advantage of the original DRE IND-CPA adversary $\mathcal{A}$ is now 0, as the succeeding probability is $\frac{1}{2}$. The indistinguishability follows from the hardness of the LPNDP.

If the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has non-negligibly different succeeding probabilities in Games 3, 4 then we can use this adversary to solve any LPNDP. To this end we can use the following distinguisher D for a given LPNDP oracle $\mathcal{O}$, which is either $\mathcal{Q}_{s,\theta}$ with $s \in \{0,1\}^l$ or $\mathcal{U}_{l+1}$.

(1) Generate the public keys U_R, U_S for $\mathcal{A}$ in 2 steps. Remember, that $U_R = (U_{R,1}|U_{R,2})$ (resp. U_S).
   - Call the LPNDP oracle for enough samples $(a_1, b_1), \cdots, (a_{2n}, b_{2n}) \leftarrow \mathcal{O}$.
   - Set $b_R = (b_1|\cdots|b_n)$ and $b_S = (b_{n+1}|\cdots|b_{2n})$.
   - Set $U_{R,1} = (a_1|\cdots|a_n)$ and $U_{S,1} = (a_{n+1}|\cdots|a_{2n})$.
   - Sample the remaining part of the public key uniformly random, i.e. $U_{R,2} \overset{R}{\leftarrow} \{0,1\}^{l_2 \times n}$ (resp. $U_{S,2}$).

   Finally, the public keys are

$$\begin{aligned} pk_R &= U_R = (U_{R,1}|U_{R,2}) \in \{0,1\}^{l \times n} \\ pk_S &= U_S = (U_{S,1}|U_{S,2}) \in \{0,1\}^{l \times n} \end{aligned}$$

(2) $(m_0, m_1) \leftarrow \mathcal{A}_1(U_R, U_S)$

(3) $b \overset{R}{\leftarrow} \{0,1\}$

(4) Set the challenge ciphertext to

$$c^* = (c_1, c_2) = ((b_R \oplus m_b \cdot U_{R,2}), (b_S \oplus m_b \cdot U_{S,2}))$$

(5) $b' \leftarrow \mathcal{A}_2(U_R, U_S, c^*)$

(6) If $b' = b$ then return 1, else return 0.

If $\mathcal{O} = \mathcal{Q}_{\mathbf{s},\theta}$, then we have the same situation as in Game 3, else $\mathcal{O} = \mathcal{U}_{l+1}$ and we have the same situation as in Game 4. Therefore:

$$\mathsf{Adv}^{CPA}_{\mathcal{A},Game3} \leq \mathsf{Adv}^{CPA}_{\mathcal{A},\text{Game 4}} + \mathsf{Adv}^{LPNDP_{\theta}(2n,l)}_{\mathcal{D}}(\lambda) = \mathsf{Adv}^{LPNDP_{\theta}(2n,l)}_{\mathcal{D}}(\lambda)$$

This concludes that the overall advantage is

$$\mathsf{Adv}^{CPA}_{\mathcal{A}} \leq \mathsf{Adv}^{LPNDP_{\theta}(2n,l)}_{\mathcal{D}}(\lambda) + 2 \times \mathsf{Adv}^{ind}_{\mathcal{B}_R,G_R}(\lambda)$$

□

The following theorem 7 was already implicitly shown in [33] by reducing it to the property of verifiability of a verifiable k-repetition PKE from [25].

Theorem 7. *The encryption scheme* $(\mathtt{gen}_{McE,2}, \mathtt{enc}_{McE,2}, \mathtt{dec}_{McE,2})$ *satisfies DRE-soundness.*

The definition of the soundness property of DRE can be found in Appendix B.4.

Proof. Consider the case $\mathtt{dec}(pk_S, sk_R, c) = \perp$ and $\mathtt{dec}(pk_R, sk_S, C) = m$. We will prove by contradiction that this case never happens. Parse C as $(\mathsf{c}_R, \mathsf{c}_S)$ and $pk_R = \mathsf{G}_R$ and $pk_S = \mathsf{G}_S$, which ultimately have the following form due to being textbook McEliece ciphertexts.

$$\mathsf{c}_R = m'\mathsf{G}_R \oplus \mathsf{e}_R$$
$$\mathsf{c}_S = m\mathsf{G}_S \oplus \mathsf{e}_S$$

From $\mathtt{dec}(pk_S, sk_R, c) = \perp$ it follows that the verification step has failed. This means that after recovering m' from c_R by $\mathtt{dec}_{McE}(sk_R, \mathsf{c}_R) = m'$ the hamming distance of $m'\mathsf{G}_S$ has to be greater or equal than t to c_S. Considering that $m'\mathsf{G}_S \oplus \mathsf{c}_S = m'\mathsf{G}_S \oplus m\mathsf{G}_S \oplus \mathsf{e}_S$ we get

$$wgt(m'\mathsf{G}_S \oplus m\mathsf{G}_S \oplus \mathsf{e}_S) \geq t$$

From this it follows that $m' \neq m$ due to e_S being guaranteed to have the hamming weight $wgt(\mathsf{e}_S) < t$ by the syndrome decoding algorithm within the textbook McEliece decryption.
However, from $\mathtt{dec}(pk_R, sk_S, c) = m$ it follows that

$$wgt(m\mathsf{G}_R \oplus m'\mathsf{G}_R \oplus \mathsf{e}_R) < t$$

Now $m\mathsf{G}_R$ and $m'\mathsf{G}_R$ are codewords for $m \neq m'$ and therefore are guaranteed to have hamming distance $d(m\mathsf{G}_R, m'\mathsf{G}_R) \geq 2t+1$. This contradicts with $wgt(m\mathsf{G}_R \oplus m'\mathsf{G}_R \oplus \mathsf{e}_R) < t$ as $wgt(\mathsf{e}_R) < t$. Therefore, this case is not possible.
The same considerations will yield that the case $\mathtt{dec}(pk_S, sk_R, c) = m_R$ and $\mathtt{dec}(pk_R, sk_S, c) = m_S$ with $m_R \neq m_S$ is impossible.
Conclusively, $\mathbb{P}[\mathsf{Exp}^{\text{sound}}_{\mathcal{A},\Pi} = 1] = 0$. □

F.3 IND-CPA DRE via LWE-Based Binding Encryption [38]

Let $PKE_{\mathrm{LBE},2}$ be a binding encryption scheme from [38] with the restriction of having two receivers, where one is the sender. The authors prove that this DRE scheme satisfies IND-CPA security and the notion of *strong decryption consistency*. However, if we have only two public keys, sender and receiver, the experiment for strong decryption consistency becomes identical to the soundness experiment from [19]. Therefore, $PKE_{\mathrm{LBE},2}$ is also a dual-receiver encryption scheme and as such it can be used to realize SB-CPA. In fact, from the perspective of size-efficiency of the ciphertext, $PKE_{\mathrm{LBE},2}$ is so far the most efficient LWE-based CPA secure DRE construction. The other works either directly construct a less efficient CCA2 secure DRE [49, 34] or concentrate on IND-ID-CPA-secure IBE-DRE constructions [49, 34, 35]. Moreover, none of these works prove the soundness property of DRE introduced by [19].

Note, that $PKE_{\mathrm{LBE},2}$ does not directly surpass prior standard model CCA2 lattice-based constructions in terms of efficiency. Recently, Boyen et al. [10] presented an efficient lattice-based CCA2 secure KEM construction in the standard model, which the authors compare to other efficient constructions from [37] and conclude that their construction surpasses these in efficiency, mainly by not requiring signatures or MACs.

While the LWE-based CPA secure DRE $PKE_{\mathrm{LBE},2}$ may be inferior in terms of efficiency to [10, 37], we would like to point out that Boyen et al. [10] do not base their KEM on plain LWE but rather on SISnLWE, which they show to be reducible to LWE but do not provide a discussion about the tightness of the reduction.

G Theoretic Classification of Security Notions

This appendix contains investigations of the relationship between different types of game-based security notions. The implications between IND-SB-CPA and TBE security notions has already been examined in Section 6. Here we firstly consider the connection between IND-gtag-wCCA and IND-stag-wCCA in Appendix G.1 and between IND-SB-CPA and classic PKE notions in Appendix G.2 before we attempt a more holistic classification of game-based security notions in Appendix G.3.

G.1 Relation between IND-gtag-wCCA and IND-stag-wCCA

In this appendix we analyze the implicational relationships between IND-gtag-wCCA and IND-stag-wCCA security.

Lemma 7. *IND-stag-wCCA* $\Rightarrow$ *IND-gtag-wCCA.*

Proof. Let $(\mathtt{gen}, \mathtt{enc}, \mathtt{dec})$ be a TBE scheme. Under assumption of an efficient adversary $\mathcal{A}_{\text{gtag-wCCA}}$ with non-negligible probability to win the IND-gtag-wCCA security game, it is very straight forward to construct an efficient adversary

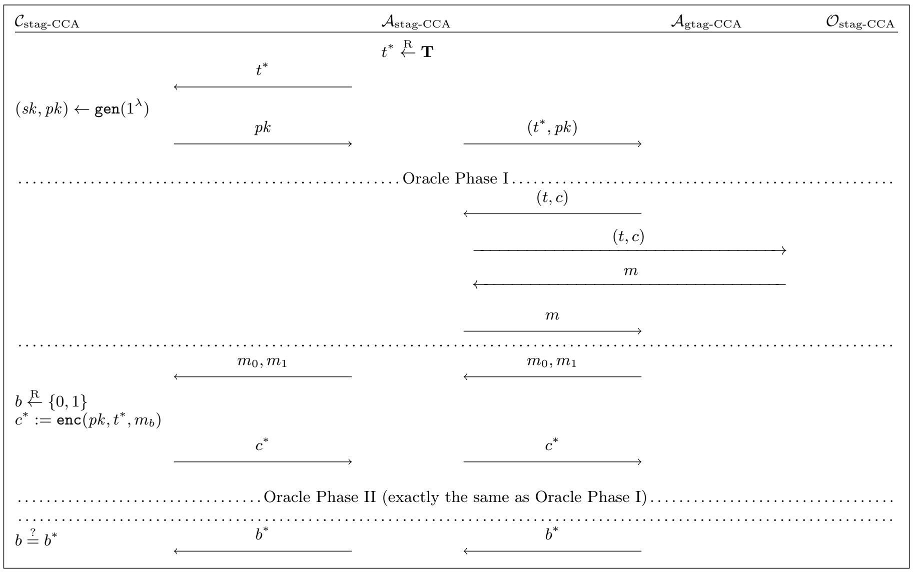
$\mathcal{C}_{\text{stag-CCA}}$
$\mathcal{A}_{\text{stag-CCA}}$
$\mathcal{A}_{\text{gtag-CCA}}$
$\mathcal{O}_{\text{stag-CCA}}$
$t^* \xleftarrow{R} \mathbf{T}$
t^*
$(sk, pk) \leftarrow \texttt{gen}(1^\lambda)$
pk
(t^*, pk)
Oracle Phase I
(t, c)
(t, c)
m
m
m_0, m_1
m_0, m_1
$b \xleftarrow{R} \{0, 1\}$
$c^* := \texttt{enc}(pk, t^*, m_b)$
c^*
c^*
Oracle Phase II (exactly the same as Oracle Phase I)
$b \stackrel{?}{=} b^*$
b^*
b^*

Fig. 16. Reduction for IND-stag-wCCA ⇒ IND-gtag-wCCA

$\mathcal{A}_{\text{stag-wCCA}}$ who has the same success probability in the IND-stag-wCCA game: An overview of the construction can be found in Figure 16.

The adversary $\mathcal{A}_{\text{stag-wCCA}}$ firstly draws a challenge tag t^* at random (just like the IND-gtag-wCCA challenger would do) and sends it to the challenger $\mathcal{C}_{\text{stag-wCCA}}$. The challenge tag is also given to $\mathcal{A}_{\text{gtag-wCCA}}$ together with the challenge key pk from $\mathcal{C}_{\text{stag-wCCA}}$. Afterwards $\mathcal{A}_{\text{stag-wCCA}}$ just forwards all messages between challenger, oracle and $\mathcal{A}_{\text{gtag-wCCA}}$. If the adversary $\mathcal{A}_{\text{gtag-wCCA}}$ wins, so will $\mathcal{A}_{\text{stag-wCCA}}$. □

On the other hand we find that an implication from IND-gtag-wCCA to IND-stag-wCCA is not true in general. For polynomially sized tag spaces, i.e. when $\frac{1}{|\mathbf{T}|}$ is non negligible, however, we find that IND-gtag-wCCA and IND-stag-wCCA are equivalent.

Lemma 8. *IND-stag-wCCA $\not\Leftarrow$ IND-gtag-wCCA.*

Proof. Let $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ be an IND-stag-wCCA secure TBE scheme with tag space $\mathbf{T} := \{0, 1\}^\lambda$. Now consider the punctured scheme

$$\texttt{Gen} := \texttt{gen}$$

$$\texttt{Enc}(pk, t, m) := \begin{cases} m & , t = 0^\lambda \\ \texttt{enc}(pk, t, m) & , \text{ else} \end{cases}$$

$$\texttt{Dec}(sk, t, c) := \begin{cases} c & , t = 0^\lambda \\ \texttt{dec}(sk, t, c) & , \text{ else.} \end{cases}$$

Since $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ is IND-stag-wCCA and by Lemma 7 in particular IND-gtag-wCCA secure, and the probability $\mathbb{P}[t^* = 0^\lambda] = \frac{1}{2^\lambda}$ is negligible, the resulting scheme $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ will still be IND-gtag-wCCA secure. It does not, however, satisfy IND-stag-wCCA security anymore, as the adversary $\mathcal{A}_{\text{stag-wCCA}}$ can choose $t^* = 0^\lambda$ in this case and win with probability one. □

Lemma 9. *IND-stag-wCCA* $\Leftarrow$ *IND-gtag-wCCA if* $\frac{1}{|\mathbf{T}|}$ *is non-negligible.*

Proof. Let $\mathcal{A}_{\text{stag-wCCA}}$ be an adversary with non-negligible success probability to win the IND-stag-wCCA game. We construct an adversary $\mathcal{A}_{\text{gtag-wCCA}}$ as follows: In the first step $\mathcal{A}_{\text{gtag-wCCA}}$ receives challenge tags $t^*_{\mathcal{C}}$ and $t^*_{\mathcal{A}}$ from $\mathcal{C}_{\text{gtag-wCCA}}$ and $\mathcal{A}_{\text{stag-wCCA}}$ respectively. If $t^*_{\mathcal{C}} \neq t^*_{\mathcal{A}}$ the adversary $\mathcal{A}_{\text{gtag-wCCA}}$ aborts. Otherwise it continues to just forward messages between challenger, oracle and $\mathcal{A}_{\text{gtag-wCCA}}$. Hence $\mathcal{A}_{\text{gtag-wCCA}}$ has success probability $\frac{1}{|\mathbf{T}|} \cdot \mathbb{P}[\mathcal{A}_{\text{stag-wCCA}} \text{ wins}]$ which is non-negligible. □

G.2 Relation between IND-SB-CPA and Classic PKE Notions

In this appendix look at the implications between IND-SB-CPA and classical IND notions ranging from CPA to CCA2. This highlights a certain skewness between the new IND-SB-CPA and classical PKE security notions which we try to understand a bit better in Appendix G.3.

The standard games from IND-CPA to IND-CCA2 are commonly defined for PKE schemes where encryption depends only on the keys of the decrypting/receiving party. Hence they only deal with one pair of keys which is used to encrypt and decrypt the challenge. For SBE, another (encrypting/sending) party and their credentials need to be fixed. This gives us a degree of freedom of when as well as by whom this is chosen.

In principle, there are (at least) four different options: The sender and its credentials are chosen...

(1) ...randomly by the challenger, right before encrypting the challenge.
(2) ...randomly by the challenger at the start of the game.
(3) ...by the adversary at the start of the game.
(4) ...by the adversary, together with the challenge messages.

We already know this degree of freedom from IBE schemes, where different notions exist according to when the receiver is chosen (cp. Remark 6 in Appendix E).

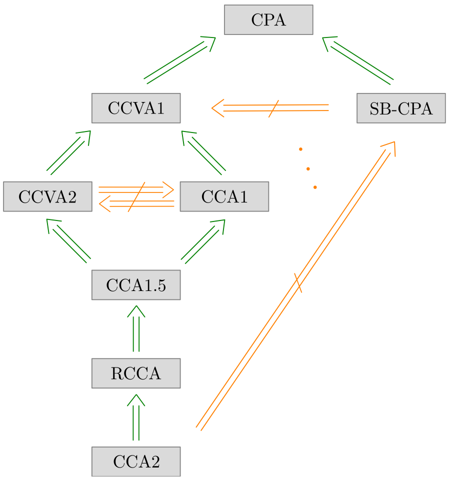

Fig. 17. Relationship to PKE Notions

These give rise to the obvious implications (4)⇒(3)⇒(2)⇒(1) for any notion. Note that by "choosing" we mean that all public credentials are handed from the choosing to the non-choosing party. Of course the values may actually be fixed earlier if the choosing party decides to do so.

To intuitively compare other games to IND-SB-CPA we will stay within the same mode of choosing the sender which our definition of IND-SB-CPA in Section 2 presents. I.e. that the challenger randomly chooses the sender together with the receiver at the start of the game ((2) in the list above).

Figure 17 gives an overview of the implications between IND-SB-CPA and common game-based security notions. We will now examine these (non)-implications in more detail. Note that these (non)-implications work equally well in most of the other sender choosing modes listed above, as long as IND-SB-CPA is also adapted to this mode.

Lemma 10 (SB-CPA ⇒ CPA). *Any IND-SB-CPA secure SBE scheme automatically satisfies IND-CPA security.*

Since it is obvious that IND-CPA is just IND-SB-CPA without any access to oracles, the adversary is strictly less powerful in this case and the implication rather trivial. We will therefore refrain from a formal proof at this point and move on to more interesting cases:

Lemma 11 (SB-CPA ⇏ CCVA1). *There is an IND-SB-CPA secure SBE scheme which does not satisfy indistinguishability under non-adaptive chosen ciphertext verification attack (IND-CCVA1) security.*

In this case we will conduct a proof by example and construct such a scheme from an IND-sID-CPA secure IBE scheme $(\texttt{gen}, \texttt{f}_{\text{ID}}, \texttt{ext}, \texttt{enc}, \texttt{dec})$. Consider the example $(\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ we give as a generic IND-SB-CPA secure SBE construction in Appendix E.3:

$$\begin{aligned}
&\texttt{Gen} := \texttt{gen} \text{ (i.e. } (SK, PK) := (msk, mpk))\\
&\texttt{Enc} : (PK_R, S, m) = (mpk_R, S, m) \mapsto \texttt{enc}(mpk_R, \texttt{f}_{\text{ID}}(S), m)\\
&\texttt{Dec} : (SK_R, S, c) = (msk_R, S, c) \mapsto \texttt{dec}(\texttt{ext}(msk_R, \texttt{f}_{\text{ID}}(S)), c).
\end{aligned}$$

We now modify its encryption and decryption algorithm such that the scheme $(\texttt{Gen}, \texttt{Enc}^*, \texttt{Dec}^*)$ still satisfies IND-SB-CPA security but is not IND-CCVA1 secure. The intuition behind this construction is the following: Each ciphertext is concatenated with a string of bits. For the generic use of honestly encrypting and decrypting messages this string is just zeros and completely uninteresting. The decryption algorithm is defined in such a way, however, that it still succeeds when part of this string of zeros is replaced by an (equally long) prefix of the secret key used for decryption. Given a verification oracle it is now fairly easy to extract this secret key by testing if a ciphertext is still valid when the string of zeros is modified bit by bit. Since the secret key used for decryption is specific to the receiver *and* the sender of the message in this particular IBE construction, the IND-SB-CPA oracles can not be used to extract any secrets, as they do not pertain to communication between the sender and receiver used for the challenge ciphertext.

More formally let k be the length of user secret keys usk when they are binary encoded and let $pr : \{0,1\}^k \times \{0,1\}^k \to \{\mathsf{true}, \mathsf{false}\}$ be defined by:

$$pr((x_1 \cdots x_k), (y_1 \cdots y_k)) = \mathsf{true}$$
$$:\Leftrightarrow \exists k_0 \in \{0, \dots, k\} : y_i = \begin{cases} x_i, & i \leq k_0 \\ 0, & \text{otherwise} \end{cases}.$$

With this function we can now (remembering $\mathtt{Gen} = \mathtt{gen}$, i.e. $(SK, PK) = (msk, mpk)$) define

$$\mathtt{Enc}^* : (PK_R, S, m) = (mpk_R, S, m) \mapsto (\mathtt{enc}(mpk_R, \mathtt{f}_{\mathrm{ID}}(S), m) \| 0^k)$$
$$\mathtt{Dec}^* : (SK_R, S, C) = (msk_R, S, (c\|x)) \mapsto \begin{cases} \mathtt{dec}(usk_S, c), & pr(usk_S, x) \\ \perp, & \text{otherwise} \end{cases}$$

where $usk_S := \mathtt{ext}(msk_R, \mathtt{f}_{\mathrm{ID}}(S))$ is defined as before and $x = (x_1 \cdots x_k) \in \{0,1\}^k$.

Lemma 12. $(\mathtt{Gen}, \mathtt{Enc}^*, \mathtt{Dec}^*)$ *is IND-SB-CPA secure.*

Proofsketch. The reduction to IBE-IND-CPA security of the underlying IBE-scheme is largely identical to Appendix E.3. The only difference is that for any oracle query $(mpk_{R'}, S', C)$ with $C = (c\|x)$, the adversary $\mathcal{A}_{\text{IBE-CPA}}$ only decrypts c if $pr(usk_{S'}, x)$ is true. Otherwise it hands back $\perp$. □

Lemma 13. $(\mathtt{Gen}, \mathtt{Enc}^*, \mathtt{Dec}^*)$ *is not IND-CCVA1 secure.*

Proof. Let $\mathcal{A}_{\text{IND-CCVA1}}$ be an adversary which queries the oracle in the following manner: Set

$$c \leftarrow \mathtt{Enc}(PK_R, S, m) = \mathtt{enc}(mpk_R, \mathtt{f}_{\mathrm{ID}}(S), m)$$

for some arbitrary message m and define queries

$$\forall i \in \{1, \dots, k\} : Q_i := (mpk_R, S, c\|x_1 \cdots x_{i-1}\|1\|0^{k-i})$$

where $x_i := 1$ if the oracles response to query Q_i was true and $x_i := 0$ otherwise. The user secret key usk_S is thus obtained as $usk_S = x_1 \cdots x_k$ from the oracle's responses and can be used to successfully decrypt any challenge ciphertext c^*. □

The proof of Lemma 11 follows directly from Lemmas 12 and 13. Note that by transitivity of implication this yields in particular that IND-SB-CPA does not imply any of the stronger security notions either.

Lemma 14 (SB-CPA $\not\Leftarrow$ CCA2). *IND-CCA2 security of an SBE scheme does not imply IND-SB-CPA security.*

Proof. This again is rather easy to see by example: Let $(\texttt{gen}, \texttt{enc}, \texttt{dec})$ be any standard IND-CCA2 secure encryption scheme, where ciphertexts are independent of the sender S. The adversary $\mathcal{A}_{\text{SB-CPA}}$ can query the oracle $\mathcal{O}_{\text{SB-CPA}}$ in the second oracle phase with input (pk_R, P, c^*) for any $P \neq S$ and obtain the content of the challenge ciphertext c^*. □

Note, again, that by transitivity of implication this means IND-SB-CPA is not implied by any of the weaker security notions either.

G.3 Attempting a More Holistic Classification

As seen in Appendix G.2, there seems to be a certain skewness between the new IND-SB-CPA and classical PKE security notions which we would like to understand a bit better. In this appendix we discuss a more holistic classification of game-based security notions for encryption schemes. We will then go on to indicate how IND-SB-CPA fits into this classification and that it is rather related to symmetric IND-CPA security. As throughout the rest of this paper we will call an encrypting party "sender" and a decrypting party "receiver". We choose these terms not only because secure message transmission is the obvious use case of encryption but borrowing terms from this setting also enhances readability of our explanations.

We consider game-based notions to generally be of the form

$$\mathsf{goal}\text{-}\mathsf{set}\text{-}\mathsf{scp}\text{-}\mathsf{pow},$$

where $\mathsf{goal} \in \{\text{IND, NM, RoR,}\dots\}$ denotes the adversary's goal and $\mathsf{pow} \in \{\text{CPA, CCVA1, RCCA, CCA2,}\dots\}$ denotes the adversary's power (restrictions), which—together with scp—determine the provision of oracles. In addition to these classically present parameters, we need set and scp to provide more degrees of freedom to accurately capture the wide range of notions.

The parameter set will capture the setting in which the game is conducted. This includes all parties (implying their public information) and other parameters which need to be fixed as well as who (challenger or adversary) chooses them and when they are chosen. The classic notions like IND-CPA use

$$\mathsf{set} = (\mathcal{C}, \text{Start}, \text{Receiver})$$

indicating that the only fixed party is the receiver who is chosen at the start of the game by the challenger. Note that, again, we mean this to imply that all public knowledge about these parties (e.g. the receiver's public key) is handed to the non-choosing party (e.g. the adversary) at the indicated time (e.g. "Start").

The variable scp indicates which scope of communication the adversary's knowledge restriction (indicated by pow) pertains to. This will definitely include the concrete configuration (S, R) of the challenge ciphertext, but may also include related configurations, like ciphertexts encrypted for the sender rather than receiver or for the same receiver but encrypted by different senders. For scp we will exemplarily introduce the terms *symmetric*, *directed*, *omnidirected* and *undique* for now:

(1) *Directed* scope means the adversary's power is only restricted with respect to the challenge configuration $(S \to R)$.
(2) *Symmetric* scope indicates restriction regarding the challenge configuration $(S \to R)$ as well as the reverse configuration $(R \to S)$. This is the scope used in IND-SB-CPA.
(3) *Omnidirected* scope includes all configurations going out from the chosen sender, i.e. $(S \to P)$ for arbitrary parties P. While this is not commonly considered for encryption schemes, it is the default setting for signatures.
(4) *Undique* scope can be viewed as the reverse of omnidirection: It includes all configurations incoming to the chosen receiver, i.e. $(P \to R)$ for arbitrary P. This scope is commonly used for PKE schemes where encryption and decryption are independent of the sender.

Note that there are many other instantiations of scp which might result in useful security notions. Outside of the indicated scope, we assume the adversary to have perfect power/knowledge. Within the scope scp, the adversary's power is determined by the parameter pow. We do not differentiate whether the adversary naturally has the required power—e.g. because they choose and therefore know a secret key or because only public knowledge is necessary—or whether it is provided in the form of an oracle.

For the relationship between the different scopes it is rather obvious that any scheme which is $\mathsf{goal\text{-}set\text{-}scp\text{-}pow}$ secure for $\mathsf{scp} \in \{$symmetric, omnidirected, undique$\}$ also satisfies $\mathsf{goal\text{-}set}$-directed-pow. But any implications between the three stronger scopes are not trivially given.

From this interpretation of game-based security notions the skewness between IND-SB-CPA and commonly considered PKE games like IND-CCA2 becomes intuitively obvious: We compare a symmetrically scoped notion with undique ones. I.e. we can not expect any implications between the security notions if there are no implications between the scopes. We also see that within this model IND-SB-CPA security is equivalent to classical symmetric key IND-CPA security. They are both of the form

$$\mathsf{goal\text{-}set\text{-}scp\text{-}pow} = \text{IND-}(\mathcal{C}, \text{Start}, (S, R))\text{-symmetric-CPA}.$$

Analyzing this further might be an interesting future question.