A Constructive Perspective on Signcryption Security*

Christian Badertscher, Fabio Banfi, and Ueli Maurer

Department of Computer Science, ETH Zurich, Switzerland
christian.badertscher@inf.ethz.ch
fabio.banfi@inf.ethz.ch
maurer@inf.ethz.ch

Abstract. Signcryption is a public-key cryptographic primitive, originally introduced by Zheng (Crypto '97), that allows parties to establish secure communication without the need of prior key agreement. Instead, a party registers its public key at a certificate authority (CA), and only needs to retrieve the public key of the intended partner from the CA before being able to protect the communication. Signcryption schemes provide both authenticity and confidentiality of sent messages and can offer a simpler interface to applications and better performance compared to generic compositions of signature and encryption schemes.
Although introduced two decades ago, the question which security notions of signcryption are adequate in which applications has still not reached a fully satisfactory answer. To resolve this question, we conduct a constructive analysis of this public-key primitive. Similar to previous constructive studies for other important primitives, this treatment allows to identify the natural goal that signcryption schemes should achieve and to formalize this goal in a composable framework. More specifically, we capture the goal of signcryption as a gracefully-degrading secure network, which is basically a network of independent parties that allows secure communication between any two parties. However, when a party is compromised, its respective security guarantees are lost, while all guarantees for the remaining users remain unaffected. We show which security notions for signcryption are sufficient to construct this kind of secure network from a certificate authority (or key registration resource) and insecure communication. Our study does not only unveil that it is the so-called *insider-security notion* that enables this construction, but also that a weaker version thereof would already be sufficient. This may be of interest in the context of practical signcryption schemes that do not achieve the stronger notions.
Last but not least, we observe that the graceful-degradation property is actually an essential feature of signcryption that stands out in comparison to alternative and more standard constructions that achieve secure communication from the same assumptions. This underlines the vital importance of the insider security notion for signcryption and strongly supports, in contrast to the initial belief, the recent trend to consider the insider security notion as the standard notion for signcryption.

1 Introduction

1.1 Motivation and Background

Signcryption is a public-key cryptographic primitive introduced by Zheng [Zhe97] in 1997, which simultaneously provides two fundamental cryptographic goals: *confidentiality* and *authenticity*. Intuitively, the first property ensures that no one except the intended recipient should be able to learn anything about a sent message, and this is typically achieved by means of an encryption algorithm, and the second property ensures that the receiver can verify that a message indeed originated from the claimed sender, which is typically achieved by employing a digital signature scheme. Signcryption is the public-key analogue of the better known symmetric-key primitive called *authenticated encryption* and shares part of its motivation: by merging the two security goals, one might gain practical efficiency and at the same time offer better usability to applications, since there is only a single scheme that needs to be employed.

* This is the full version of the article due to appear at SCN 2018. The final publication will be available at `link.springer.com`.

Since its introduction in 1997, several concrete schemes have emerged in the literature based on different hardness assumptions [Zhe97, ZI98, SZ00, LQ03, LQ04]. Also, new properties beyond the basic security goals have been introduced recently, such as identity-based [ML02, Boy03, LQ03, LBZ10, SVR10, SVVR12], hybrid [Den05], KEM-DEM-based [BD06], certificateless [BF08], verifiable [SVR10], attribute-based [PPB14, DDM15a], functional [DDM15b], or key invisible [WMAS13] signcryption schemes. But finding the basic (or initial) security definitions for signcryption proved to be a very subtle and challenging task. In fact, the original signcryption scheme by Zheng was formally proven secure only about ten years after its introduction by Baek, Steinfield, and Zheng [BSZ07]. While (symmetric) authenticated encryption was put on solid security definitions directly from the start (cf. [BN00]), the basic security notions for signcryption have had a more difficult path and converged to a set of commonly agreed notions only recently [YDZ10] and only thanks to the merits of a sequence of foundational works [An01, ADR02, BSZ07] that formally introduced what is now known as the outsider security model (to capture various network attacks) and the insider security model (to capture attacks of corrupted users).

Only little effort has subsequently been made to investigate what the security notions precisely mean and whether they provide the expected service to higher-level protocols. An initial approach to this question was taken in [GK07] where a functionality is presented that idealizes the process of using the signcryption algorithm to ensure unforgeability and confidentiality (focusing on the outsider security model) along the lines of the signature and public-key encryption functionality in the UC framework [Can01].

In this work, we significantly advance this line of research and provide a detailed application-centric analysis of the basic security notions of signcryption. Our novel view underlines the importance of insider security as a distinctive feature that indeed assigns signcryption a special significance in actual deployments of network protocols. We note that its importance has been (and still is) overlooked by a substantial fraction of works. In particular, our results contrast the line of previous works that propose, analyze and revisit signcryption schemes and their security, including [GK07, TP14, BSZ07], recent developments in practical lattice-based schemes [GM18], and one of the main references on the basic notions [YDZ10], that assign too little credit to the relevance of insider security. We believe that our analysis provides sufficient evidence that insider security has to be termed the standard notion for signcryption and that it identifies which of the proposed variants for insider security should be the preferred choice.

The fundamental question of signcryption security. There is one main reason why finding suitable definitions for signcryption has turned out to be a very intricate task: its application as part of public-key infrastructures. In such a scenario, users register their public-keys with a certificate authority, but are otherwise independent of each other, unlike in a setup with pre-shared keys for symmetric-key cryptography where any two parties possess the same key. To protect the communication, a user has to retrieve the public key of the communication partner from the certificate authority (or any other trusted source) and protects the communication using the signcryption scheme. The core question becomes:

What are the attacks we have to protect against in this setting?

In the process of finding an answer to this question two main notions emerged: the *outsider security model* and the *insider security model*. Roughly speaking, outsider security considers an attacker as being an outsider to the system, such as a network attacker or an adversarial entity that registers a public key with a certificate authority. Insider security additionally tries to retain security even when the attacker is an insider, for example an (a priori) legitimate user whose key got compromised. While both models seem reasonable, insider security is intuitively more appealing. However, only very recently, the idea of denoting insider security as the standard notion seems to become mainstream. This is why some existing works slightly underestimate the importance of insider security, including one of the main references on the basic notions [YDZ10, page 29]:

> [...], however, it might still seem that the distinction between insider and outsider security is a bit contrived, especially for privacy. [...]. Similarly for authenticity, if non-repudiation is not an

> issue, then insider security seems to make little sense [...]. Still, there are some cases where the extra strength of the insider security might be important.

or [YDZ10, page 46]:

> [...] the insider confidentiality model is under normal circumstances not of significant importance because it effectively assumes that the sender S is trying to decrypt (unsigncrypt) a signcryptext which was sent by herself. Thus, this model appears only useful in providing "forward secrecy," i.e., providing security under the special circumstances in which an adversary who breaks into S's system obtains her secret key in order to unsigncrypt a message previously signcrypted by S to R. As pointed out by Zheng in the full version of the original signcryption paper [...]. In view of this discussion, we believe that for most applications it suffices for a signcryption scheme to achieve confidentiality in the "multi-user outsider" model. [...] The signcryption literature is currently confused as to which of these models best represents multi-user insider confidentiality, with some papers preferring the weaker notion of insider security and some papers preferring the stronger notion. [...]

In this paper, we take a step towards resolving this confusion. We present a systematic way towards answering the question:

Which basic notion should a signcryption scheme fulfill and why?

We hope that the methodology that we put forward in this work will be applied to existing and future, more enhanced notions of signcryption security in order to resolve similar questions.

1.2 Our Analysis

Defining an application scenario. To answer the above question, we formalize the typical application of signcryption as a construction following the real-world/ideal-world paradigm: this means we have to specify what resources are available in the real world (e.g., a certificate authority or a network), we have to specify how the users in the real world employ a signcryption scheme to protect their communication, and finally, we have to specify what they achieve. This is captured by specifying an ideal world, where all desirable security properties are ideally ensured. The protocol is called secure if it constructs the ideal specification, i.e., if the real world (where parties execute the protocol) is as useful to an adversary as the ideal world, the latter world being secure by definition. Formally, one has to construct a simulator in the ideal world to make the worlds computationally indistinguishable.

In this work, the real world consists of the usual ingredients inspired by public-key infrastructures:

- An insecure network **Net**, where each user can register themselves with a unique identity and send and receive messages, and where a network attacker, say Eve, has full control over the network, including message delivery.
- A certificate authority **CA**, where users and the attacker Eve can register public keys in the name of the identity. The certificate authority only guarantees that there is exactly one value registered for an identity, but does not verify knowledge of, for example, a secret key.
- A memory resource **Mem** that models the storage of the secret values of each user. The storage is possibly compromised by an intruder, say Mallory, which models key compromise.

Defining the goal for signcryption. The security goal of signcryption can be identified in a very natural way: due to the nature of public-key cryptography, the security depends on which user gets compromised. Furthermore, in a public-key setting, in sharp contrast to the secret-key setting, parties are independent in principle. Hence, if a user is compromised, we have to give up his security: this means that messages sent to this user can be read by the attacker, and the attacker can act in the name of this user. This directly gives rise to a notion of a secure network that gracefully degrades depending on which users gets compromised as described below. We denote this gracefully-degrading secure network by **SecNT** and its main properties are as follows:

1. If two uncompromised legitimate users communicate, then the secure network guarantees that the network attacker learns at most the length of the messages and the attacker cannot inject any message into this communication: the communication between them can be called secure.
2. If, however, the legitimate sender is compromised, but not the receiver, then the network allows the attacker to inject messages in the name of this sender. Still, Eve does not learn the contents of the messages to the receiver: the communication is thus only confidential.
3. If, on the other hand, the legitimate receiver is compromised, but not the sender, the secure network allows Eve to read the contents of the messages sent to this compromised user. Still, no messages can be injected into this communication: the communication is only authentic.
4. If both, sender and receiver, are compromised, then the network does not give any guarantee on their communication, Eve can read every message and inject anything at will.

As a special case, we observe that if no user is compromised, we have a fully secure network between the users. And when a user is compromised, we lose just his respective guarantee. Our main technical result is the proof of the following theorem in the constructive cryptography framework.[1]

Theorem (informal). *If a signcryption scheme is secure in the multi-user outsider security model and in the multi-user insider security model as specified in Fig. 1, Fig. 2, and Fig. 3, then the associated protocol constructs a gracefully-degrading secure network from an insecure network and a certificate authority with respect to any number of compromised keys of legitimate users (and with respect to static security).*

If the signcryption scheme is secure in the multi-user outsider security model as specified in Fig. 1, then the secure network is constructed if no key of legitimate users is compromised.

1.3 Implications of our Analysis and the Importance of Insider Security

This work enriches the study on signcryption security by giving an additional, application-centric viewpoint for understanding the different notions in a composable setting. This paragraph provides the consequences of our main result:

The preferred insider security notion. Our analysis identifies the notions that imply the above construction and this provides confidence that the security games in Fig. 2 and Fig. 3 are an adequate choice to model game-based insider security. The notions we use are in particular implied by what is denoted in [YDZ10] as "multi-user insider confidentiality in the FSO/FUO-IND-CCA2 sense" and "multi-user insider unforgeable in the FSO/FUO-sUF-CMA sense", respectively. The presented games are, however, weaker forms of insider security, which has the advantage that it might be possible to construct more efficient schemes for this broader class. We discuss further definitions in Sect. 3.

Graceful degradation thanks to insider security. One crucial point of our main theorem is that it is insider security that provably assures that the secure network degrades gracefully as a function of compromised keys and does not lose the security guarantees in a coarse-grained fashion (for example per pair of parties instead of a single party). This view assigns a more crucial, practical role to the insider security model than what is commonly assumed.

Comparison with other constructions. By specifying the assumed resources and the desired goal, we can now ask the question whether there exist other natural schemes that achieve the same construction and to compare them. For example, in a recent work [FHH14], it is shown that universally composable, non-interactive key-exchange (NIKE) protocols realize a functionality that provides a shared key to each pair of (honest) users. This key can be used to protect the

[1] Our results are, however, not specific to the framework itself and developing our approach in another framework like Canetti's UC framework [Can01] would yield closely related findings (cf. [HMM15]).

session between any such pair by employing a (symmetric) authenticated-encryption scheme and is thus sufficient to realize a secure network. NIKE needs as a setup a certificate authority (as specified in our real world), and based on this setup, a shared secret key can be established with minimal communication and interaction between any two parties.[2] The schemes are in addition arguably practically efficient [ÇGP+13]. We hence observe that this would be a second method to achieve the same as signcryption does for the case when we only have a network attacker (i.e., no key is compromised). This second method based on NIKE schemes [FHKP13] and authenticated encryption [HKR15] is likely to outperform the signcryption schemes in terms of efficiency.

We point out that such comparisons help to identify the specific core use-cases of a cryptographic primitive that conceptually separates it from other primitives. In the context of signcryption, the above observation might suggest that the real benefit of introducing signcryption as a public-key primitive is to demand insider-security as the standard formal capability to limit the damage against insider attacks or key compromises.

Remark 1 (On the comparison to AEAD). We point out that the above comparison does not mean that the usage of AEAD is doomed to fail in achieving insider security. What we observe is that if insider security is not demanded from a signcryption scheme, then there seem to be alternative schemes that perform arguably better and achieve the same goal under the same assumptions. On the other hand, finding a mode of operation for AEAD (in combination with key exchange) that outperforms signcryption also in the case of key compromises is an interesting question for future work.

Modeling corruptions. Our composable security analysis considers *static corruptions*: static corruption means that a party either behaves honestly, or it is corrupted (or compromised) at the beginning of the protocol execution. During the execution, an honest party cannot adaptively become compromised. In our setting, the only secret information that a party carries is its secret key. Therefore, compromising the key fully captures corruptions in this setting, as it allows the attacker to entirely impersonate the party, i.e., reading all incoming messages, sending any message in the name of this party, and isolating the party by not delivering messages.

Our analysis puts forward a conceptual contribution of independent interest in this setting. Namely, to refrain from letting compromised parties be formally absorbed by the adversary. Instead we leave them operational as a stand-alone protocol machine and just assign enough power to the adversary to actually completely impersonate the party (via explicit storage corruptions in our concrete case). On one hand, our statements therefore still contain the typical *full corruption* case, but this modeling approach also enables to formally express security guarantees for *partially corrupted* parties. In fact, identifying reasonable partial corruption scenarios seems to be crucial in building formal models that are able to capture a range of real-world threats, such as bugs that result in key leakage [NSS+17, DLK+14], and to express which of the security guarantees can still be retained in the presence of a specific threat.

As explained in the previous section, assigning guarantees to partially corrupted parties is very vital to understand the benefit of insider security of signcryption: When a compromised party sends a message to an uncompromised receiver, we want to express that the attacker still does not learn the contents. If the compromised party was modeled as being completely absorbed by the adversary, then no such guarantee could be formally expressed.

Remark 2 (On adaptive corruptions and forward secrecy). The static corruption model is probably the most prominent corruption model when analyzing security of network protocols in simulation-based frameworks. Furthermore, achieving security against adaptive corruptions is substantially harder and even requires additional, non-standard, assumptions due to the so-called *commitment problem* of the simulator. The commitment problem is a standard technical problem in simulation-based security which also occurs in the context of signcryption. To illustrate the very basic problem,

[2] In case the NIKE scheme required a trusted CRS, distributing it could also be accomplished by the trusted CA.

assume Alice sends an encrypted message m to Bob over an authenticated channel (encrypted using Bob's public key). The goal is to construct a secure channel, i.e., an authenticated channel that leaks only the length of m to a network attacker as long as sender and receiver are not corrupted). A simulator in the ideal world, with access to the secure channel functionality, therefore needs to be able to simulate a ciphertext[3] without knowing the plaintext m. Later, if Bob gets (adaptively) corrupted, the simulator additionally has to simulate the compromised real-world decryption key sk_R (and any internal state) that correctly decrypts the previously made-up ciphertext c to the original message m. As shown by Nielsen [Nie02], if the message space is large compared to the key size, then this simulation problem cannot be solved consistently. In particular it is shown in [Nie02] that a public-key encryption scheme that can encrypt an unbounded number of messages with keys that do not change (and have a reasonable size) cannot achieve adaptive security in the standard model. To circumvent this problem, so-called *non-committing* schemes have been developed based on stronger assumptions such as programmable random oracles or some forms of synchronization or interaction between sender and receiver (cf. [CHK05] for an example in the context public-key encryption). An interesting research direction is to develop adaptively secure signcryption schemes since they would allow to construct a gracefully-degrading secure network under adaptive key-compromise attacks.

We further note that our analysis does not cover forward secrecy.[4] The reason for this is that standard signcryption security, and in particular the insider-security model, does not provide any guarantee with respect to forward secrecy. Again, the same considerations from standard public-key encryption apply here: once compromised, knowledge of the secret key allows to decrypt all past messages of that party. As we know from the design of forward-secure public-key encryption schemes as in [CHK05], or [CHK03], to achieve forward secrecy requires radically different schemes, for example schemes that admit stateful decryption (i.e., the decryption key needs to be updated as otherwise the above problem occurs that excludes forward secrecy). Here too, an interesting question is to formally define forward secrecy for signcryption schemes and to develop schemes that provably fulfill such a definition.

2 Preliminaries

2.1 Notation

We describe our systems with pseudocode using the following conventions: We write $x \leftarrow y$ for assigning the value y to the variable x. For a distribution $\mathcal{D}$ over some set, $x \twoheadleftarrow \mathcal{D}$ denotes sampling x according to $\mathcal{D}$. For a finite set X, $x \twoheadleftarrow X$ denotes assigning to x a uniformly random value in X. Typically queries to systems (for example a network) consist of a suggestive keyword and a list of arguments (e.g., $(\mathtt{send}, m, \mathsf{ID}_r)$ to send a message m to a receiver with identity ID_r). We ignore keywords in writing the domains of arguments, e.g., $(\mathtt{send}, m, \mathsf{ID}_r) \in \mathcal{M} \times \{0,1\}^*$ indicates that $m \in \mathcal{M}$ and $\mathsf{ID}_r \in \{0,1\}^*$. The systems generate a return value upon each query which is output at an interface of the system. We omit writing return statements in case the output is a simple constant whose only purpose is to indicate the completion of an operation.

For the sake of presentation, we assume throughout the paper that the message space is represented by $\mathcal{M} := \{0,1\}^k$ for some fixed (and globally known) integer $k > 0$.[5]

We conduct a concrete security treatment in this work and therefore omit the security parameter (and additional global domain parameters) as an additional input to the algorithms to simplify

[3] One very common strategy is that the simulator just encrypts a random message of the correct length.

[4] Recall that technically, adaptive security and forward secrecy are different properties: while forward secrecy tries to retain privacy of messages received *before* a corruption happened, adaptive security refers to simulation-based security with respect to a certain corruption model in composable frameworks. Although related in spirit, the two notions do not imply each other. We refer to [CHK05] and [CHK03] for further discussions.

[5] It is typically assumed that a message has an encoding as a bitstring. Therefore, we do not distinguish between a message and its encoding as an element in $\{0,1\}^k$ (and fixed-length is always achievable by an appropriate padding scheme).

notation. If needed, one can think of the experiments being indexed by a security parameter and efficiency and negligibility being defined with respect to this parameter. The algorithms and systems in this work are efficient with respect to the usual asymptotic notions.

2.2 Definition of Signcryption Schemes

We present the formal syntactic definition of Signcryption from [BSZ07]. For convenience, we assume that the global domain parameters of a scheme (including the security parameter or the description of a specific finite field $\mathbb{F}$ for the computations), are known.[6]

Definition 1 (Signcryption Scheme). *A signcryption scheme* $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ *for* key space[7] $\mathcal{K}$, message space $\mathcal{M}$, *and* signcryptext space $\mathcal{S}$, *is a collection of four (efficient) algorithms:*

- *A* sender key generation algorithm, *denoted* Gen_S, *which outputs a* sender key-pair (sk_S, pk_S), *i.e., the* sender private key $sk_S \in \mathcal{K}$ *and the* sender public key $pk_S \in \mathcal{K}$, *respectively. We write* $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S$.
- *A* receiver key generation algorithm, *denoted* Gen_R, *which outputs a* receiver key-pair (sk_R, pk_R), *i.e., the* receiver private key $sk_R \in \mathcal{K}$ *and the* receiver public key $pk_R \in \mathcal{K}$, *respectively. We write* $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R$.
- *A (possibly randomized)* signcryption algorithm, *denoted* $\mathsf{Signcrypt}$, *which takes as input a sender private key* sk_S, *a receiver public key* pk_R, *and a* message $m \in \mathcal{M}$, *and outputs a* signcryptext $s \in \mathcal{S}$. *We write* $c \leftarrow \mathsf{Signcrypt}(sk_S, pk_R, m)$.
- *A (usually deterministic)* unsigncryption algorithm, *denoted* $\mathsf{Unsigncrypt}$, *which takes as input a receiver private key* sk_R, *a sender public key* pk_S, *and a* signcryptext ("the ciphertext") $s \in \mathcal{S}$, *and outputs a* message $m \in \mathcal{M}$, *or a special symbol* $\perp$. *We write* $m \leftarrow \mathsf{Unsigncrypt}(sk_R, pk_S, s)$.

The correctness condition *requires that for all sender key pairs* (sk_S, pk_S) *in the support of* Gen_S, *and for all receiver key pairs* (sk_R, pk_R) *in the support of* Gen_R, *and for all messages* $m \in \mathcal{M}$ *it holds that*

$$\mathsf{Unsigncrypt}(sk_R, pk_S, (\mathsf{Signcrypt}(sk_S, pk_R, m)) = m.$$

2.3 Constructive Cryptography

Discrete Systems The basic objects in our constructive security statements are reactive discrete systems that can be queried by their environment: Each interaction consists of an input from the environment and an output that is given by the system in response. Discrete reactive systems are modeled formally by random systems [Mau02], and an important similarity measure on those is given by the distinguishing advantage. More formally, the advantage of a distinguisher $\mathbf{D}$ in distinguishing two discrete systems, say $\mathbf{R}$ and $\mathbf{S}$, is defined as

$$\Delta^{\mathbf{D}}(\mathbf{R}, \mathbf{S}) \;=\; \Pr[\mathbf{DR} = 1] - \Pr[\mathbf{DS} = 1],$$

where $\Pr[\mathbf{DR} = 1]$ denotes the probability that $\mathbf{D}$ outputs 1 when connected to the system $\mathbf{R}$. More concretely, $\mathbf{DR}$ is a random experiment, where the distinguisher repeatedly provides an input to one of the interfaces and observes the output generated in reaction to that input before it decides on its output bit.

[6] We therefore omit these parameters as explicit inputs to the algorithms and do not need to specify a special Setup algorithm that generates such global parameters for a scheme.

[7] For better readability, we assume that all the keys (both private and public) belong to the the same set, but clearly this need not to be the case. The definition could be easily modified accordingly.

Resources and converters. The central object in constructive cryptography is that of a resource available to parties, and the resources we discuss in this work are modeled by reactive discrete systems. As in general the same resource may be accessible to multiple parties, such as a communication channel that allows a sender to input a message and a receiver to read it, we assign inputs to certain *interfaces* that correspond to the parties: the sender's interface allows to input a message to the channel, and the receiver's interface allows to read what is in the channel. More generally, a resource is a discrete system with a finite set of interfaces $\mathcal{I}$ via which the resource interacts with its environment.

Converters model protocols used by parties and can attach to an interface of a resource to change the inputs and outputs at that interface. This composition, which for a converter π, interface I, and resource $\mathbf{R}$ is denoted by $\pi^I\mathbf{R}$, again yields a resource. In this work, a converter π is modeled as a system with two interfaces: the *inner interface* $\texttt{in}$ and the outer interface $\texttt{out}$. The inner interface can be connected to an interface I of a resource $\mathbf{R}$ and the outer interface then becomes the new interface I of resource $\pi^I\mathbf{R}$. For a vector of converters $\pi = (\pi_{I_1}, \ldots, \pi_{I_n})$ with $I_i \in \mathcal{I}$, and a subset of interfaces $\mathcal{P} \subseteq \{I_1, \ldots, I_n\}$, $\pi_{\mathcal{P}}\mathbf{R}$ denotes the resource where π_I is connected to interface I of $\mathbf{R}$ for every $I \in \mathcal{P}$. We write $\overline{\mathcal{P}} := \mathcal{I} \setminus \mathcal{P}$. Two special converters in this work are the identity converter $\mathbf{1}$, which does not change the behavior at an interface, and the converter $\mathbf{0}$, which blocks all interaction at an interface (no inputs or outputs).

For $\mathcal{I}$-resources $\mathbf{R}_1, \ldots \mathbf{R}_m$ the *parallel composition* $[\mathbf{R}_1, \ldots, \mathbf{R}_m]$ is again an $\mathcal{I}$-resource that provides at each interface access to the corresponding interfaces of all subsystems.[8]

In this paper, we make statements about resources with interfaces from the set $\mathcal{I} = \{\texttt{P}_1, \ldots, \texttt{P}_n, \texttt{M}_1, \ldots, \texttt{M}_n, \texttt{E}\}$. Interface $\texttt{P}_i$ can be thought of as being the access point of the ith honest party to the system. Interface $\texttt{M}_i$ is the access point of an intruder (i.e., a hypothetical attacker entity like Mallory), and $\texttt{E}$ is the access point of the network attacker Eve (also a hypothetical entity).

Formally, a *protocol* is a vector $\pi = (\pi_{I_1}, \ldots, \pi_{I_{|\mathcal{I}|}})$ that specifies one converter for each interface $I \in \mathcal{I}$. For the honest parties, this corresponds to the actions they are expected to execute (for example, encrypt to protect the content of a message). For the hypothetical attacker entities, the converter specifies their default behavior when no attack happens. Typically, for purely hypothetical entities such as a network attacker or the intruder, we assign the identity converter since they are not expected to perform additional tasks. However, the interfaces are possibly dishonest, which means that the default behavior is not necessarily applied, but replaced by an arbitrary, adversarial strategy that makes use of all potentially available capabilities (e.g., to inject messages into a network).

Filtered resources. Typically, one would like to specify that certain capabilities at an interface are only potentially available (e.g., to an attacker), but not guaranteed to be available (i.e, not a feature of a protocol). A typical example is that the leakage to the network attacker of a secure channel at interface $\texttt{E}$ is at most the length of the message $|m|$ (potentially available), but of course not guaranteed (there exist encryption schemes that hide the length of the message). To model such situation, constructive cryptography offers the concept called *filtered resources*. Let $\mathbf{R}$ be a resource and $\phi = (\phi_{I_1}, \ldots, \phi_{I_n})$ be a vector of converters. Then, the filtered resource $\mathbf{R}_\phi$ is a $\mathcal{I}$-resource, where for an honest party at interface I_j, the interaction through the converter ϕ_{I_j} is guaranteed to be available, while interactions with $\mathbf{R}$ directly is only potentially available to dishonest parties. The converter ϕ_{I_j} can be thought of as filtering or shielding certain capabilities of interface I_j of system $\mathbf{R}$, we hence denote ϕ as the filter. We refer the reader to [MR11] for more details and briefly mention that this concept has turned out to be useful in modeling cryptographic problems [HMM15].

The way we use filters in this work is as follows: we want to make security statements that depend on the set of compromised keys of honest parties. We model this in the real world with a memory functionality, where each party can store its own key. We model that this storage is

[8] Note that if the two interfaces do not have the same interface set, one can simply add "dummy interfaces" that do not take any input and output to lift them to identical interface sets. In this sense, we also allow resources with different interface sets here.

potentially unsafe, meaning that if an intruder is present at interface $\mathtt{M}_i$, he potentially gets the key. However, the memory does not guarantee that the key is leaked (e.g., if no intruder is present, no key is leaked at interface $\mathtt{M}_i$). The same idea is used to model the capabilities of the network attacker. This is also reflected in the ideal world, where a dishonest intruder (and the network attacker if present) can potentially get more power by removing the filter.[9]

Construction. A constructive security definition then specifies the goal of a protocol in terms of *assumed* (also known as hybrid functionalities) and *constructed* resources (ideal functionality). The goal of a protocol is to construct the ideal functionality from the given ones. We directly state the central definition of a construction of [MR11] and briefly explain the relevant condition.

Definition 2. *Let $\mathbf{R}_\phi$ and $\mathbf{S}_\psi$ be filtered resources with interface set $\mathcal{I}$ and let $\pi = (\pi_{I_1}, \ldots, \pi_{I_{|\mathcal{I}|}})$ be a protocol. Let further be $\mathcal{U} \subseteq \mathcal{I}$ be the set of interfaces with potentially dishonest behavior and let ε be a function that maps distinguishers to a value in $[-1, 1]$. The protocol π constructs $\mathbf{S}_\psi$ from $\mathbf{R}_\phi$ within ε and with respect to potentially dishonest $\mathcal{U}$, denoted by*

$$\mathbf{R}_\phi \quad \overset{(\pi, \varepsilon, \mathcal{U})}{\Longrightarrow} \quad \mathbf{S}_\psi,$$

if there exist converters $\sigma = (\sigma_{U_1}, \ldots, \sigma_{U_{|\mathcal{U}|}})$, $U_i \in \mathcal{U}$, such that for all (dishonest) subsets $\mathcal{C} \subseteq \mathcal{U}$ we have that

$$\Delta^{\mathbf{D}}(\pi_{\overline{\mathcal{C}}}\, \phi_{\overline{\mathcal{C}}}\mathbf{R},\, \sigma_{\mathcal{C}}\, \psi_{\overline{\mathcal{C}}}\mathbf{S}) \leq \varepsilon(\mathbf{D})$$

for any distinguisher $\mathbf{D}$.

The condition in Definition 2 ensures that for any combination of dishonest interfaces, whatever they can do in the assumed resource using the unfiltered capabilities, they could do as well with the constructed resource by applying the *simulators* σ_{U_i} to the respective (unfiltered) interfaces U_i of the ideal resource. Turned around, if the constructed resource is secure by definition (for example, a secure channel does potentially leak at most the length of a message), there is no successful attack on the protocol. The notion of construction is composable, which intuitively means that the constructed resource can be replaced in any context by the assumed resource with the protocol attached without affecting the security. We refer to [MR11, Mau11] for a proof. For readers more familiar with Canetti's UC Framework [Can01], we refer to [HMM15] for explanations of how the above concepts relate to similar concepts in UC. We refer to Fig. 4 for a graphical illustration of our main construction, for the case of two dishonest interfaces.

We are interested in concrete security statements and reductions in this work and typically $\varepsilon(\cdot)$ is the advantage of an adversary $\mathcal{A} = \rho(\mathbf{D})$ in a related security game (such as the outsider security game of signcryption) where $\rho(\cdot)$ is a mapping from distinguishers to adversaries, for example implemented by a black-box construction of such an adversary $\mathcal{A}$ from a distinguisher $\mathbf{D}$.

3 An Overview of Signcyrption Security

In this section we present the security definitions of signcryption which we use in this work. In the literature, two main models are defined:

– The *ADR model*: this model corresponds to the two-user setting (comprising both outsider and insider security, called TOS and TIS, respectively), and was extensively studied by An, Dodis, and Rabin in [ADR02];

[9] This concept can be seen as a variant of the following UC concept: in UC, a functionality is informed which party is corrupted and its behavior can depend on this corruption set (e.g., leaking inputs to parties that get corrupted to the simulator). The same is achieved using the concept of filters in constructive cryptography, where removing the filter uncovers potential information needed to simulate.

– The *BSZ model*: this model corresponds to the multi-user setting (comprising both outsider and insider security, called MOS and MIS, respectively), and was extensively studied by Baek, Steinfield, and Zheng in [BSZ07].

We focus on the multi-user model in this work and present security definitions for outsider security and insider security.[10]

3.1 Multi-User Outsider Security

The security for signcryption schemes is usually proven based on two separate notions defined by two games, one for confidentiality and one for authenticity. For multi-user outsider security, such experiments are *indistinguishability of signcryptexts under a chosen-signcryptext attack by an outsider adversary* (which is abbreviated to MOS-Conf) and *strong unforgeability of signcryptexts* (also called *integrity of signcryptexts*) *under a chosen-message attack by an outsider adversary* (which is abbreviated to MOS-Auth). We defer the specification of such security definitions to Appendix A.1. In this work we define a new and more handy all-in-one definition of multi-user outsider security in the spirit of the all-in-one security definition for authenticated encryption introduced by Rogaway and Shrimpton in [RS06], and show its equivalence to the combination of the two mentioned separate security notions in Appendix A.2.

In the experiment associated with this security notion, a sender key-pair and a receiver key-pair (corresponding to a fixed sender and a fixed receiver, respectively) are first generated. Then the goal of the adversary is to distinguish two systems: the real system $\mathbf{Real}_{\Psi}^{\mathsf{MOS}}$ where he can interact with so-called *flexible signcryption/unsigncryption oracles*, that is, he is allowed to signcrypt messages under any receiver public key and also to unsigncrypt signcryptexts under any sender public key (but where he gets $\perp$ when querying previously obtained signcryptexts, in order to avoid trivial attacks), and the ideal system $\mathbf{Ideal}_{\Psi}^{\mathsf{MOS}}$ where he can again signcrypt messages under any receiver public key with the catch that if the public key corresponds to the one of the fixed receiver, a uniformly random message (of the same length) is signcrypted instead, and he can again unsigncrypt signcryptexts under any sender public key with the catch that if the public key corresponds to the one of the fixed sender, he obtains $\perp$. We call this new notion *real-or-random*[11] *multi-user outsider security*, which is abbreviated to simply MOS security.

In the following, we use the standard notation $\mathcal{A}^{\mathbf{G}}$ to denote the random experiment of adversary $\mathcal{A}$ interacting with (the oracles of) a game $\mathbf{G}$. We succinctly write $\Pr\left[\mathcal{A}^{\mathbf{G}} = 1\right]$ to denote the probability that $\mathcal{A}$ returns the output 1 when interacting with $\mathbf{G}$. The formal specifications of the real and ideal security games are illustrated in Fig. 1.

Definition 3. *Let* $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ *be a signcryption scheme and* $\mathcal{A}$ *a probabilistic algorithm. Consider the games* $\mathbf{Real}_{\Psi}^{\mathsf{MOS}}$ *and* $\mathbf{Ideal}_{\Psi}^{\mathsf{MOS}}$ *from Fig. 1. We define the* real-or-random multi-user outsider security all-in-one advantage *of* $\mathcal{A}$ *as*

$$\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MOS}} := \Pr\left[\mathcal{A}^{\mathbf{Real}_{\Psi}^{\mathsf{MOS}}} = 1\right] - \Pr\left[\mathcal{A}^{\mathbf{Ideal}_{\Psi}^{\mathsf{MOS}}} = 1\right].$$

We say that the scheme Ψ *is* MOS secure *if* $\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MOS}}$ *is negligible for all efficient adversaries* $\mathcal{A}$.

For completeness, we refer the reader to Lemma 1 and Appendix A.2 for the details and proof of the following statement that MOS security is equivalent to MOS-Conf security[12] coupled with MOS-Auth security.

Lemma 1 (MOS $\longleftrightarrow$ MOS-Conf + MOS-Auth). *A signcryption scheme* $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ *is MOS secure if and only if it is both MOS-Conf secure and MOS-Auth secure.*

[10] From an application perspective, the two-user case is a special case of the multi-user case and therefore not explicitly considered here.

[11] Note that in this context we use the terms "ideal" and "random" interchangeably.

[12] We also define MOS-Conf using the real-or-random paradigm, which can be easily seen to be equivalent to the more frequent version using indistinguishability found in the literature.

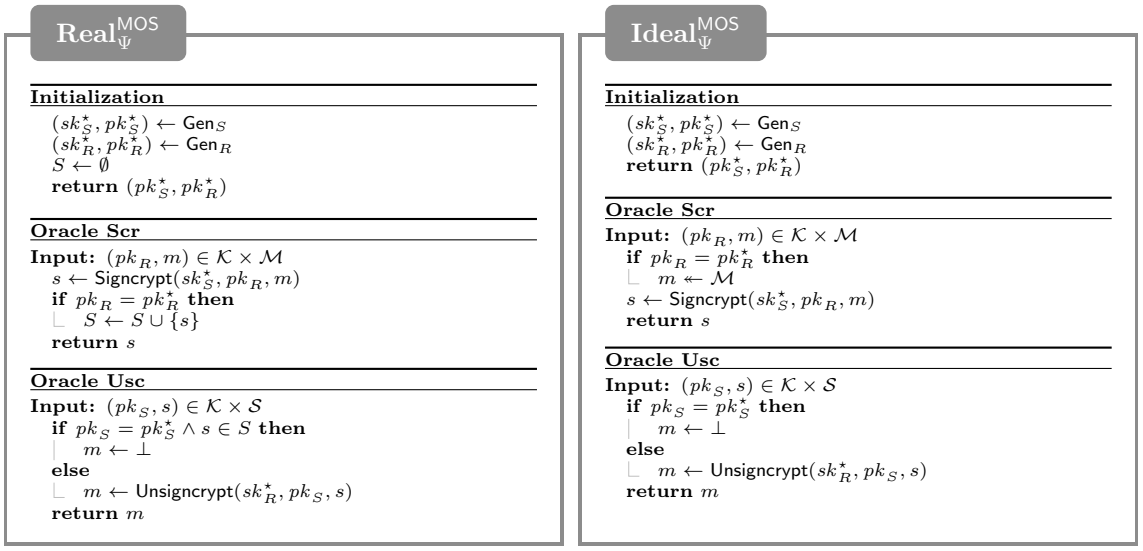

Fig. 1. Games $\mathbf{Real}_\Psi^{\mathsf{MOS}}$ and $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$ to define multi-user outsider secure signcryption schemes.

3.2 Multi-User Insider Security

Also for insider security, signcryption schemes are proven secure based on two separate notions defined by two separate experiments, one for confidentiality and one for authenticity. Such experiments are *indistinguishability of signcryptexts under a chosen-signcryptext attack by an insider adversary* (which is abbreviated to MIS-Conf) and *strong unforgeability of signcryptexts* (also called *integrity of signcryptexts*) *under a chosen-message attack by an insider adversary* (which is abbreviated to MIS-Auth).

Confidentiality. We will again employ the real-or-random paradigm as we do for outsider security, by defining the two games $\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}$ and $\mathbf{Ideal}_\Psi^{\mathsf{MIS\text{-}Conf}}$ in Fig. 2. We point out that it is easy to verify that our real-or-random formulation for confidentiality for insider security is equivalent to the one found in the literature which is based on indistinguishability of signcryptexts, by using a standard hybrid argument.

The experiment proceeds as follows. For confidentiality, where the adversary is supposed to be the sender, only a receiver key-pair (sk_R, pk_R) for the fixed receiver is generated, and the respective public key pk_R is given to the adversary. Consequently, the adversary has access to a (flexible) unsigncryption oracle,[13] but not a (flexible) signcryption oracle, as no fixed sender is defined in the experiment. Instead, we define a special signcryption oracle which allows the adversary to specify a target receiver public key (just as the regular flexible signcryption oracle does), and in addition it also requires a *valid* sender key-pair. We specify two different ways of different strength to enforce that the key-pair provided by the adversary is to be considered valid or not: on one hand, as defined usually in the literature, one considers a sender key-pair valid if it belongs to the support of the sender key-generation algorithm,[14] on the other hand, this requirement can be sharpened, thereby formalize a weaker notion, by requiring the adversary to use a specific oracle **Gen** (offered by the game itself) for generating the sender key-pairs, and requesting that only keys generated as such

[13] As usual, the unsigncryption oracle returns $\perp$ when the adversary queries previously obtained signcryptexts, in order to avoid trivial attacks.

[14] This requirement can be implemented by an (efficiently computable) membership-test for the support of Gen_S, and it is actually indispensable in order to avoid trivial attacks. For example, an attacker could specify a pair $(sk_S, 0)$ in a signcryption query, which allows him to unsigncrypt the respective result using the actual (correct) public key pk_S.

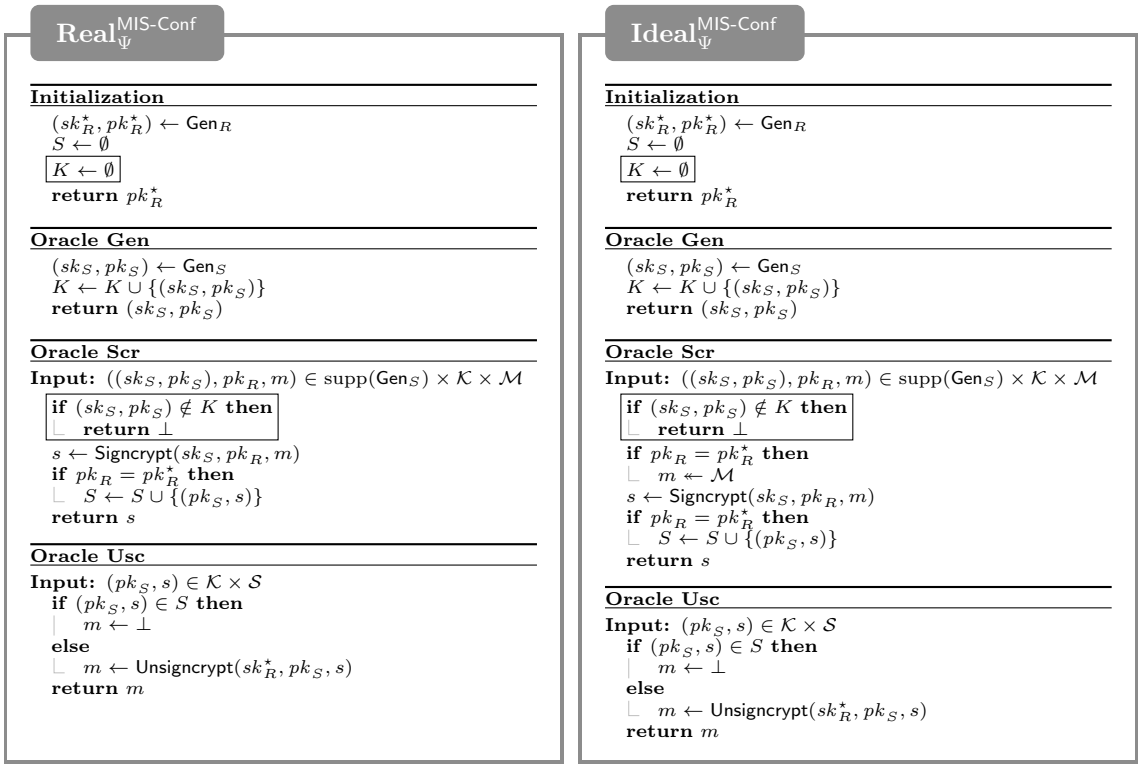

Fig. 2. Games $\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}$ and $\mathbf{Ideal}_\Psi^{\mathsf{MIS\text{-}Conf}}$ to define confidentiality for multi-user insider secure signcryption schemes. The game including the boxed statements and the oracle **Gen** constitutes the weaker version.

are provided to signcryption queries.[15] Finally, the sender secret key provided by the adversary is then used together with the fixed receiver public key to signcrypt the message. As before the only difference between $\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}$ and $\mathbf{Ideal}_\Psi^{\mathsf{MIS\text{-}Conf}}$ is that in the latter a uniformly random message is encrypted instead of the queried one in case the adversary specifies the target receiver key in the query to the oracle.

From the above two variants of MIS-Conf, we now define the advantage for the weaker version, since this is the one we use in this work. The other definition would be analogous.

Definition 4. *Let* $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ *be a signcryption scheme and* $\mathcal{A}$ *a probabilistic algorithm. We define the advantage of* $\mathcal{A}$ *in distinguishing* $\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}$ *and* $\mathbf{Ideal}_\Psi^{\mathsf{MIS\text{-}Conf}}$ *from Fig. 2 as*

$$\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MIS\text{-}Conf}} := \Pr\left[\mathcal{A}^{\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] - \Pr\left[\mathcal{A}^{\mathbf{Ideal}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right].$$

We say that the scheme Ψ *is* MIS-Conf secure *if* $\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MIS\text{-}Conf}}$ *is negligible for all efficient adversaries* $\mathcal{A}$*, where we consider the weaker game including the boxed lines (and considering the version which excludes those lines, and also the* **Gen** *oracle, would yield the definition traditionally found in the literature).*

Authenticity. In this case, the forgery game $\mathbf{Auth}_\Psi^{\mathsf{MIS}}$ appearing in Fig. 3 is considered. Here, the adversary is supposed to be the receiver, and therefore only a sender key-pair (sk_S, pk_S) for the fixed

[15] Essentially, this requirement "shifts" the choice of the randomness of the keys from the adversary to the game. This is a weaker security notion in the sense that any attack against the new game can be translated into an attack against the traditional game.

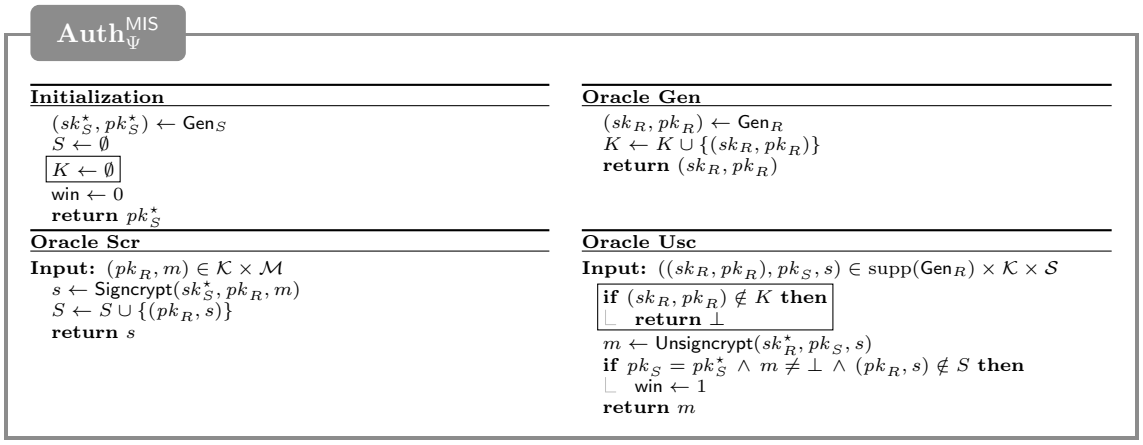

Fig. 3. Game $\mathbf{Auth}_{\Psi}^{\mathsf{MIS}}$ to define authenticity for multi-user insider secure signcryption schemes. The game including the boxed statements and the oracle **Gen** constitutes the weaker version.

sender is generated, and the respective public key pk_S is given to the adversary. Consequently, the adversary can only have access to a (flexible) signcryption oracle, but not a (flexible) unsigncryption oracle, as no fixed receiver is defined in the experiment. Instead, we define a special unsigncryption oracle which allows the adversary to specify a target sender public key (just as the regular flexible unsigncryption oracle does), and in addition it also requires a *valid* receiver key-pair. Again, the validity requirement can come in two flavours: either as defined traditionally in the literature, where one considers a receiver key-pair valid if it belongs to the support of the receiver key-generation algorithm,[16] or as before by sharpening this requirement by demanding the adversary to use a specific oracle **Gen** for generating the receiver key-pairs, and requesting that only keys generated as such are provided to unsigncryption queries. This again yields a weaker notion. The receiver secret key provided by the adversary is then used together with the fixed sender public key to unsigncrypt the signcryptext. Finally, the special unsigncryption oracle checks for every query whether the unsigncrypted message is valid (in case the provided sender public key corresponds to the one fixed in the experiment) and whether such a signcryptext was new, in which case the adversary wins this forgery game. The formal definition of (the weaker version) MIS-Auth follows.

Definition 5. *Let* $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ *be a signcryption scheme and* $\mathcal{A}$ *a probabilistic algorithm. We define the advantage of* $\mathcal{A}$ *when interacting with* $\mathbf{Auth}_{\Psi}^{\mathsf{MIS}}$ *from Fig. 3 as*

$$\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MIS\text{-}Auth}} := \Pr\left[\mathcal{A}^{\mathbf{Auth}_{\Psi}^{\mathsf{MIS}}} \text{ sets } \mathsf{win}\right].$$

We say that the scheme Ψ *is* MIS-Auth secure *if* $\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MIS\text{-}Auth}}$ *is negligible for all efficient adversaries* $\mathcal{A}$*, where we consider the weaker game including the boxed lines (and considering the version which excludes those lines, and also the* **Gen** *oracle, would yield the definition traditionally found in the literature).*

Secret Key Ignorance. In the literature, an even stronger variant for both insider confidentiality and authenticity has been proposed: the *secret key ignorant (SKI) models* [YDZ10]). The corresponding definitions of security are almost identical to Definition 4 and Definition 5, except that for confidentiality the adversary is required to give as argument to the special signcryption oracle only a sender public key instead of a valid sender key-pair, whereas for authenticity the adversary

[16] Again, this requirement can be implemented by an (efficiently computable) membership-test for the support of Gen_S, and it is again indispensable in order to avoid trivial attacks.

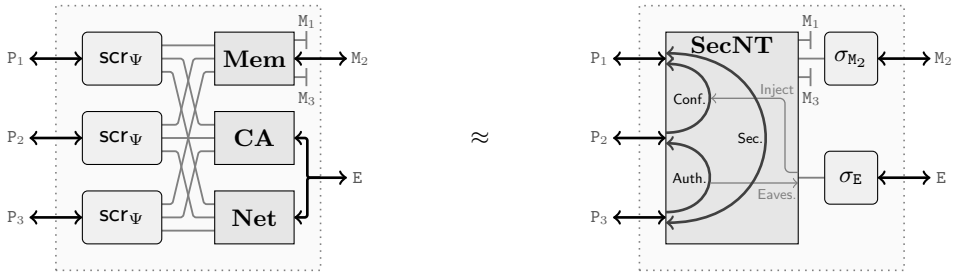

Fig. 4. An illustration of the construction notion: On the left, we have the real world for three parties with interfaces $\mathtt{P}_i$ and where the interfaces with dishonest behavior are $\mathtt{E}$ and $\mathtt{M}_2$. This models the case that the second party's key got compromised. The other keys are not compromised. In the ideal world, this translates to a secure network resource which guarantees secure (confidential *and* authentic) communication between $\mathtt{P}_1$ and $\mathtt{P}_3$, *only* confidential communication from party $\mathtt{P}_2$ to party $\mathtt{P}_1$ (but not vice-versa), and *only* authentic communication from party $\mathtt{P}_3$ to party $\mathtt{P}_2$ (but not vice-versa). Note that we do not depict the filter explicitly, but only indicate their effect, e.g., by not giving output at intruder interfaces $\mathtt{M}_1$ and $\mathtt{M}_3$.

is required to give as argument to the special unsigncryption oracle only a receiver public key instead of a valid receiver key-pair. Our analysis, however, suggests that such a strengthening is not of major practical relevance.

4 Constructive Analysis

4.1 Assumed Resources

Insecure network. We assume a network resource $\mathbf{Net}_n$ that accepts, at each interface $\mathtt{P}_i$, $i \in [n]$, a registration query that assigns an identifier to that interface. Any bitstring $\mathsf{ID} \in \{0,1\}^*$ is valid, and uniqueness is enforced (reflecting IP-addresses). Subsequently, messages can be sent at this interface in the name of that identifier, by indicating the message content m and a destination identifier. Any request is leaked at interface $\mathtt{E}$ of the network (to the network attacker). Eve can further inject any message it wants to each destination address and indicate any source address as sender. The specification as pseudo-code is found in Fig. 5.

At interface $\mathtt{E}$, these capabilities are *only potentially* available and thus not guaranteed. We thus specify a filter converter for this interface, denoted dlv, which, upon any $(\cdot, \mathsf{ID}_s, \mathsf{ID}_r)$ from interface $\mathtt{E}$ of $\mathbf{Net}_n$, it immediately outputs $(\mathtt{inject}, \cdot, \mathsf{ID}_s, \mathsf{ID}_r)$ at interface $\mathtt{E}$ of $\mathbf{Net}_n$ to reliably deliver the message and does not give any output at its outer interface and it does not react on any other input. If no attacker is present, i.e., if the filter is not removed, then the network is trivially "secure". However, if an attacker is there, it can access all the potentially available capabilities. Formally, the filter for the network is defined as $\phi^{\text{net}} := (\mathbf{1}, \ldots, \mathbf{1}, \mathsf{dlv})$ for interfaces $\mathtt{P}_1, \ldots, \mathtt{P}_n, \mathtt{E}$, where $\mathbf{1}$ is the identity converter (no changes at a party's interface).

Memory. We model the local memory of each honest party by a memory resource $\mathbf{Mem}_n$. The memory can be thought of as being composed of n local memory modules. For the ease of exposition, we summarize these modules in one memory functionality that mimics this behavior (each party can read and write to *its* (and only this) memory location). The memory allows each party to store a value. In the construction, this will be the key storage. We make the storage explicit to model key compromises. To this end, we associate an intruder interface $\mathtt{M}_i$ to each party interface $\mathtt{P}_i$. At interface $\mathtt{M}_i$, the key is *only potentially* available to an intruder Mallory and thus not guaranteed. This means that we consider a filtered memory as an assumed resource where the filter is $\phi^{\text{mem}} := (\mathbf{1}, \ldots, \mathbf{1}, \mathbf{0}, \ldots, \mathbf{0})$ for interfaces $\mathtt{P}_1, \ldots, \mathtt{P}_n, \mathtt{M}_1, \ldots, \mathtt{M}_n$, where $\mathbf{1}$ is again the identity converter, and $\mathbf{0}$ is the converter that blocks all interaction (at an intruder's interface). Therefore, key-compromise attacks (or key leakage) is captured with this filtered resource. To see this recall the construction notion of Definition 2: for every potentially dishonest interface, we consider the case when no

Resource CA_n

Initialization

$T \leftarrow$ empty table
$J \leftarrow \emptyset$

Interface $\mathtt{P}_i, i \in [n]$

Input: $(\mathtt{register}, \mathsf{ID}, val) \in \{0,1\}^* \times \{0,1\}^*$
if $\mathsf{ID} \notin J$ **then**
$J \leftarrow J \cup \{\mathsf{ID}\}$
$T[\mathsf{ID}] \leftarrow val$
output Success at $\mathtt{P}_i$
else
output Fail at $\mathtt{P}_i$

Input: $(\mathtt{fetch}, \mathsf{ID}) \in \{0,1\}^*$
if $\mathsf{ID} \in J$ **then**
output $T[\mathsf{ID}]$ at $\mathtt{P}_i$
else
output $\perp$ at $\mathtt{P}_i$

Interface $\mathtt{E}$

Input: $(\mathtt{register}, \mathsf{ID}, val) \in \{0,1\}^* \times \{0,1\}^*$
if $\mathsf{ID} \notin J$ **then**
$T[\mathsf{ID}] \leftarrow val$
$J \leftarrow J \cup \{\mathsf{ID}\}$
output Success at $\mathtt{E}$
else
output Fail at $\mathtt{E}$

Input: $\mathtt{fetchAll}$
output (J, T) at $\mathtt{E}$

Resource Net_n

Initialization

$I_1, I_2 \leftarrow$ empty tables
$J \leftarrow \emptyset$

Interface $\mathtt{P}_i, i \in [n]$

Input: $(\mathtt{register}, \mathsf{ID}) \in \{0,1\}^*$
if $\mathsf{ID} \notin J \ \wedge \ I_1[\mathtt{P}_i] = \perp$ **then**
$J \leftarrow J \cup \{\mathsf{ID}\}$
$I_1[\mathtt{P}_i] \leftarrow \mathsf{ID}$
$I_2[\mathsf{ID}] \leftarrow \mathtt{P}_i$
output Success at $\mathtt{P}_i$
else
output Fail at $\mathtt{P}_i$

Input: $(\mathtt{send}, m, \mathsf{ID}_r) \in \mathcal{M} \times \{0,1\}^*$
$\mathsf{ID}_s \leftarrow I_1[\mathtt{P}_i]$
if $\mathsf{ID}_s \neq \perp$ **then**
output $(m, \mathsf{ID}_s, \mathsf{ID}_r)$ at $\mathtt{E}$

Interface $\mathtt{E}$

Input: $(\mathtt{inject}, m, \mathsf{ID}_s, \mathsf{ID}_r) \in \mathcal{M} \times \{0,1\}^* \times \{0,1\}^*$
if $\mathsf{ID}_r \in J$ **then**
$\mathtt{P} \leftarrow I_2[\mathsf{ID}_r]$
output (m, ID_s) at $\mathtt{P}$

Resource Mem_n

Initialization

$val \leftarrow \perp$

Interface $\mathtt{P}_i, i \in [n]$

Input: $(\mathtt{store}, x) \in \{0,1\}^*$
$val \leftarrow x$

Input: $\mathtt{read}$
output val at $\mathtt{P}_i$

Interface $\mathtt{M}_i, i \in [n]$

Input: $\mathtt{reveal}$
output val at $\mathtt{M}_i$

Fig. 5. The (unfiltered) behavior of the assumed resources.

attacker is there—in which case no key is leaked because the filter is there—and the case when the attacker is present—in which case the filter is removed and the key readable by the attacker.[17] This allows to model each key compromise as a separate event.

Certificate authority. The resource $\mathbf{CA}_n$ models a key registration functionality, and we denote it by certificate authority to stick to the common term in public-key infrastructures. The resource allows to register at an interface with an identity-value pair. The resource stores this assignment and does not accept any further registration with the same identity. The certificate authority is weak in the sense that it does not perform any further test and corresponds to typical formalizations of key registration functionalities. Any party can query to $(\texttt{fetch}, \mathsf{ID})$ to retrieve the value registered for identity ID. Eve can register any value with any identity, under the constraint that the identity is not already registered. The specification as pseudo-code is found in Fig. 5. The capabilities at interface $\texttt{E}$ are again not guaranteed and will be filtered as in the case of the network.

4.2 The Protocol

Signcryption converter. The signcryption converter scr_Ψ is defined for any given signcryption scheme $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$. The converter specifies the actions that each party takes to secure the communication over the insecure network at interface $\texttt{P}_i$. Upon a registration query, a party generates the two key-pairs required by the signcryption scheme, i.e., a sender key pair and a receiver key pair that it uses to send and receive message, respectively. It then tries to register its identity at the insecure network and tries to register the identity and the two public keys with the certificate authority. If everything succeeded, the converter stores the keys to its local memory. Otherwise, the initialization is not complete and the party remains un-initialized.

Upon sending a message, an initialized party retrieves the receiver public key of its intended communication partner, and signcrypts the message according to the signcryption scheme (and retrieves the secret key from the memory) and sends the signcryptext over the network (indicating the destination address). Upon receiving a pair (s, ID) consisting of a signcryptext and a candidate source address from the insecure network, it tries to unsigncrypt the given value and outputs the resulting message. The formal specification of this protocol converter appears in Fig. 6.

The default behavior for possibly dishonest interfaces. The converters for the potentially dishonest interfaces are quite simple: the intruder is assumed to perform no additional operation (the filter is not removed and exports no capability) and this converter is therefore simply the identity converter $\mathbf{1}$. The same holds for the network attacker where no additional operation needs to be specified. Recall that attackers are hypothetical entities as discussed in Sect. 2.3.

4.3 Goal: A Secure Network with Graceful Degradation

The ideal system we want to achieve is a secure network that gracefully degrades and is specified in Fig. 7. This ideal network is basically a secure network. To see this, imagine there was no interface $\texttt{M}_i$: then parties register to the resource like to the normal network and can send and receive messages. In addition, the adversary learns the length of the message (and sender and receiver identities), and cannot inject messages. The reason for this behavior is that in the case of an honest registration query, if party $\texttt{P}_i$ registers its identity successfully, then its associated identity is only added to the special set S if there was no input $\texttt{reveal}$ at interface $\texttt{M}_i$. Now observe that the condition under which the network attacker can inject a message for some party identity ID includes that $\mathsf{ID} \notin S$. In addition, the network attacker learns only the length of the messages

[17] Looking ahead, our ideal resource will depend on which parties have been subject to an attack by an intruder and will weaken the security guarantees for this party accordingly as described later. Note that we consider static security only in this work.

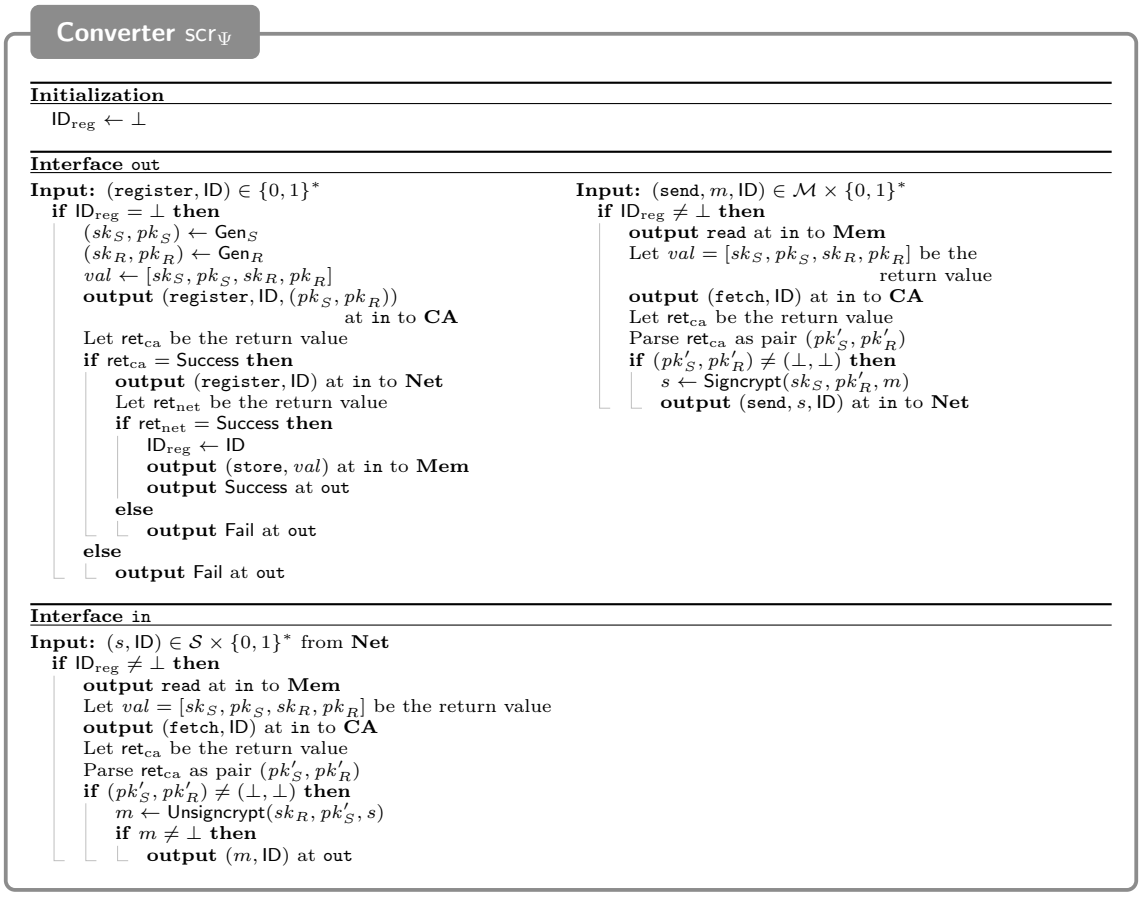

Fig. 6. The signcryption converter.

whenever a message is sent to an identity $\mathsf{ID} \in S$. Thus, since all registered identities of honest parties are in S, communication between any two of them is secure.

Now, the input `reveal` is potentially available at interface $\mathtt{M}_i$ (this models the fact that the party is compromised). Whenever this input happens, then the corresponding party identity is not included in S. This means that the network attacker at interface $\mathtt{E}$ can inject messages on behalf of the identity registered at interface $\mathtt{P}_i$ and obtains the content of any message sent to $\mathtt{P}_i$. We see that only the security of $\mathtt{P}_i$ is affected. To complete this description, note that the secure network outputs shared randomness between the intruder of party $\mathtt{P}_i$ and the network attacker. This models that in the ideal world, shared randomness is potentially available to the parties. This is indeed the case, since the network attacker learns signcryptexts that are created with the secret key leaked at interface $\mathtt{M}_i$.[18]

At interface $\mathtt{M}_i$, the capability to reveal is *only potentially* available to an intruder Mallory and thus not guaranteed. This means that we actually consider the filtered resource $\mathbf{SecNT}_{n\,\phi^{\text{ideal}}}$ with the filter $\phi^{\text{ideal}} := (\mathbf{1}, \ldots, \mathbf{1}, \mathbf{0}, \ldots, \mathbf{0}, \mathsf{dlv})$ for interfaces $\mathtt{P}_1, \ldots, \mathtt{P}_n, \mathtt{M}_1, \ldots, \mathtt{M}_n, \mathtt{E}$, where converters $\mathbf{1}$, $\mathbf{0}$, and dlv are as above. Looking ahead, the potentially available capability to compromise a party corresponds to the potentially available input `reveal` in the ideal world.

[18] There is a more technical argument why shared randomness is leaked: the simulator for the intruder has to simulate a correct key pair, and the simulator for the network attacker has to compute the same key pair to simulate the correct distribution of signcryptexts. Otherwise, the simulation is not consistent. Looking ahead, both simulators will run the key generation algorithms using the provided randomness of sufficient length.

Resource SecNT_n

Initialization

$r_1||r_2||\ldots||r_n \leftarrow (\{0,1\}^{2\kappa})^n$ ▷ Common randomness
$I_1, I_2 \leftarrow$ empty tables ▷ Mapping interfaces and identities
$J, S \leftarrow \emptyset$ ▷ All registered, and honest identities
$L \leftarrow [\,]$ ▷ Inputs of parties
for $i = 1$ to n **do**
  $\mathsf{setupCompleted}_i \leftarrow \mathsf{false}$ ▷ Init. flag

Interface $\mathtt{M}_i$

Input: `reveal`
  $I_c \leftarrow I_c \cup \{i\}$
  if $\mathsf{setupCompleted}_i$ **then**
    output r_i at $\mathtt{M}_i$

Interface `E`

Input: $(\mathtt{deliver}, j) \in \mathbb{N}$
  if $j \leq |L|$ **then**
    Parse $L[j]$ as $(m, \mathsf{ID}_s, \mathsf{ID}_r)$
    $\mathtt{P} \leftarrow I_2[\mathsf{ID}_r]$
    output (m, ID_s) at $\mathtt{P}$

Input: $(\mathtt{register}, \mathsf{ID}) \in \{0,1\}^*$
  if $\mathsf{ID} \notin J$ **then**
    $J \leftarrow J \cup \{\mathsf{ID}\}$
    output Success at `E`
  else
    output Fail at `E`

Input: $(\mathtt{inject}, m, \mathsf{ID}_s, \mathsf{ID}_r) \in \mathcal{M} \times \{0,1\}^* \times \{0,1\}^*$
  $\mathtt{P} \leftarrow I_2[\mathsf{ID}_r]$
  if $\mathsf{ID}_s \in J$ **and** $\mathtt{P} \neq \perp$ **and** $\mathsf{ID}_s \notin S$ **then**
    output (m, ID_s) at $\mathtt{P}$

Input: `getCommonRand`
  output $\{(i, r_i) \mid i \in I_c\}$ at `E`

Input: `getMapping`
  output (I_1, I_2, J) at `E`

Interface $\mathtt{P}_i$

Input: $(\mathtt{register}, \mathsf{ID}) \in \{0,1\}^*$
  if $\mathsf{ID} \notin J \ \wedge\ I_1[\mathtt{P}_i] = \perp$ **then**
    $J \leftarrow J \cup \{\mathsf{ID}\}$
    $I_1[\mathtt{P}_i] \leftarrow \mathsf{ID}$
    $I_2[\mathsf{ID}] \leftarrow \mathtt{P}_i$
    $\mathsf{setupCompleted}_i \leftarrow \mathsf{true}$
    if $i \notin I_c$ **then**
      $S \leftarrow S \cup \{\mathsf{ID}\}$
    output Success at $\mathtt{P}_i$
  else
    output Fail at $\mathtt{P}_i$

Input: $(\mathtt{send}, m, \mathsf{ID}) \in \mathcal{M} \times \{0,1\}^*$
  $\mathsf{ID}_s \leftarrow I_1[\mathtt{P}_i]$
  if $\mathsf{ID}_s \neq \perp$ **then**
    if $\mathsf{ID} \in J$ **then**
      if $\mathsf{ID} \in S$ **then**
        $L \leftarrow L||(m, \mathsf{ID}_s, \mathsf{ID})$
        output $(|m|, \mathsf{ID}_s, \mathsf{ID}_r)$ at `E`
      else
        output $(m, \mathsf{ID}_s, \mathsf{ID}_r)$ at `E`

Fig. 7. The (unfiltered) behavior of the constructed resource.

4.4 Formal Statement

We are now ready to formally state the main theorem of this work. Recall that we assign to every honest (party) interface the signcryption converter scr_Ψ, whereas to the possibly dishonest network attacker interface `E` and to the potentially dishonest intruder interfaces $\mathtt{M}_i$, we assign the identity converter (they model hypothetical entities). This can be summarized by the vector $\pi^\Psi = (\mathsf{scr}_\Psi, \ldots, \mathsf{scr}_\Psi, \mathbf{1}, \ldots, \mathbf{1}, \mathbf{1})$. The real system is the parallel composition of the assumed resources $[\mathbf{Net}_n, \mathbf{CA}_n, \mathbf{Mem}_n]_{\phi^{\text{real}}}$, where ϕ^{real} is the filter that shields the memory (interfaces $\mathtt{M}_i$), the network, and the certificate authority (interface `E`), as described above and thus is equal to the filter in the ideal world.[19] We prove the following result:

Theorem 1. *Let Ψ be a signcryption scheme, let $n > 0$ be an integer, and let κ be an upper bound on the randomness used in one invocation of the key-generation algorithm. Then, the associated protocol $\pi^\Psi := (\mathsf{scr}_\Psi, \ldots, \mathsf{scr}_\Psi, \mathbf{1}, \ldots, \mathbf{1}, \mathbf{1})$ constructs the gracefully-degrading secure network from an insecure network, a certificate authority, and a memory resource within $\varepsilon(\cdot)$ and with respect to potentially dishonest $\mathcal{U} := \{\mathtt{M}_1, \ldots, \mathtt{M}_n, \mathtt{E}\}$, i.e.,*

$$[\mathbf{Net}_n, \mathbf{CA}_n, \mathbf{Mem}_n]_{\phi^{\text{real}}} \overset{(\pi^\Psi, \varepsilon, \mathcal{U})}{\Longrightarrow} \mathbf{SecNT}_{n\,\phi^{\text{ideal}}},$$

[19] Formally, this means that for all interfaces $\mathtt{P}_i$, the filter ϕ^{real} applies the identity converter, for all interfaces $\mathtt{M}_i$ it applies $\mathbf{0}$, and for interface `E` it applies the filter converter dlv. The filter in the real world and in the ideal world are thus the same.

for $\varepsilon(\mathbf{D}) := n^2 \cdot \mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\rho_1(\mathbf{D})} + n \cdot \mathsf{Adv}^{\mathsf{MIS\text{-}Auth}}_{\Psi,\rho_2(\mathbf{D})} + n \cdot \mathsf{Adv}^{\mathsf{MIS\text{-}Conf}}_{\Psi,\rho_3(\mathbf{D})}$*, where the (efficient) black-box reductions* ρ_1*,* ρ_2*, and* ρ_3 *are defined below in the proofs for Lemma 2, Lemma 3, and Lemma 4, respectively.*

Stated differently, if the signcryption scheme is secure in the respective multi-user, outsider-security and insider-security model, then the construction is achieved.

5 Proof of Theorem 1

To prove Theorem 1, we specify the two converters σ_{mem} and σ_{net} in Fig. 8 and prove that Definition 2 is fulfilled for $\sigma = (\sigma_{\mathtt{M}_1}, \ldots, \sigma_{\mathtt{M}_n}, \sigma_{\mathtt{E}})$, where $\sigma_{\mathtt{E}} := \sigma_{\text{net}}$ and $\sigma_{\mathtt{M}_i} := \sigma_{\text{mem}}$ and for the above choice of $\varepsilon(.)$. In particular, we show that for any subset $\mathcal{C} \subseteq \mathcal{U}$ we have that

$$\Delta^{\mathbf{D}}(\pi^{\Psi}_{\overline{\mathcal{C}}}\, \phi^{\text{real}}_{\overline{\mathcal{C}}}[\mathbf{Net}_n, \mathbf{CA}_n, \mathbf{Mem}_n],\, \sigma_{\mathcal{C}}\, \phi^{\text{ideal}}_{\overline{\mathcal{C}}}\mathbf{SecNT}_n) \leq \varepsilon(\mathbf{D}), \tag{1}$$

for any distinguisher $\mathbf{D}$. Fix any set $\mathcal{C} \subseteq \{\mathtt{M}_i, \ldots, \mathtt{M}_n, \mathtt{E}\}$. We first observe that if $\mathtt{E} \notin \mathcal{C}$, then the real and the ideal world are indistinguishable: both systems behave like a secure network, and for any interface $\mathtt{M}_i \in \mathcal{C}$, two signcryption key pairs are leaked. Since the network attacker is not present, this has no observable effect to the security properties. We can thus, without loss of generality assume that $\mathtt{E} \in \mathcal{C}$. The set $\mathcal{C}$ induces a special corruption set $\mathcal{Z} \subseteq \{\mathtt{M}_1, \ldots, \mathtt{M}_n\}$, i.e., $\mathtt{M}_i \in \mathcal{Z} \leftrightarrow \mathtt{M}_i \in \mathcal{C}$. The set $\mathcal{Z}$ intuitively describes the corruption set, i.e., the set of parties whose keys are stolen. We prove the statement by a game-hopping argument. We start with the real world $\mathbf{H}^{\mathcal{Z}}_0$ which is equivalent to the real world $\pi^{\Psi}_{\overline{\mathcal{C}}}\, \phi^{\text{real}}_{\overline{\mathcal{C}}}[\mathbf{Net}_n, \mathbf{CA}_n, \mathbf{Mem}_n]$ with $\mathcal{C} := \{\mathtt{E}\} \cup \mathcal{Z}$, and end with system $\mathbf{H}^{\mathcal{Z}}_6$ which is equivalent to ideal world $\sigma_{\mathcal{C}}\, \phi^{\text{ideal}}_{\overline{\mathcal{C}}}\mathbf{SecNT}_n$ with with $\mathcal{C} := \{\mathtt{E}\} \cup \mathcal{Z}$. Each hop in this sequence is justified by careful syntactic inspection of the differences of the systems and their difference is either 0 (in case of syntactic modifications that do not affect the behavior) or can be bounded by the respective advantage of an attacker against the security games of signcryption by means of a reduction. Note that we prove the statement for a insider-security notion that is implied by the traditional insider-security notion. By transitivity, Theorem 1 also holds with respect to the traditional notion.

5.1 Sequence of Hybrid Worlds

Let $\mathcal{Z}$ be an arbitrary and fixed corruption set. We describe the systems below informally, indicating what is the change from one hybrid to the next. The code is given in the supplementary material Appendix C including graphical indication in the pseudo-code what changes from one hybrid to the next.

The first hybrid system $\mathbf{H}^{\mathcal{Z}}_0$, depicted in Fig. 15 in Appendix C, describes the behavior of the real world, where we plugged together the protocol converters and the assumed system. We further introduce some new variables such as bad_1 and bad_2 that do not have an impact on the behavior.

The second hybrid system $\mathbf{H}^{\mathcal{Z}}_1$, first depicted in Fig. 16 in Appendix C, is a slight modification of the first, where we replace encryptions of messages by encryptions of random messages, in case the communication is between two honest parties whose keys were not stolen. Furthermore, the adversary cannot inject messages for any such pairs of parties. The variable bad_1 is true, if and only if the adversary succeeds in breaking the security between any such pair. Note that the set S computed by the hybrids comprises all parties which are not corrupted, that is, all parties not in $\mathcal{Z}$.

For the difference between the first and second hybrid, we can prove the following lemma.

Lemma 2. *If* Ψ *is MOS secure, then* $\mathbf{H}^{\mathcal{Z}}_0 \approx \mathbf{H}^{\mathcal{Z}}_1$ *(that is, systems* $\mathbf{H}^{\mathcal{Z}}_0$ *and* $\mathbf{H}^{\mathcal{Z}}_1$ *are computationally indistinguishable). More precisely, for any distinguisher* $\mathbf{D}$ *we have* $\Delta^{\mathbf{D}}(\mathbf{H}^{\mathcal{Z}}_0, \mathbf{H}^{\mathcal{Z}}_1) \leq n^2 \cdot \mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\rho_{\mathcal{Z}}(\mathbf{D})}$*, that is, a (successful) distinguisher* $\mathbf{D}$ *for* $\mathbf{H}^{\mathcal{Z}}_0$ *and* $\mathbf{H}^{\mathcal{Z}}_1$ *can be transformed into a (successful) distinguisher* $\mathcal{A} = \rho_{\mathcal{Z}}(\mathbf{D})$ *(defined in the proof) for* $\mathbf{Real}^{\mathsf{MOS}}_{\Psi}$ *and* $\mathbf{Ideal}^{\mathsf{MOS}}_{\Psi}$*.*

Converter σ_{net}

Initialization

$T \leftarrow$ empty table; $val_{\mathtt{P}_k} \leftarrow \perp$ for all $k \in [n]$; $j \leftarrow 0$

Interface `out`

Input: $(\mathtt{inject}, s, \mathsf{ID}_s, \mathsf{ID}_r)$ (for **Net**)
UpdateKeyTable ▷ $s \in \mathcal{S}$
Let $val_{\mathtt{P}_i} = [sk_S, pk_S, sk_R, pk_R]$
Parse $T[\mathsf{ID}_s]$ as (pk'_S, pk'_R)
if $(pk'_S, pk'_R) \neq (\perp, \perp)$ **then**
  if $\exists j : L_{sim}[c, \mathsf{ID}_s, \mathsf{ID}_r] = j\}$ **then**
    $j \leftarrow L_{sim}[c, \mathsf{ID}_s, \mathsf{ID}_r]$
    output $(\mathtt{deliver}, j)$ to **SecNT**
  else
    $m \leftarrow \mathsf{Unsigncrypt}(sk_R, pk'_S, s)$
    if $m \neq \perp$ **then**
      $(\mathtt{inject}, m, \mathsf{ID}_s, \mathsf{ID}_r)$

Input: $(\mathtt{register}, \mathsf{ID}, val)$ (for **CA**)
output $(\mathtt{register}, \mathsf{ID})$ at **SecNT**
Let v be the returned answer
if $v = \mathsf{Success}$ **then**
  $T[\mathsf{ID}] \leftarrow val$
  output Success at `E`
else
  output Fail at `E`

Input: `fetchAll` (for **CA**)
UpdateKeyTable
output `getMapping` at `in` to **SecNT**
Let (I_1, I_2, J) be the returned value
output (J, T) at `out`

Interface `in`

Input: $(v, \mathsf{ID}_s, \mathsf{ID}_r)$ (from **Net**)
UpdateKeyTable ▷ $v \in \mathcal{M}$ or $v \in \mathbb{N}$
output `getMapping` at `in` to **SecNT**
Let (I_1, I_2, J) be the returned value
Let $\mathtt{P}_i \leftarrow I_2[\mathsf{ID}_s]$
Parse val_i as $[sk_S, pk_S, sk_R, pk_R]$
Parse $T[\mathsf{ID}]$ as (pk'_S, pk'_R)
if $(pk'_S, pk'_R) \neq (\perp, \perp)$ **then**
  if $v \in \mathbb{N}$ **then**
    Let $\ell \leftarrow v$
    $m^* \twoheadleftarrow \{0,1\}^\ell$
    $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m^*)$
  else
    $m^* \leftarrow v$
    $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m^*)$
  $L_{sim}[c, \mathsf{ID}_s, \mathsf{ID}_r] \leftarrow j$
  $j \leftarrow j + 1$
  output $(s, \mathsf{ID}_s, \mathsf{ID}_r)$ at `E`

Helper procedure of this simulator:
procedure UpdateKeyTable
  output `getMapping` at `in` to **SecNT**
  Let (I_1, I_2, J) be the returned value
  output `getCommonRand` to **SecNT**
  Let $Z := \{(i, r_i)\}$ be the return value
  for each $\mathsf{ID} \in J$ **do**
    if $I_2[\mathsf{ID}] \neq \perp \ \wedge\ T[\mathsf{ID}] = \perp$ **then**
      Let $\mathtt{P}_k \leftarrow I_2[\mathsf{ID}]$
      if $(k, \cdot) \notin Z$ **then** $(r_k^1, r_k^2) \twoheadleftarrow \{0,1\}^{2\kappa}$
      $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_k^1)$
      $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_k^2)$
      $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
      $val_{\mathtt{P}_k} \leftarrow [sk_S, pk_S, sk_R, pk_R]$

Converter σ_{mem}

Initialization

$val \leftarrow \perp$
output `reveal` at `in` to **SecNT** ▷ Compromise this party in the ideal world.

Interface `out`

Input: `reveal`
if $val = \perp$ **then**
  output `reveal` at `in` to **SecNT**
  Let r be the returned value
  if r equals $r^1 || r^2 \in \{0,1\}^{2\kappa}$ **then**
    $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r^1)$
    $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r^2)$
    $val \leftarrow [sk_S, pk_S, sk_R, pk_R]$
output val at `out`

Fig. 8. The simulator converters for the construction.

Proof (sketch). The complete formal proof of Lemma 2 is deferred to Appendix B.1. The idea is to use a standard hybrid argument on the set of all pairs of users which (pairwise) do not leak their secret key in the real world, that is, the set of all pairs of users which in the ideal world can communicate in a secure (both confidential and authentic) fashion. More precisely, we select one of those pairs of users uniformly at random, and for their (mono-directional) communications and design a reduction $\rho_{\mathcal{Z}}(\mathbf{D})$ that uses the oracles provided to the adversary by the security experiment to emulate the correct view towards $\mathbf{D}$. □

For the theorem statement, we will set $\rho_1(\mathbf{D}) \coloneqq \arg\max_{\rho_{\mathcal{Z}}(\mathbf{D}):\mathcal{Z}\subseteq\mathcal{U}\setminus\{\mathrm{E}\}}\{\mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\rho_{\mathcal{Z}}(\mathbf{D})}\}$, where we assume that the return value is a single adversary (note that there will always be a maximum in this finite set of possibilities and in case of multiple maxima, we simple apply a tie-breaking rule like lexicographic ordering).

The third hybrid system $\mathbf{H}^{\mathcal{Z}}_2$, first depicted in Fig. 17 in Appendix C, is a slight modification of the second: in this hybrid system, a message cannot be injected for pairs of parties, where the source of a message is a honest party whose key was not stolen (and the recipient is a party about whom we do not make an assumption). The variable bad_2 is true, if and only if the adversary succeeds in breaking the security between any such pair.

For the difference between the second and third hybrid, we can prove the following lemma.

Lemma 3. *If Ψ is MIS-Auth secure, then $\mathbf{H}^{\mathcal{Z}}_1 \approx \mathbf{H}^{\mathcal{Z}}_2$ (that is, systems $\mathbf{H}^{\mathcal{Z}}_1$ and $\mathbf{H}^{\mathcal{Z}}_2$ are computationally indistinguishable). More precisely, for any distinguisher $\mathbf{D}$ we have $\Delta^{\mathbf{D}}(\mathbf{H}^{\mathcal{Z}}_1, \mathbf{H}^{\mathcal{Z}}_2) \leq n \cdot \mathsf{Adv}^{\mathsf{MIS\text{-}Auth}}_{\Psi,\rho_{\mathcal{Z}}(\mathbf{D})}$, that is, a (successful) distinguisher $\mathbf{D}$ for $\mathbf{H}^{\mathcal{Z}}_1$ and $\mathbf{H}^{\mathcal{Z}}_2$ can be transformed into a (successful) forger $\mathcal{A} = \rho_{\mathcal{Z}}(\mathbf{D})$ (defined in the proof) for $\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}$.*

Proof (sketch). The complete formal proof of Lemma 3 is deferred to Appendix B.2. The idea is again to use a standard hybrid argument, but this time on the set of all senders which did not leak their secret key in the real world, that is, the set of all users which in the ideal world can send messages in an authentic fashion to other users. More precisely, we select one of those users uniformly at random, and for his (outgoing) communications we use the oracles provided to the adversary by the security experiment (again, the reduction will depend on the corruption set $\mathcal{Z}$). Note that for the very special case that $\mathcal{Z} = \emptyset$ the statement is trivial since there is no difference between $\mathbf{H}^{\mathcal{Z}}_1$ and $\mathbf{H}^{\mathcal{Z}}_2$ by definition. □

Finally, the reduction $\rho_2(\cdot)$ stated in the theorem is defined analogous to the reduction $\rho_1(\cdot)$ above.

The fourth hybrid system $\mathbf{H}^{\mathcal{Z}}_3$, first depicted in Fig. 18 in Appendix C, is a slight modification of the third, where we replace encryptions of messages by encryptions of random messages, in case the recipient of a message is an honest party whose key was not stolen (and the source is a party about whom we make no assumption).

For the difference between the third and the fourth hybrid, we can prove the following lemma.

Lemma 4. *If Ψ is MIS-Conf secure, then $\mathbf{H}^{\mathcal{Z}}_2 \approx \mathbf{H}^{\mathcal{Z}}_3$ (that is, systems $\mathbf{H}^{\mathcal{Z}}_2$ and $\mathbf{H}^{\mathcal{Z}}_3$ are computationally indistinguishable). More precisely, for any distinguisher $\mathbf{D}$ we have $\Delta^{\mathbf{D}}(\mathbf{H}^{\mathcal{Z}}_2, \mathbf{H}^{\mathcal{Z}}_3) \leq n \cdot \mathsf{Adv}^{\mathsf{MIS\text{-}Conf}}_{\Psi,\rho_{\mathcal{Z}}(\mathbf{D})}$, that is, a (successful) distinguisher $\mathbf{D}$ for $\mathbf{H}^{\mathcal{Z}}_2$ and $\mathbf{H}^{\mathcal{Z}}_3$ can be transformed into a (successful) distinguisher $\mathcal{A} = \rho_{\mathcal{Z}}(\mathbf{D})$ (defined in the proof) for $\mathbf{Real}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$ and $\mathbf{Ideal}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$.*

Proof (sketch). The complete formal proof of Lemma 4 is deferred to Appendix B.3. The idea is again to use a standard hybrid argument, but this time on the set of all receivers which did not leak their secret key in the real world, that is, the set of all users which in the ideal world can receive messages in a confidential fashion. More precisely, we select one of those users uniformly at random, and for his (ingoing) communications we use the oracles provided to the adversary by the security experiment (again, the reduction will depend on the corruption set $\mathcal{Z}$). Note that for the very special case that $\mathcal{Z} = \emptyset$ the statement is trivial since there is no difference between $\mathbf{H}^{\mathcal{Z}}_2$ and $\mathbf{H}^{\mathcal{Z}}_3$ by definition. □

Finally, the reduction $\rho_3(\cdot)$ stated in the theorem is defined analogous to the reduction $\rho_1(\cdot)$ above.

The fifth hybrid system $\mathbf{H}_4^{\mathcal{Z}}$, first depicted in Fig. 19 in Appendix C, is a syntactic modification of the fourth. In particular, we observe that the mapping of identities to interfaces is stored redundantly, once within the network and once within the certification authority. Hence, it is sufficient to store it only once. We further simplify the case distinction upon input $(\texttt{send}, m, \mathsf{ID})$ at an interface $\texttt{P}_i$, and upon input $(\texttt{inject}, s, \mathsf{ID}_s, \mathsf{ID}_r)$ at interface $\texttt{E}$. The behavior of the resulting system is not affected by any of these changes and the two hybrids are equivalent, i.e., $\mathbf{H}_3^{\mathcal{Z}} = \mathbf{H}_4^{\mathcal{Z}}$.

The sixth hybrid system $\mathbf{H}_5^{\mathcal{Z}}$, first depicted in Fig. 20 in Appendix C, is a syntactic modification of the previous one. In this system, we observe that also the identities are stored redundantly, so it is sufficient to only store the identities in one set (which is J_{ca} in this case). Furthermore, we can test various conditions at once and do not need nested if-statements upon input $(\texttt{register}, \mathsf{ID})$ at an interface $\texttt{P}_i$. The modifications we make in this step do not affect the behavior. Their sole purpose is to bring this system closer to the ideal world system. By inspecting the pseudo-code in Fig. 20 we conclude $\mathbf{H}_4^{\mathcal{Z}} = \mathbf{H}_5^{\mathcal{Z}}$.

Finally, the seventh hybrid system $\mathbf{H}_6^{\mathcal{Z}}$, first depicted in Fig. 21 in Appendix C, contains the final, syntactic modifications and equals the ideal world with simulators attached at the corresponding interfaces.

These final modifications include the following items: first, instead of generating the key pairs upon registration, we simply generate them when needed. For this, the system first generates some sufficiently long shared randomness. Whenever a key is generated for an honest party whose key is stolen, the corresponding part of the shared randomness is used to generate it (the simulator will learn the shared randomness for those people). If a key is to be generated for a party whose key is not stolen, the randomness is sampled uniformly at random to generate the keys (this will eventually be done within the simulator). Finally, we replace the the list L that stores messages, source and destination identities, and the ciphertext by two lists and implement an equivalent lookup using these two lists. Finally, we also rename the set J_{ca} to J and conclude that $\mathbf{H}_5^{\mathcal{Z}} = \mathbf{H}_6^{\mathcal{Z}}$.

We depict the final hybrid system $\mathbf{H}_6^{\mathcal{Z}}$ a second time in Fig. 22 in Appendix C. For better accessibility, we color the corresponding parts of the simulators in blue and surround it by a solid line, and we color the code executed by the constructed resource $\mathbf{SecNT}_n$ green and surround it by a dashed line. The last hybrid system is thus the compilation of the simulator and the constructed resource–and thus equals the ideal system—which can be concluded by inspection.

This concludes the game-hopping argument and the proof of Theorem 1.

A special case. An interesting corollary that we can directly observe by looking at the game-hopping argument is that in the special case when the set of interfaces with potential dishonest behavior is the set $\{\texttt{E}\}$, we get the following statement: The outsider security model implies the construction of a secure network if no honest parties' keys are compromised.

Corollary 1. *If there are no key compromises, i.e,* $\mathcal{U} = \{\texttt{E}\}$*, then*

$$[\mathbf{Net}_n, \mathbf{CA}_n, \mathbf{Mem}_n]_{\phi^{\text{real}}} \overset{(\pi^{\Psi}, \varepsilon, \{\texttt{E}\})}{\longmapsto} \mathbf{SecNT}_{n\,\phi^{\text{ideal}}},$$

for $\varepsilon(\mathbf{D}) := n^2 \cdot \mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\rho_1(\mathbf{D})}$*, where the (efficient) black-box reduction* ρ_1 *is defined in the proof of Lemma 2.*

Proof. The proof follows by inspecting the above arguments for $\mathcal{Z} := \emptyset$. □

6 Conclusions

In this work, we have taken a novel look at the basic notions of signcryption security. We have identified and formalized an important application of signcryption schemes as network protocols in a typical PKI setup. We observed which game-based security notions are adequate to conclude the security of this construction. This is important as it serves at least two purposes: (1) it helps to understand the importance of insider security and strongly supports considering it as the standard notion, and (2) it helps to identify which variant of insider security is the preferred one by providing evidence that the one discussed in this work is adequate to achieve a meaningful construction. The methodology that we put forward in our work can be the basis for future reviews on different notions for signcryption security in order to investigate what they exactly achieve in a given application scenario, i.e., based on a set of assumed resources modeling the real-world. It is an interesting future research direction to analyze alternative definitions, including the recently introduced enhanced definitions, by conducting an application-centric analysis in our spirit.

References

ADR02. Jee An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In *Advances in Cryptology—EUROCRYPT 2002*, pages 83–107. Springer, 2002.

An01. Jee Hea An. Authenticated encryption in the public-key setting: Security notions and analyses. Cryptology ePrint Archive, Report 2001/079, 2001. http://eprint.iacr.org/2001/079.

BD06. Tor E Bjørstad and Alexander W Dent. Building better signcryption schemes with tag-kems. In *Public Key Cryptography - PKC 2006*, pages 491–507. Springer, 2006.

BF08. Manuel Barbosa and Pooya Farshim. Certificateless signcryption. In *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, pages 369–372. ACM, 2008.

BN00. Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *Advances in Cryptology—ASIACRYPT 2000*, pages 531–545. Springer, 2000.

Boy03. Xavier Boyen. Multipurpose identity-based signcryption. In *Advances in Cryptology - CRYPTO 2003*, pages 383–399. Springer, 2003.

BR06. Mihir Bellare and Phillip Rogaway. Code-based game-playing proofs and the security of triple encryption. In *Advances in Cryptology–EUROCRYPT 2006*, pages 409–426, 2006.

BSZ07. Joonsang Baek, Ron Steinfeld, and Yuliang Zheng. Formal proofs for the security of signcryption. *Journal of cryptology*, 20(2):203–235, 2007.

Can01. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42nd Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

ÇGP+13. Çağatay Çapar, Dennis Goeckel, Kenneth G Paterson, Elizabeth A Quaglia, Don Towsley, and Murtaza Zafer. Signal-flow-based analysis of wireless security protocols. *Information and Computation*, 226:37–56, 2013.

CHK03. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology — EUROCRYPT 2003*, pages 255–271. Springer, 2003.

CHK05. Ran Canetti, Shai Halevi, and Jonathan Katz. Adaptively-secure, non-interactive public-key encryption. In *Theory of Cryptography - TCC 2005*, pages 150–168. Springer, 2005.

DDM15a. Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Compact attribute-based encryption and signcryption for general circuits from multilinear maps. In *Progress in Cryptology – INDOCRYPT 2015*, pages 3–24. Springer, 2015.

DDM15b. Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Functional signcryption: Notion, construction, and applications. In *Provable Security – ProvSec 2015*, pages 268–288. Springer, 2015.

Den05. Alexander W Dent. Hybrid signcryption schemes with insider security. In *Information Security and Privacy: 10th Australasian Conference, ACISP 2005*, pages 253–266. Springer, 2005.

DLK+14. Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, and J. Alex Halderman. The matter of heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 475–488. ACM, 2014.

FHH14. Eduarda SV Freire, Julia Hesse, and Dennis Hofheinz. Universally composable non-interactive key exchange. In *Security and Cryptography for Networks – SCN 2014*, pages 1–20. Springer, 2014.

FHKP13. Eduarda SV Freire, Dennis Hofheinz, Eike Kiltz, and Kenneth G Paterson. Non-interactive key exchange. In *Public-Key Cryptography – PKC 2013*, pages 254–271. Springer, 2013.

GK07. Kristian Gjosteen and Lillian Krakmo. Universally composable signcryption. In *EuroPKI 2007*, pages 346–353. Springer, 2007.

GM18. François Gérard and Keno Merckx. Post-quantum signcryption from lattice-based signatures. Cryptology ePrint Archive, Report 2018/056, 2018.

HKR15. Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption aez and the problem that it solves. In *Advances in Cryptology – EUROCRYPT 2015, Part I*, pages 15–44. Springer, 2015.

HMM15. Dennis Hofheinz, Christian Matt, and Ueli Maurer. Idealizing identity-based encryption. In *Advances in Cryptology – ASIACRYPT 2015, Part I*, pages 495–520. Springer, 2015.

LBZ10. Joseph K Liu, Joonsang Baek, and Jianying Zhou. Online/offline identity-based signcryption revisited. In *Inscrypt 2010*, pages 36–51. Springer, 2010.

LQ03. Benoit Libert and Jean-Jacques Quisquater. A new identity based signcryption scheme from pairings. In *Information Theory Workshop, 2003. Proceedings.*, pages 155–158. IEEE, 2003.

LQ04. Benoît Libert and Jean-Jacques Quisquater. Efficient signcryption with key privacy from gap diffie-hellman groups. In *Public Key Cryptography – PKC 2004*, pages 187–200. Springer, 2004.

Mau02. Ueli Maurer. Indistinguishability of random systems. In *Advances in Cryptology — EUROCRYPT 2002*, pages 110–132. Springer, 2002.

Mau11. Ueli Maurer. Constructive cryptography - a new paradigm for security definitions and proofs. In *TOSCA 2011*, pages 33–56, 2011.

ML02. John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. https://eprint.iacr.org/2002/098.

MR11. Ueli Maurer and Renato Renner. Abstract cryptography. In *Innovations in Theoretical Computer Science*, pages 1–21. Tsinghua University Press, 2011.

Nie02. Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *Advances in Cryptology — CRYPTO 2002*. Springer, 2002.

NSS+17. Matus Nemec, Marek Sys, Petr Svenda, Dusan Klinec, and Vashek Matyas. The return of coppersmith's attack: Practical factorization of widely used rsa moduli. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1631–1648. ACM, 2017.

PPB14. Tapas Pandit, Sumit Kumar Pandey, and Rana Barua. Attribute-based signcryption: Signer privacy, strong unforgeability and ind-cca2 security in adaptive-predicates attack. In *Provable Security – ProvSec 2014*, pages 274–290. Springer, 2014.

RS06. Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In *Advances in Cryptology – EUROCRYPT 2006*, pages 373–390. Springer, 2006.

SVR10. S Sharmila Deva Selvi, S Sree Vivek, and C Pandu Rangan. Identity based public verifiable signcryption scheme. In *Provable Security – ProvSec 2010*, pages 244–260. Springer, 2010.

SVVR12. S Sharmila Deva Selvi, S Sree Vivek, Dhinakaran Vinayagamurthy, and C Pandu Rangan. Id based signcryption scheme in standard model. In *Provable Security – ProvSec 2012*, pages 35–52. Springer, 2012.

SZ00. Ron Steinfeld and Yuliang Zheng. A signcryption scheme based on integer factorization. *Information Security: Third International Workshop, ISW 2000*, pages 131–146, 2000.

TP14. Youliang Tian and Changgen Peng. Universally composable secure group communication. Cryptology ePrint Archive, Report 2014/647, 2014. https://eprint.iacr.org/2014/647.

WMAS13. Yang Wang, Mark Manulis, Man Ho Au, and Willy Susilo. Relations among privacy notions for signcryption and key invisible "sign-then-encrypt". In *Information Security and Privacy: 18th Australasian Conference, ACISP 2013*, pages 187–202. Springer, 2013.

YDZ10. Moti Young, Alexander W Dent, and Yuliang Zheng. *Practical signcryption*. Springer Science & Business Media, 2010.

Zhe97. Yuliang Zheng. Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature)+ cost (encryption). In *Advances in Cryptology – CRYPTO 1997*, pages 165–179. Springer, 1997.

ZI98. Yuliang Zheng and Hideki Imai. How to construct efficient signcryption schemes on elliptic curves. *Information Processing Letters*, 68(5):227–233, 1998.

A Security of Signcryption

In this section we first state the separate definitions of confidentiality and authenticity (as found in the literature) for signcryption in the multi-user setting for an outsider adversary, and then show the equivalence of their coupling with our all-in-one formulation.

A.1 Standard Notions for Multi-User Outsider Security

In the literature, usually confidentiality of signcryption is defined using an experiment in which the adversary is supposed to distinguish signcryptexts of two messages chosen by him and signcrypted using the keys of some fixed sender and receiver, while having access to both flexible signcryption and unsigncryption oracles (which as usual, returns $\perp$ when the adversary queries previously obtained signcryptexts, in order to avoid trivial attacks), with the restriction that the unsigncryption oracle does not unsigncrypt the challenge signcryptexts under the fixed sender's public key. Using a standard hybrid argument, it is easy to see that such definitions are equivalent to real-or-random definitions, where the adversary must distinguish whether it is interacting with a true flexible signcryption oracle, or one which always signcrypts freshly uniform messages when queried with the fixed receiver's public key, and a flexible unsigncryption oracle which in both cases does not unsigncrypt a signcryptext under the fixed sender's public key previously returned by the signcryption oracle where it was signcrypted under the fixed receiver's public key (formally, in such cases the adversary obtains the special symbol $\perp$ back from the unsigncryption oracle). We formally define confidentiality of signcryption in the multi-user setting for an outsider adversary as follows.

Definition 6. *Let* $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ *be a signcryption scheme and* $\mathcal{A}$ *a probabilistic algorithm. We define the advantage of* $\mathcal{A}$ *in distinguishing* $\mathbf{Real}_{\Psi}^{\mathsf{MOS\text{-}Conf}}$ *and* $\mathbf{Ideal}_{\Psi}^{\mathsf{MOS\text{-}Conf}}$ *from Fig. 9 as*

$$\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MOS\text{-}Conf}} := \Pr\left[\mathcal{A}^{\mathbf{Real}_{\Psi}^{\mathsf{MOS\text{-}Conf}}} = 1\right] - \Pr\left[\mathcal{A}^{\mathbf{Ideal}_{\Psi}^{\mathsf{MOS\text{-}Conf}}} = 1\right].$$

We say that the scheme Ψ *is* MOS-Conf secure *if* $\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MOS\text{-}Conf}}$ *is negligible for all efficient adversaries* $\mathcal{A}$.

For authenticity we define an experiment in which the adversary has free access to both flexible signcryption and unsigncryption oracles, and wins if it is able to submit a signcryptext query[20] to the unsigncryption oracle which is *new* (was never returned by the signcryption oracle under the fixed receiver's public key) and *valid* (successfully decrypts to a message from the message space other than the special symbol $\perp$, under the fixed sender's public key). Such a definition is called strong-unforgeability of signcryptexts in the multi-user setting for an outsider adversary, and is stated formally as follows.

Definition 7. *Let* $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ *be a signcryption scheme and* $\mathcal{A}$ *a probabilistic algorithm. We define the advantage of* $\mathcal{A}$ *when interacting with* $\mathbf{Auth}_{\Psi}^{\mathsf{MOS}}$ *from Fig. 10 as*

$$\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MOS\text{-}Auth}} := \Pr\left[\mathcal{A}^{\mathbf{Auth}_{\Psi}^{\mathsf{MOS}}} \text{ sets } \mathsf{win}\right].$$

We say that the scheme Ψ *is* MOS-Auth secure *if* $\mathsf{Adv}_{\Psi,\mathcal{A}}^{\mathsf{MOS\text{-}Auth}}$ *is negligible for all efficient adversaries* $\mathcal{A}$.

[20] Note that such a definition of authenticity is equivalent to one where the adversary is required to *output* such a forgery, instead of just querying it to the unsigncryption oracle.

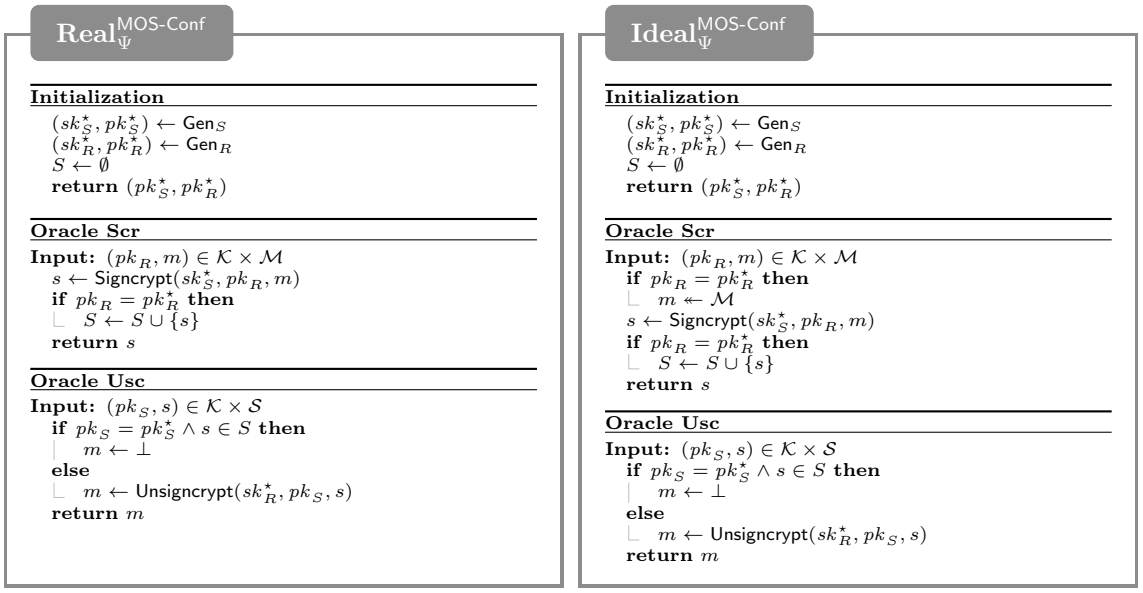

Fig. 9. $\mathbf{Real}_{\Psi}^{\text{MOS-Conf}}$ and $\mathbf{Ideal}_{\Psi}^{\text{MOS-Conf}}$ experiments for confidentiality for multi-user outsider secure signcryption schemes.

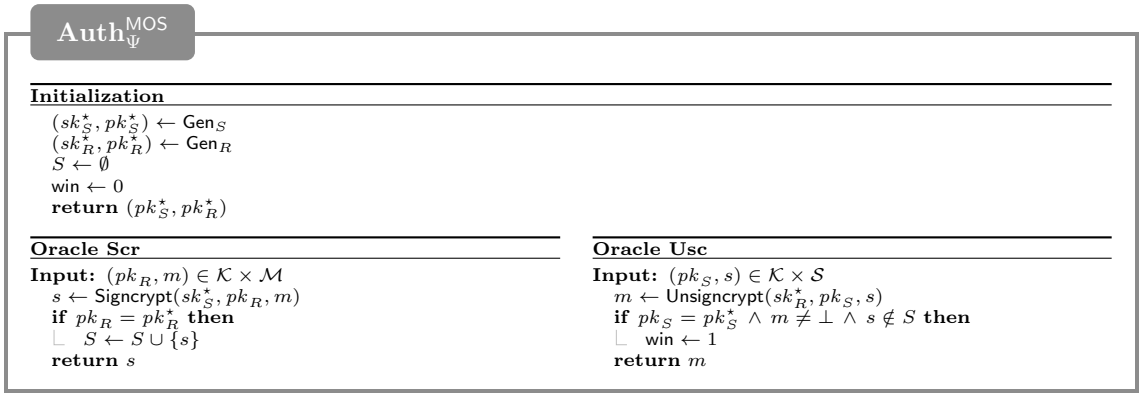

Fig. 10. $\mathbf{Auth}_{\Psi}^{\text{MOS}}$ experiment for authenticity for multi-user outsider secure signcryption schemes.

A.2 All-in-One Security

Outline. In this section we show that our all-in-one formulation of multi-user outsider secure signcryption (MOS security) is equivalent to the combination of the security notions of MOS-Conf (for confidentiality) and MOS-Auth (for authenticity). Concretely, we show Lemma 1 by showing that for every PPT adversary $\mathcal{A}_c$, there exists a PPT adversary $\mathcal{A}_s$ such that

$$\mathsf{Adv}^{\text{MOS-Conf}}_{\Psi,\mathcal{A}_c} \leq 2 \cdot \mathsf{Adv}^{\text{MOS}}_{\Psi,\mathcal{A}_s},$$

for every PPT $\mathcal{A}_a$, there exists a PPT adversary $\mathcal{A}_{s'}$ such that

$$\mathsf{Adv}^{\text{MOS-Auth}}_{\Psi,\mathcal{A}_a} \leq \mathsf{Adv}^{\text{MOS}}_{\Psi,\mathcal{A}_{s'}},$$

and for every PPT $\mathcal{A}_{s''}$, there exists a PPT adversary $\mathcal{A}_{a'}$ such that

$$\mathsf{Adv}^{\text{MOS}}_{\Psi,\mathcal{A}_{s''}} \leq \mathsf{Adv}^{\text{MOS-Conf}}_{\Psi,\mathcal{A}_{s''}} + \mathsf{Adv}^{\text{MOS-Auth}}_{\Psi,\mathcal{A}_{a'}}.$$

Adversary $\mathcal{A}_s(pk_S^\star, pk_R^\star)$ with oracle access to $\mathrm{Scr}(\cdot,\cdot)$ and $\mathrm{Usc}(\cdot,\cdot)$

Procedure Main

$\hat{b} \leftarrow\!\!\!\!\!\!\$\ \{0,1\}$
obtain b' by running $\mathcal{A}_c(pk_S^\star, pk_R^\star)$ with oracle access to $\mathbf{Scr}'(\cdot,\cdot)$ and $\mathbf{Usc}(\cdot,\cdot)$
return $1 - b' \oplus \hat{b}$

Procedure Scr′

Input: $(pk_R, m) \in \mathcal{K} \times \mathcal{M}$
if $\hat{b} = 0$ **then**
  $m \leftarrow\!\!\!\!\!\!\$\ \mathcal{M}$
$s \leftarrow \mathbf{Scr}(pk_R, m)$
return s

Fig. 11. Adversary $\mathcal{A}_s$ for $\mathbf{Real}_\Psi^{\mathsf{MOS}}/\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$.

Proof (of Lemma 1). We start by showing the first direction of the theorem, namely that MOS security implies both MOS-Conf and MOS-Auth security. For this, we first define in Fig. 11 a PPT adversary $\mathcal{A}_s$ interacting with either $\mathbf{Real}_\Psi^{\mathsf{MOS}}$ or $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$, which is given access to oracles $\mathbf{Scr}(\cdot,\cdot)$, $\mathbf{Usc}(\cdot,\cdot)$, and internally uses a PPT adversary $\mathcal{A}_c$ for either $\mathbf{Real}_\Psi^{\mathsf{MOS\text{-}Conf}}$ or $\mathbf{Ideal}_\Psi^{\mathsf{MOS\text{-}Conf}}$, defined in Fig. 9. Specifically, $\mathcal{A}_s$ chooses a random bit $\hat{b} \in \{0,1\}$ uniformly at random, invokes $\mathcal{A}_c$ with public keys $pk_S^\star$ and $pk_R^\star$ as argument, and forwards each unsigncryption query to its own oracle $\mathbf{Usc}(\cdot,\cdot)$ and each signcryption query of $\mathcal{A}_c$ to its own oracle $\mathbf{Scr}(\cdot,\cdot)$, but replaces the queried message with a uniform randomly selected message if $\hat{b} = 0$. Note that if $\mathcal{A}_s$ is interacting with $\mathbf{Real}_\Psi^{\mathsf{MOS}}$, it then perfectly simulates the game where $\mathcal{A}_c$ is interacting with either $\mathbf{Real}_\Psi^{\mathsf{MOS\text{-}Conf}}$ or $\mathbf{Ideal}_\Psi^{\mathsf{MOS\text{-}Conf}}$, while if it is interacting with $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$, then for both possible values of the bit $\hat{b}$ chosen by $\mathcal{A}_s$, the signcryptexts which $\mathcal{A}_c$ gets from $\mathbf{Usc}'(\cdot,\cdot)$ are identically distributed, that is, $\mathcal{A}_c$ has (information-theoretically) no information about $\hat{b}$, and thus when interacting with $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$, $\mathcal{A}_s$ wins with probability exactly $\frac{1}{2}$ (by simply guessing). We can now analyze the MOS advantage of $\mathcal{A}_s$ for Ψ as follows:

$$
\begin{aligned}
\mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\mathcal{A}_s} &= \Pr\left[\mathcal{A}_s^{\mathbf{Real}_\Psi^{\mathsf{MOS}}} = 1\right] - \Pr\left[\mathcal{A}_s^{\mathbf{Ideal}_\Psi^{\mathsf{MOS}}} = 1\right] \\
&= \frac{1}{2} \cdot \Pr\left[\mathcal{A}_s^{\mathbf{Real}_\Psi^{\mathsf{MOS}}} = 1 \;\middle|\; \hat{b} = 1\right] + \frac{1}{2} \cdot \Pr\left[\mathcal{A}_s^{\mathbf{Real}_\Psi^{\mathsf{MOS}}} = 1 \;\middle|\; \hat{b} = 0\right] - \frac{1}{2} \\
&= \frac{1}{2} \cdot \left(\Pr\left[\mathcal{A}_c^{\mathbf{Real}_\Psi^{\mathsf{MOS\text{-}Conf}}} = 1\right] - \left(1 - \Pr\left[\mathcal{A}_s^{\mathbf{Real}_\Psi^{\mathsf{MOS\text{-}Conf}}} = 0\right]\right)\right) \\
&= \frac{1}{2} \cdot \left(\Pr\left[\mathcal{A}_c^{\mathbf{Real}_\Psi^{\mathsf{MOS\text{-}Conf}}} = 1\right] - \Pr\left[\mathcal{A}_c^{\mathbf{Ideal}_\Psi^{\mathsf{MOS\text{-}Conf}}} = 1\right]\right) \\
&= \frac{1}{2} \cdot \mathsf{Adv}^{\mathsf{MOS\text{-}Conf}}_{\Psi,\mathcal{A}_c},
\end{aligned}
$$

where we used $\Pr\left[\mathcal{A}_s^{\mathbf{Ideal}_\Psi^{\mathsf{MOS}}} = 1\right] = \frac{1}{2}$ and the fact that, since $\mathcal{A}_s$ outputs $1 - b' \oplus \hat{b}$, if $\mathcal{A}_s$ outputs 1 then $\mathcal{A}_c$ has output $b' = \hat{b}$.

As the next step, we define in Fig. 12 a PPT adversary $\mathcal{A}_{s'}$ interacting with either $\mathbf{Real}_\Psi^{\mathsf{MOS}}$ or $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$, which is given access to oracles $\mathbf{Scr}(\cdot,\cdot)$, $\mathbf{Usc}(\cdot,\cdot)$, and internally uses a PPT adversary $\mathcal{A}_a$ for the $\mathbf{Auth}_\Psi^{\mathsf{MOS}}$ security game defined in Fig. 10. Specifically, $\mathcal{A}_{s'}$ invokes $\mathcal{A}_a$ with public keys $pk_S^\star$ and $pk_R^\star$ as argument, and forwards each signcryption query of $\mathcal{A}_a$ to its own oracle $\mathbf{Scr}(\cdot,\cdot)$ and in case $pk_R = pk_R^\star$, it keeps track of the returned signcryptexts (via the set S) as well as the original plaintext associated with every element of S (via the mapping M). $\mathcal{A}_{s'}$ also forwards unsigncryption queries of $\mathcal{A}_a$ to its own oracle $\mathbf{Usc}(\cdot,\cdot)$, but if $pk_S = pk_S^\star$ and some of those queries were already asked (and therefore $\perp$ is returned by $\mathbf{Usc}(\cdot,\cdot)$), then the corresponding plaintext is retrieved from M. Note that if $\mathcal{A}_{s'}$ is interacting with $\mathbf{Real}_\Psi^{\mathsf{MOS}}$, it then perfectly simulates the $\mathbf{Auth}_\Psi^{\mathsf{MOS}}$ experiment for $\mathcal{A}_a$, while if it is interacting with $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$, $\mathbf{Usc}'(\cdot,\cdot)$ always returns $\perp$

Adversary $\mathcal{A}_{s'}(pk_S^\star, pk_R^\star)$ with oracle access to $\mathrm{Scr}(\cdot,\cdot)$ and $\mathrm{Usc}(\cdot,\cdot)$

Procedure Main

$\tilde{b} \leftarrow 0$
$S \leftarrow \emptyset$
$M \leftarrow$ empty table
Run $\mathcal{A}_a(pk_S^\star, pk_R^\star)$ with oracle access to $\mathbf{Scr}'(\cdot,\cdot)$ and $\mathbf{Usc}'(\cdot,\cdot)$
return $\tilde{b}$

Procedure Scr′

Input: $(pk_R, m) \in \mathcal{K} \times \mathcal{M}$
$s \leftarrow \mathbf{Scr}(pk_R, m)$
if $pk_R = pk_R^\star$ **then**
  $S \leftarrow S \cup \{s\}$
  $M[s] \leftarrow m$
return s

Procedure Usc′

Input: $(pk_S, s) \in \mathcal{K} \times \mathcal{S}$
$m \leftarrow \mathbf{Usc}(pk_S, s)$
if $pk_S = pk_S^\star$ **then**
  if $m \neq \perp \wedge s \notin S$ **then**
    $\tilde{b} \leftarrow 1$
  if $m = \perp \wedge s \in S$ **then**
    $m \leftarrow M[s]$
return m

Fig. 12. Adversary $\mathcal{A}_{s'}$ for $\mathbf{Real}_\Psi^{\mathsf{MOS}}/\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$.

if $pk_S = pk_S^\star$, and thus $\tilde{b}$ will never be set to 1, and in turn $\mathcal{A}_{s'}$ will always (correctly) return 0 (recall that 0 identifies the ideal world). We can now analyze the MOS advantage of $\mathcal{A}_{s'}$ for Ψ as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\mathcal{A}_{s'}} &= \Pr\left[\mathcal{A}_s^{\mathbf{Real}_\Psi^{\mathsf{MOS}}} = 1\right] - \Pr\left[\mathcal{A}_s^{\mathbf{Ideal}_\Psi^{\mathsf{MOS}}} = 1\right] \\ &= \Pr\left[\mathcal{A}_s^{\mathbf{Auth}_\Psi^{\mathsf{MOS}}} \text{ sets } \mathsf{win}\right] - 0 \\ &= \mathsf{Adv}^{\mathsf{MOS\text{-}Auth}}_{\Psi,\mathcal{A}_a},\end{aligned}$$

where we used $\Pr\left[\mathcal{A}_s^{\mathbf{Ideal}_\Psi^{\mathsf{MOS}}} = 1\right] = 1 - \Pr\left[\mathcal{A}_s^{\mathbf{Ideal}_\Psi^{\mathsf{MOS}}} = 0\right] = 1 - 1 = 0$.

From the two calculations above, it follows that $\mathsf{Adv}^{\mathsf{MOS\text{-}Conf}}_{\Psi,\mathcal{A}_c} \leq 2 \cdot \mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\mathcal{A}_s}$ and $\mathsf{Adv}^{\mathsf{MOS\text{-}Auth}}_{\Psi,\mathcal{A}_a} \leq \mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\mathcal{A}_{s'}}$ as claimed, which means that if Ψ is MOS secure, then it is also MOS-Conf secure and MOS-Auth secure.

Finally, it remains to show the opposite direction of the theorem, that is, that together MOS-Conf and MOS-Auth security imply MOS security. In turn, this means that a PPT adversary for distinguishing $\mathbf{Real}_\Psi^{\mathsf{MOS}}$ and $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$ can be used to construct a PPT adversary for distinguishing $\mathbf{Real}_\Psi^{\mathsf{MOS\text{-}Conf}}$ and $\mathbf{Ideal}_\Psi^{\mathsf{MOS\text{-}Conf}}$ or a PPT adversary for winning $\mathbf{Auth}_\Psi^{\mathsf{MOS}}$. For this, let define the four games $\mathbf{Real}_\Psi^0, \mathbf{Ideal}_\Psi^0$ and $\mathbf{Real}_\Psi^1, \mathbf{Ideal}_\Psi^1$ as in Fig. 13.

The lines within the dashed box of $\mathbf{Ideal}_\Psi^0/\mathbf{Ideal}_\Psi^1$ highlight the difference of the two games with $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$. The line within the solid box in $\mathbf{Ideal}_\Psi^1$ denotes the only difference between $\mathbf{Ideal}_\Psi^0$ and $\mathbf{Ideal}_\Psi^1$. Clearly, $\mathbf{Ideal}_\Psi^0$ and $\mathbf{Ideal}_\Psi^1$ are identical until bad is set to 1. Note that $\mathbf{Ideal}_\Psi^0$ is identical to $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$ to an adversary, because setting bad to 1 does not affect the behavior of the adversary in any way (the adversary cannot learn that bad has been set). On the other hand, $\mathbf{Ideal}_\Psi^1$ is identical to $\mathbf{Ideal}_\Psi^{\mathsf{MOS\text{-}Conf}}$ to an adversary, because now the unsigncryption oracle does not return $\perp$ for valid signcryptexts if $pk_S = pk_S^\star$ and $b = 0$ (that is, in the random world), but it returns the correct message as it would when $b = 1$ (that is, in the real world), thus behaving exactly as the unsigncryption oracle of $\mathbf{Real}_\Psi^{\mathsf{MOS\text{-}Conf}}$ and $\mathbf{Ideal}_\Psi^{\mathsf{MOS\text{-}Conf}}$. Also note that $\mathbf{Real}_\Psi^{\mathsf{MOS}}$, $\mathbf{Real}_\Psi^0$, $\mathbf{Real}_\Psi^1$, and $\mathbf{Real}_\Psi^{\mathsf{MOS\text{-}Conf}}$ are all the same. We can therefore proceed in analyzing the MOS

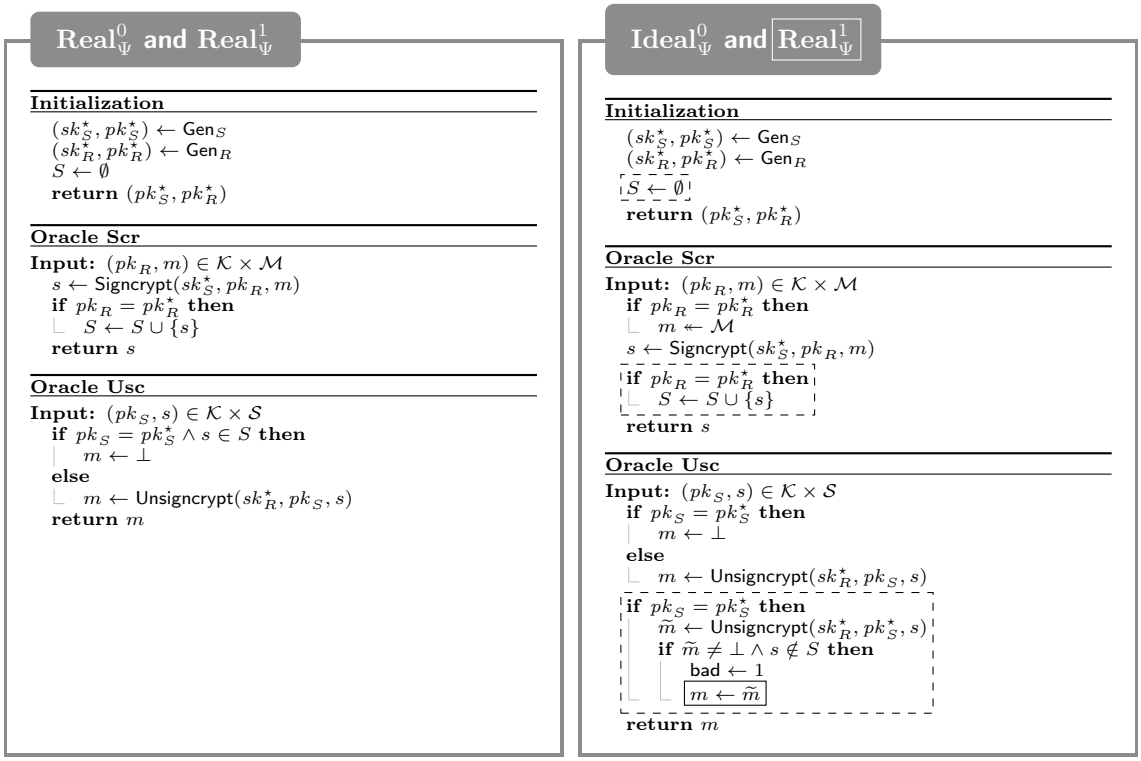

Fig. 13. Security games $\mathbf{Real}^0_\Psi, \mathbf{Ideal}^0_\Psi$ and $\mathbf{Real}^1_\Psi, \mathbf{Ideal}^1_\Psi$.

advantage of $\mathcal{A}_{s''}$ for Ψ as follows:

$$
\begin{aligned}
\mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\mathcal{A}_{s''}} &= \Pr\left[\mathcal{A}_{s''}^{\mathbf{Real}^{\mathsf{MOS}}_\Psi} = 1\right] - \Pr\left[\mathcal{A}_{s''}^{\mathbf{Ideal}^{\mathsf{MOS}}_\Psi} = 1\right] \\
&= \Pr\left[\mathcal{A}_{s''}^{\mathbf{Real}^0_\Psi} = 1\right] - \Pr\left[\mathcal{A}_{s''}^{\mathbf{Ideal}^0_\Psi} = 1\right] \\
&= \Pr\left[\mathcal{A}_{s''}^{\mathbf{Real}^0_\Psi} = 1\right] - \Pr\left[\mathcal{A}_{s''}^{\mathbf{Ideal}^0_\Psi} = 1\right] + \\
&\quad + \Pr\left[\mathcal{A}_{s''}^{\mathbf{Ideal}^1_\Psi} = 1\right] - \Pr\left[\mathcal{A}_{s''}^{\mathbf{Ideal}^1_\Psi} = 1\right] \\
&\leq \Pr\left[\mathcal{A}_{s''}^{\mathbf{Ideal}^1_\Psi} \text{ sets } \mathsf{bad}\right] + \Pr\left[\mathcal{A}_{s''}^{\mathbf{Real}^{\mathsf{MOS\text{-}Conf}}_\Psi} = 1\right] + \\
&\quad - \Pr\left[\mathcal{A}_{s''}^{\mathbf{Ideal}^{\mathsf{MOS\text{-}Conf}}_\Psi} = 1\right] \qquad (2) \\
&= \Pr\left[\mathcal{A}_{a'}^{\mathbf{Auth}^{\mathsf{MOS}}_\Psi} \text{ sets } \mathsf{win}\right] + \mathsf{Adv}^{\mathsf{MOS\text{-}Conf}}_{\Psi,\mathcal{A}_{s''}} \qquad (3) \\
&= \mathsf{Adv}^{\mathsf{MOS\text{-}Auth}}_{\Psi,\mathcal{A}_{a'}} + \mathsf{Adv}^{\mathsf{MOS\text{-}Conf}}_{\Psi,\mathcal{A}_{s''}},
\end{aligned}
$$

where for (2) we used from [Mau02, Theorem 1] (or equivalently, a concretization thereof for code-based games in [BR06, Lemma 2 (*"Fundameltal Lemma of Game-Playing"*)]) and the fact that $\mathbf{Ideal}^1_\Psi$ and $\mathbf{Ideal}^0_\Psi$ behave identically until bad is set, and for (3) we used the reduction implemented by the adversary $\mathcal{A}_{a'}$ for $\mathbf{Auth}^{\mathsf{MOS}}_\Psi$ described from Fig. 14, which simply simulates $\mathbf{Ideal}^1_\Psi$ to $\mathcal{A}_{s''}$. Clearly, as soon as $\mathcal{A}_{s''}$ queries a winning query to $\mathbf{Usc}'(\cdot,\cdot)$, so does $\mathcal{A}_{a'}$ to $\mathbf{Usc}(\cdot,\cdot)$. Note that the bit output by $\mathcal{A}_{s''}$ is useless and thus ignored.

Therefore, $\mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\mathcal{A}_{s''}} \leq \mathsf{Adv}^{\mathsf{MOS\text{-}Conf}}_{\Psi,\mathcal{A}_{s''}} + \mathsf{Adv}^{\mathsf{MOS\text{-}Auth}}_{\Psi,\mathcal{A}_{a'}}$ as claimed, which means that if Ψ is both MOS-Conf secure and MOS-Auth secure, then it is also MOS secure. □

Adversary $\mathcal{A}_{a'}(pk_S^\star, pk_R^\star)$ with oracle access to $\mathrm{Scr}(\cdot,\cdot)$ and $\mathrm{Usc}(\cdot,\cdot)$

Procedure Main

$S \leftarrow \emptyset$
Run $\mathcal{A}_{s''}(pk_S^\star, pk_R^\star)$ with oracle access to $\mathbf{Scr}'(\cdot,\cdot)$ and $\mathbf{Usc}'(\cdot,\cdot)$

Procedure Scr′

Input: $(pk_R, m) \in \mathcal{K} \times \mathcal{M}$
if $pk_R = pk_R^\star$ **then**
  $m \twoheadleftarrow \mathcal{M}$
$s \leftarrow \mathbf{Scr}(pk_R, m)$
if $pk_R = pk_R^\star$ **then**
  $S \leftarrow S \cup \{s\}$
return s

Procedure Usc′

Input: $(pk_S, s) \in \mathcal{K} \times \mathcal{S}$
if $pk_S \neq pk_S^\star \wedge s \in S$ **then**
  $m \leftarrow \bot$
else
  $m \leftarrow \mathbf{Usc}(pk_S, s)$
return m

Fig. 14. Adversary $\mathcal{A}_{a'}$ for $\mathbf{Auth}_\Psi^{\mathsf{MOS}}$.

B Proofs of Lemma 2, Lemma 3, and Lemma 4

B.1 Proof of the First Game-Hop

Proof (of Lemma 2). We need to provide a reduction $\rho_{\mathcal{Z}}(\cdot)$ so that distinguishing $\mathbf{Real}_\Psi^{\mathsf{MOS}}$ from $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$ can be reduced to distinguishing $\mathbf{H}_0^{\mathcal{Z}}$ from $\mathbf{H}_1^{\mathcal{Z}}$. Let the system $\mathbf{D}$ be a distinguisher for $\mathbf{H}_0^{\mathcal{Z}}$ and $\mathbf{H}_1^{\mathcal{Z}}$, and for the set $\mathcal{I} \coloneqq \{i \in [n] \,|\, \mathtt{M}_i \notin \mathcal{Z}\}$ of indexes of uncorrupted parties,[21] let $\mathcal{L} \coloneqq \mathcal{I} \times \mathcal{I}$ denote the set of all ℓ^2 two-element[22] tuples over the set $\mathcal{I}$, with $\ell \coloneqq |\mathcal{I}|$. Let also fix an order over $\mathcal{L}$, that is, fix some efficiently computable bijection $\omega : [\ell^2] \to \mathcal{L}$ as well as its efficiently computable inverse map $\omega^{-1} : \mathcal{L} \to [\ell^2]$. We construct an adversary $\mathcal{A}$ for distinguishing $\mathbf{Real}_\Psi^{\mathsf{MOS}}$ from $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$ using distinguisher $\mathbf{D}$ via a reduction $\rho_{\mathcal{Z}}(\cdot)$, denoted $\mathcal{A} \coloneqq \rho_{\mathcal{Z}}(\mathbf{D})$.

The reduction works by first choosing an index t uniformly at random from $[\ell^2]$, and then computing the pair of indexes of *designated parties* $(S, R) \coloneqq \omega(t)$ (thus $S, R \in [n]$), corresponding to the *designated sender* $\mathtt{P}_S$ and the *designated receiver* $\mathtt{P}_R$, respectively. In the following, let $\mathcal{A}_t$ be the same as $\mathcal{A}$ but where the index t is fixed instead of uniformly randomly selected. For a random variable T uniformly distributed over $[\ell^2]$, this implies $\mathcal{A} = \mathcal{A}_T$. Recall that when $\mathcal{A}_t$ interacts with $\mathbf{Real}_\Psi^{\mathsf{MOS}}$ or $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$, it receives a pair of public keys, a sender public key $pk_S^\star$, which will be set as $\mathtt{P}_S$'s sender public key, and receiver public key $pk_R^\star$, which will be set as $\mathtt{P}_R$'s receiver public key. Upon registration of an identity via an honest interface, $\mathcal{A}_t$ generates and stores both sender and receiver key-pairs for the respective user, except that for party $\mathtt{P}_S$ only a receiver key-pair is generated and stored and for party $\mathtt{P}_R$ only a sender key-pair is generated and stored (recall that for both those parties $\mathcal{A}_t$ uses one of the public keys provided by either $\mathbf{Real}_\Psi^{\mathsf{MOS}}$ or $\mathbf{Ideal}_\Psi^{\mathsf{MOS}}$, whereas the corresponding secret keys are "hard-coded" into the provided oracles). Upon registration directly at interface $\mathtt{E}$, $\mathcal{A}_t$ internally stores the mapping of this identity to the registered public key. Whenever `reveal` is input at interface $\mathtt{M}_i \in \mathcal{Z}$ (i.e., $i \notin \mathcal{I}$), $\mathcal{A}_t$ returns the two generated key-pairs to the distinguisher $\mathbf{D}$ (and $\bot$ if $\mathtt{M}_i \notin \mathcal{Z}$, i.e., $i \in \mathcal{I}$). Similarly, it is easy for $\mathcal{A}_t$ to generate the mapping of identities to public keys when asked to reveal this information via a query `fetchAll`. We now describe the behavior of $\mathcal{A}_t$ on the remaining inputs.

On input $(\mathtt{send}, m, \mathsf{ID})$ **at interface** $\mathtt{P}_i$**:** The reduction $\mathcal{A}_t$ retrieves $\mathsf{ID}_{\mathtt{P}_i}$ and $\mathtt{P}_j$ from ID (recall that $i, j \in [n]$), and if both parties have previously successfully registered, $\mathcal{A}_t$ performs the following case distinction:

[21] Recall that the corruption set is defined as $\mathcal{Z} \subseteq \{\mathtt{M}_i \,|\, i \in [n]\}$, and is known to the adversary.
[22] Note that same-element tuples, i.e. $(\mathsf{ID}, \mathsf{ID})$, for $\mathsf{ID} \in \mathcal{I}$, are also included in $\mathcal{L}$.

- If $(i,j) \notin \mathcal{L}$, then the message m is signcrypted into s using $\mathtt{P}_i$'s sender private key and $\mathtt{P}_j$'s receiver public key, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$.
- If $(i,j) \in \mathcal{L}$, then the further case distinction is made:
  - If $\omega^{-1}(i,j) < t$, then the message m is replaced by a uniform message m^* of the same length which is signcrypted into s using $\mathtt{P}_i$'s sender private key and $\mathtt{P}_j$'s receiver public key, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$. Note that if $i = S$ (that is, the sender is the designated sender $\mathtt{P}_S$), then $\mathcal{A}_t$ uses the provided signcryption oracle. In any case, the mapping $((s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}) \mapsto m)$ is stored into a table M for later reference.
  - If $\omega^{-1}(i,j) > t$, then the message m is signcrypted into s using $\mathtt{P}_i$'s sender private key and $\mathtt{P}_j$'s receiver public key, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$. Note that if $i = S$ (that is, the sender is the designated sender $\mathtt{P}_S$), then $\mathcal{A}_t$ uses the provided signcryption oracle.
  - If $\omega^{-1}(i,j) = t$ (that is, the parties are exactly the designated sender $\mathtt{P}_S$ and receiver $\mathtt{P}_R$), then the message m is signcrypted into s using the provided signcryption oracle, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$. Moreover, in this case the mapping $((s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}) \mapsto m)$ is stored into table M.

On input $(\mathtt{inject}, s, \mathsf{ID}, \mathsf{ID}')$ **at interface** $\mathtt{E}$**:** The reduction $\mathcal{A}_t$ retrieves $\mathsf{ID}_{\mathtt{P}_i}$ from ID and $\mathtt{P}_j$ from ID' (recall that $i, j \in [n]$), and if both parties have previously successfully registered, $\mathcal{A}_t$ performs the following case distinction:

- If $(i,j) \notin \mathcal{L}$, then the signcryptext s is unsigncrypted into m using $\mathtt{P}_j$'s receiver private key and $\mathtt{P}_i$'s sender public key, and (m, ID) is output at $\mathtt{P}_j$.
- If $(i,j) \in \mathcal{L}$, then the further case distinction is made:
  - If $\omega^{-1}(i,j) < t$, then, if possible, the corresponding value m is retrieved from table M and (m, ID) is output at $\mathtt{P}_j$. In case no mapping exists, no output is produced for interface $\mathtt{P}_j$.
  - If $\omega^{-1}(i,j) > t$, then the signcryptext s is unsigncrypted into m using $\mathtt{P}_j$'s receiver private key and $\mathtt{P}_i$'s sender public key, and (m, ID) is output at $\mathtt{P}_j$. Note that if $j = R$ (that is, the receiver is the designated receiver $\mathtt{P}_R$), then $\mathcal{A}_t$ uses the provided unsigncryption oracle.
  - If $\omega^{-1}(i,j) = t$ (that is, the parties are exactly the designated sender $\mathtt{P}_S$ and receiver $\mathtt{P}_R$), then if possible the corresponding value m is retrieved from table M, otherwise the signcryptext s is unsigncrypted into m using the provided unsigncryption oracle. If a value $m \neq \perp$ can be obtained this way, (m, ID) is output at $\mathtt{P}_j$ and otherwise, no output is produced.

Towards a standard hybrid argument, note that:

- $\Pr\left[\mathcal{A}_1^{\mathbf{Real}_\Psi^{\mathsf{MOS}}} = 1\right] = \Pr\left[\mathbf{DH}_0^{\mathcal{Z}} = 1\right]$, that is, if the reduction adversary is connected to real oracles, and it sets (S,R) as the first pair of indexes in $\mathcal{L}$ according to the ordering induced by ω, then for the distinguisher $\mathbf{D}$ the view is the same as if it was connected to the real world resource $\mathbf{H}_0^{\mathcal{Z}}$, since all pairs of parties with indexes after (S,R) as well as $(\mathtt{P}_S, \mathtt{P}_R)$ act as real users.
- $\Pr\left[\mathcal{A}_{\ell^2}^{\mathbf{Ideal}_\Psi^{\mathsf{MOS}}} = 1\right] = \Pr\left[\mathbf{DH}_1^{\mathcal{Z}} = 1\right]$, that is, if the reduction adversary is connected to ideal oracles, and it sets (S,R) as the last pair of indexes in $\mathcal{L}$ according to the ordering induced by ω, then for the distinguisher $\mathbf{D}$ the view is the same as if it was connected to the ideal world resource $\mathbf{H}_1^{\mathcal{Z}}$, since all pairs of parties with indexes before (S,R) as well as $(\mathtt{P}_S, \mathtt{P}_R)$ act as ideal users. In particular, upon an input $(\mathtt{inject}, \cdot, \cdot, \cdot)$, if no corresponding input was given to the system before, no message is output, i.e., in both systems, even if the condition of bad_1 would be satisfied, both system define the resulting plaintext to be $\perp$.
- $\Pr\left[\mathcal{A}_t^{\mathbf{Ideal}_\Psi^{\mathsf{MOS}}} = 1\right] = \Pr\left[\mathcal{A}_{t+1}^{\mathbf{Real}_\Psi^{\mathsf{MOS}}} = 1\right]$, that is, if the reduction adversary $\mathcal{A}_t$ is connected to ideal oracles, then for the distinguisher $\mathbf{D}$ the view is the same as if it was being used by the reduction adversary $\mathcal{A}_{t+1}$ when connected to real oracles.

We can now conclude the proof using the hybrid argument:

$$\mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\mathcal{A}} = \sum_{t=1}^{\ell^2} \mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\mathcal{A}_t} \cdot \Pr[T=t] \tag{4}$$

$$= \frac{1}{\ell^2} \sum_{t=1}^{\ell^2} \left(\Pr\left[\mathcal{A}_t^{\mathbf{Real}^{\mathsf{MOS}}_{\Psi}} = 1\right] - \Pr\left[\mathcal{A}_t^{\mathbf{Ideal}^{\mathsf{MOS}}_{\Psi}} = 1\right] \right) \tag{5}$$

$$= \frac{1}{\ell^2} \sum_{t=1}^{\ell^2} \left(\Pr\left[\mathcal{A}_t^{\mathbf{Real}^{\mathsf{MOS}}_{\Psi}} = 1\right] - \Pr\left[\mathcal{A}_{t+1}^{\mathbf{Real}^{\mathsf{MOS}}_{\Psi}} = 1\right] \right) \tag{6}$$

$$= \frac{1}{\ell^2} \left(\Pr\left[\mathcal{A}_1^{\mathbf{Real}^{\mathsf{MOS}}_{\Psi}} = 1\right] - \Pr\left[\mathcal{A}_{\ell^2+1}^{\mathbf{Real}^{\mathsf{MOS}}_{\Psi}} = 1\right] \right) \tag{7}$$

$$= \frac{1}{\ell^2} \left(\Pr\left[\mathcal{A}_1^{\mathbf{Real}^{\mathsf{MOS}}_{\Psi}} = 1\right] - \Pr\left[\mathcal{A}_{\ell^2}^{\mathbf{Ideal}^{\mathsf{MOS}}_{\Psi}} = 1\right] \right) \tag{8}$$

$$= \frac{1}{\ell^2} \left(\Pr\left[\mathbf{DH}_0^{\mathcal{Z}} = 1\right] - \Pr\left[\mathbf{DH}_1^{\mathcal{Z}} = 1\right] \right) \tag{9}$$

$$= \frac{1}{\ell^2} \cdot \Delta^{\mathbf{D}}(\mathbf{H}_0^{\mathcal{Z}}, \mathbf{H}_1^{\mathcal{Z}}), \tag{10}$$

where for (4) we used $\mathcal{A} = \mathcal{A}_T$ and the law of total probability, for (5) we used $\Pr[T=t] = \frac{1}{\ell^2}$ (for any $t \in [\ell^2]$), for (6), (8), and (9) we used the three equalities outlined above, for (7) we used the hybrid argument, and for (10) we used the definition of Δ. This proves that systems $\mathbf{H}_0^{\mathcal{Z}}$ and $\mathbf{H}_1^{\mathcal{Z}}$ are computationally indistinguishable, that is, for any distinguisher $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathbf{H}_0^{\mathcal{Z}}, \mathbf{H}_1^{\mathcal{Z}}) \leq n^2 \cdot \mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\rho_{\mathcal{Z}}(\mathbf{D})},$$

since $\ell \leq n$. □

B.2 Proof of the Second Game-Hop

Proof (of Lemma 3). We need to provide a reduction $\rho_{\mathcal{Z}}(\cdot)$ so that winning $\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}$ can be reduced to distinguishing $\mathbf{H}_1^{\mathcal{Z}}$ from $\mathbf{H}_2^{\mathcal{Z}}$. Let the system $\mathbf{D}$ be a distinguisher for $\mathbf{H}_1^{\mathcal{Z}}$ and $\mathbf{H}_2^{\mathcal{Z}}$, and let $\mathcal{I} \coloneqq \{i \in [n] \,|\, \mathtt{M}_i \notin \mathcal{Z}\}$ be the set of indexes of uncorrupted parties, with $\ell \coloneqq |\mathcal{I}|$. Let also fix an order over $\mathcal{I}$, that is, fix some efficiently computable bijection $\omega : [\ell] \to \mathcal{I}$ as well as its efficiently computable inverse map $\omega^{-1} : \mathcal{I} \to [\ell]$. We construct an adversary $\mathcal{A}$ for winning $\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}$ using distinguisher $\mathbf{D}$ via a reduction $\rho_{\mathcal{Z}}(\cdot)$, denoted $\mathcal{A} \coloneqq \rho_{\mathcal{Z}}(\mathbf{D})$.

The reduction works by first choosing an index t uniformly at random from $[\ell]$, and then computing the index $S \coloneqq \omega(t)$, $S \in [n]$, of a *designated sender* $\mathtt{P}_S$. In the following, let $\mathcal{A}_t$ be the same as $\mathcal{A}$ but where the index t is fixed instead of uniformly randomly selected. For a random variable T uniformly distributed over $[\ell]$, this implies $\mathcal{A} = \mathcal{A}_T$. Recall that when $\mathcal{A}_t$ interacts with $\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}$, it receives a sender public key $pk^{\star}_S$, which will be set as $\mathtt{P}_S$'s receiver public key. Upon honest registration, $\mathcal{A}_t$ stores for each user both a sender key-pair (generated using Gen_S) and a receiver key-pair (generated using the oracle **Gen** provided by $\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}$), except that for party $\mathtt{P}_S$ only a receiver key-pair is generated (recall that for $\mathtt{P}_S$, $\mathcal{A}_t$ uses the sender public key provided by $\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}$, whereas the corresponding sender secret key is "hard-coded" into the provided signcryption oracle). Dishonest registrations and queries `fetchAll` are handled as in the proof of Lemma 2. Moreover, whenever `reveal` is input at interface $\mathtt{M}_i \in \mathcal{Z}$ (i.e., $i \notin \mathcal{I}$), $\mathcal{A}_t$ returns the two generated key-pairs to the distinguisher $\mathbf{D}$ (and $\perp$ if $\mathtt{M}_i \notin \mathcal{Z}$, i.e., $i \in \mathcal{I}$). We now describe the behavior of $\mathcal{A}_t$ on the remaining inputs.

On input $(\mathtt{send}, m, \mathsf{ID})$ **at interface** $\mathtt{P}_i$**:** The reduction $\mathcal{A}_t$ retrieves $\mathsf{ID}_{\mathtt{P}_i}$ and $\mathtt{P}_j$ from ID (recall that $i, j \in [n]$), and if both parties have previously successfully registered, $\mathcal{A}_t$ performs the following case distinction:

- If $i \notin \mathcal{I}$, then the message m is signcrypted into s using $\mathtt{P}_i$'s sender private key and $\mathtt{P}_j$'s receiver public key, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$.
- If $i \in \mathcal{I}$, then the further case distinction is made. First, if $(i,j) \in \mathcal{L}$ then the message m is first replaced replaced by a uniform message m^* of the same length. Otherwise, m is left untouched. Continue with the remaining two conditions:
  - If $\omega^{-1}(i) \neq t$, then the message m is signcrypted into s using $\mathtt{P}_i$'s sender private key and $\mathtt{P}_j$'s receiver public key, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$. The mapping $((s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}) \mapsto m)$ is stored into a table M for later reference.
  - If $\omega^{-1}(i) = t$ (that is, the party $\mathtt{P}_i$ is exactly the designated sender $\mathtt{P}_S$), then the message m is signcrypted into s using the provided signcryption oracle, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is then output at $\mathtt{E}$. The mapping $((s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}) \mapsto m)$ is stored into a table M for later reference.

On input $(\mathtt{inject}, s, \mathsf{ID}, \mathsf{ID}')$ **at interface** $\mathtt{E}$**:** The reduction $\mathcal{A}_t$ retrieves $\mathsf{ID}_{\mathtt{P}_i}$ from ID and $\mathtt{P}_j$ from ID' (recall that $i, j \in [n]$), and if both parties have previously successfully registered, $\mathcal{A}_t$ performs the following case distinction:

- If $i \notin \mathcal{I}$, then the signcryptext s is unsigncrypted into m using $\mathtt{P}_j$'s receiver private key and $\mathtt{P}_i$'s sender public key, and (m, ID) is output at $\mathtt{P}_j$.
- If $i \in \mathcal{I}$, then the further case distinction is made:
  - If $(i,j) \in \mathcal{L}$ **or** $\omega^{-1}(i) < t$ then, if possible, the corresponding value m is retrieved from table M and (m, ID) is output at $\mathtt{P}_j$. In case no mapping exists, no output is produced for interface $\mathtt{P}_j$.
  - If $\omega^{-1}(i) > t$ and $j \notin \mathcal{I}$, then the signcryptext s is unsigncrypted into m using $\mathtt{P}_j$'s receiver private key and $\mathtt{P}_i$'s sender public key, and (m, ID) is output at $\mathtt{P}_j$.
  - If $\omega^{-1}(i) = t$ (that is, the party $\mathtt{P}_i$ is exactly the designated sender $\mathtt{P}_S$) and $j \notin \mathcal{I}$, then the signcryptext s is unsigncrypted into m using the provided unsigncryption oracle, and (m, ID) is output at $\mathtt{P}_j$.

Note that we can naturally define further hybrid systems $\mathbf{H}^{\mathcal{Z}}_{1,t}$ and $\mathbf{H}^{\mathcal{Z}}_{2,t}$, for $t \in [\ell]$, such that $\mathbf{H}^{\mathcal{Z}}_{2,t}$ and $\mathbf{H}^{\mathcal{Z}}_{1,t+1}$ are exactly the system that $\mathcal{A}_t$ emulates to $\mathbf{D}$ In particular the flag win in $\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}$ is set if and only if the flag bad_2 is set upon successfully injecting a forged signcryptext on behalf of $\omega(t)$. For sake of clarity, we will refer to this new flag by bad^t_2. Moreover, systems $H^{\mathcal{Z}}_{1,t}$ and $\mathbf{H}^{\mathcal{Z}}_{2,t}$ have an equivalent behavior unless this flag is set. Since the behavior of the fine-grained hybrids is analogous to the one described above for $\mathcal{A}_t$, we refrain from formally describing $\mathbf{H}^{\mathcal{Z}}_{1,t}$ and $\mathbf{H}^{\mathcal{Z}}_{2,t}$. Towards a standard hybrid argument, note that:

- $\Pr\big[\mathbf{DH}^{\mathcal{Z}}_{1,1} = 1\big] = \Pr\big[\mathbf{DH}^{\mathcal{Z}}_{1} = 1\big]$, that is, the view of distinguisher $\mathbf{D}$ when interacting with $\mathbf{H}^{\mathcal{Z}}_{1,1}$ or $\mathbf{H}^{\mathcal{Z}}_{1}$ is the same.
- $\Pr\Big[\mathbf{DH}^{\mathcal{Z}}_{2,\ell} = 1\Big] = \Pr\big[\mathbf{DH}^{\mathcal{Z}}_{2} = 1\big]$, that is, the view of distinguisher $\mathbf{D}$ when interacting with $\mathbf{H}^{\mathcal{Z}}_{2,\ell}$ or $\mathbf{H}^{\mathcal{Z}}_{2}$ is the same.
- $\Pr\big[\mathbf{DH}^{\mathcal{Z}}_{2,t} = 1\big] = \Pr\big[\mathbf{DH}^{\mathcal{Z}}_{1,t+1} = 1\big]$, that is, the view of distinguisher $\mathbf{D}$ when interacting with $\mathbf{H}^{\mathcal{Z}}_{2,t}$ or $\mathbf{H}^{\mathcal{Z}}_{1,t+1}$ is the same.

We can now conclude the proof:

$$\mathsf{Adv}^{\mathsf{MIS\text{-}Auth}}_{\Psi,\mathcal{A}} = \sum_{t=1}^{\ell} \mathsf{Adv}^{\mathsf{MIS\text{-}Auth}}_{\Psi,\mathcal{A}_t} \cdot \Pr[T=t] \tag{11}$$

$$= \frac{1}{\ell} \sum_{t=1}^{\ell} \Pr\left[\mathcal{A}_t^{\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}} \text{ sets } \mathsf{win}\right] \tag{12}$$

$$= \frac{1}{\ell} \sum_{t=1}^{\ell} \Pr\left[\mathbf{DH}^{\mathcal{Z}}_{2,t} \text{ sets } \mathsf{bad}^t_2\right] \tag{13}$$

$$\geq \frac{1}{\ell} \sum_{t=1}^{\ell} \left(\Pr\left[\mathbf{DH}^{\mathcal{Z}}_{1,t}=1\right] - \Pr\left[\mathbf{DH}^{\mathcal{Z}}_{2,t}=1\right]\right) \tag{14}$$

$$= \frac{1}{\ell} \sum_{t=1}^{\ell} \left(\Pr\left[\mathbf{DH}^{\mathcal{Z}}_{1,t}=1\right] - \Pr\left[\mathbf{DH}^{\mathcal{Z}}_{1,t+1}=1\right]\right) \tag{15}$$

$$= \frac{1}{\ell} \left(\Pr\left[\mathbf{DH}^{\mathcal{Z}}_{1,1}=1\right] - \Pr\left[\mathbf{DH}^{\mathcal{Z}}_{1,\ell+1}=1\right]\right) \tag{16}$$

$$= \frac{1}{\ell} \left(\Pr\left[\mathbf{DH}^{\mathcal{Z}}_{1,1}=1\right] - \Pr\left[\mathbf{DH}^{\mathcal{Z}}_{2,\ell}=1\right]\right) \tag{17}$$

$$= \frac{1}{\ell} \left(\Pr\left[\mathbf{DH}^{\mathcal{Z}}_{1}=1\right] - \Pr\left[\mathbf{DH}^{\mathcal{Z}}_{2}=1\right]\right) \tag{18}$$

$$= \frac{1}{\ell} \cdot \Delta^{\mathbf{D}}(\mathbf{H}^{\mathcal{Z}}_1, \mathbf{H}^{\mathcal{Z}}_2), \tag{19}$$

where for (11) we used $\mathcal{A} = \mathcal{A}_T$ and the law of total probability, for (12) we used $\Pr[T=t] = \frac{1}{\ell}$ (for any $t \in [\ell]$) and the definition of the advantage of $\mathcal{A}_t$ when interacting with $\mathbf{Auth}^{\mathsf{MIS}}_{\Psi}$, for (13) we used the fact that $\mathcal{A}_t$ perfectly simulates $\mathbf{H}^{\mathcal{Z}}_{2,t}$ to $\mathbf{D}$, for (14) we used [Mau02, Theorem 1] (or equivalently, a concretization thereof for code-based games in [BR06, Lemma 2 (*"Fundameltal Lemma of Game-Playing"*)]), for (15), (17), and (18) we used the three equalities outlined above, for (16) we used the hybrid argument, and for (19) we used the definition of Δ. This proves that systems $\mathbf{H}^{\mathcal{Z}}_1$ and $\mathbf{H}^{\mathcal{Z}}_2$ are computationally indistinguishable, that is, for any distinguisher $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathbf{H}^{\mathcal{Z}}_1, \mathbf{H}^{\mathcal{Z}}_2) \leq n \cdot \mathsf{Adv}^{\mathsf{MIS\text{-}Auth}}_{\Psi,\rho_{\mathcal{Z}}(\mathbf{D})},$$

since $\ell \leq n$. □

B.3 Proof of the Third Game-Hop

Proof (of Lemma 4). We need to provide a reduction $\rho_{\mathcal{Z}}(\cdot)$ so that distinguishing $\mathbf{Real}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$ from $\mathbf{Ideal}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$ can be reduced to distinguishing $\mathbf{H}^{\mathcal{Z}}_2$ from $\mathbf{H}^{\mathcal{Z}}_3$. Let the system $\mathbf{D}$ be a distinguisher for $\mathbf{H}^{\mathcal{Z}}_2$ and $\mathbf{H}^{\mathcal{Z}}_3$, and let $\mathcal{I} \coloneqq \{i \in [n] \,|\, \mathtt{M}_i \notin \mathcal{Z}\}$ be the set of indexes of uncorrupted parties, with $\ell \coloneqq |\mathcal{I}|$. Let also fix an order over $\mathcal{I}$, that is, fix some efficiently computable bijection $\omega : [\ell] \to \mathcal{I}$ as well as its efficiently computable inverse map $\omega^{-1} : \mathcal{I} \to [\ell]$. We construct an adversary $\mathcal{A}$ for distinguishing $\mathbf{Real}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$ from $\mathbf{Ideal}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$ using distinguisher $\mathbf{D}$ via a reduction $\rho_{\mathcal{Z}}(\cdot)$, denoted $\mathcal{A} \coloneqq \rho_{\mathcal{Z}}(\mathbf{D})$.

The reduction works by first choosing an index t uniformly at random from $[\ell]$, and then computing the index $R \coloneqq \omega(t)$, $R \in [n]$, of a *designated receiver* $\mathtt{P}_R$. In the following, let $\mathcal{A}_t$ be the same as $\mathcal{A}$ but where the index t is fixed instead of uniformly randomly selected. For a random variable T uniformly distributed over $[\ell]$, this implies $\mathcal{A} = \mathcal{A}_T$. Recall that when $\mathcal{A}_t$ interacts with $\mathbf{Real}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$ or $\mathbf{Ideal}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$, it receives a receiver public key $pk^{\star}_R$, which will be set as $\mathtt{P}_R$'s receiver public key. Upon registration, $\mathcal{A}_t$ stores for each user both a receiver key-pair (generated using Gen_R) and a sender key-pair (generated using the oracle **Gen** provided by either $\mathbf{Real}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$ or $\mathbf{Ideal}^{\mathsf{MIS\text{-}Conf}}_{\Psi}$), except that for party $\mathtt{P}_R$ only a sender key-pair is generated

(recall that for $\mathtt{P}_R$, $\mathcal{A}_t$ uses the receiver public key provided by either $\mathbf{Real}_{\Psi}^{\mathsf{MIS\text{-}Conf}}$ or $\mathbf{Ideal}_{\Psi}^{\mathsf{MIS\text{-}Conf}}$, whereas the corresponding receiver secret key is "hard-coded" into the provided unsigncryption oracle). Whenever `reveal` is input at interface $\mathtt{M}_i \in \mathcal{Z}$ (i.e., $i \notin \mathcal{I}$), $\mathcal{A}_t$ returns the two generated key-pairs to the distinguisher $\mathbf{D}$ (and $\perp$ if $\mathtt{M}_i \notin \mathcal{Z}$, i.e., $i \in \mathcal{I}$). We now describe the behavior of $\mathcal{A}_t$ on the remaining inputs.

On input $(\mathtt{send}, m, \mathsf{ID})$ **at interface** $\mathtt{P}_i$**:** The reduction $\mathcal{A}_t$ retrieves $\mathsf{ID}_{\mathtt{P}_i}$ and $\mathtt{P}_j$ from ID (recall that $i, j \in [n]$), and if both parties have previously successfully registered, $\mathcal{A}_t$ performs the following case distinction:

- If $j \notin \mathcal{I}$, then the message m is signcrypted into s using $\mathtt{P}_i$'s sender private key and $\mathtt{P}_j$'s receiver public key, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$.
- If $j \in \mathcal{I}$, then the further case distinction is made:
  - If $(i, j) \in \mathcal{L}$ **or** $\omega^{-1}(j) < t$, then the message m is replaced by a uniform message m^* of the same length which is signcrypted into s using $\mathtt{P}_i$'s sender private key and $\mathtt{P}_j$'s receiver public key, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$.
  - Else if $\omega^{-1}(j) > t$, then the message m is signcrypted into s using $\mathtt{P}_i$'s sender private key and $\mathtt{P}_j$'s receiver public key, and $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is output at $\mathtt{E}$.
  - Else if $\omega^{-1}(j) = t$ (that is, the party $\mathtt{P}_j$ is exactly the designated receiver $\mathtt{P}_R$) (and $i \notin \mathcal{I}$), then the message m is signcrypted into s using the provided signcryption oracle, by providing as input the key-pair of the sender $\mathtt{P}_i$ and the receiver public key $pk_R^{\star}$. $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ is then output at $\mathtt{E}$.

  In any of the above cases, the mapping $((s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}) \mapsto m)$ is stored into a table M for later reference.

On input $(\mathtt{inject}, s, \mathsf{ID}, \mathsf{ID}')$ **at interface** $\mathtt{E}$**:** The reduction $\mathcal{A}_t$ retrieves $\mathsf{ID}_{\mathtt{P}_i}$ from ID and $\mathtt{P}_j$ from ID' (recall that $i, j \in [n]$), and if both parties have previously successfully registered, $\mathcal{A}_t$ performs the following case distinction:

- If $i \in \mathcal{I}$ then, if possible, the corresponding value m is retrieved from table M and (m, ID) is output at $\mathtt{P}_j$. In case no mapping exists, no output is produced for interface $\mathtt{P}_j$.
- If $i \notin \mathcal{I}$, then the further case distinction is made:
  - If $\omega^{-1}(j) < t$ then if possible the corresponding value m is retrieved from table M, otherwise the signcryptext s is unsigncrypted into m using $\mathtt{P}_j$'s receiver private key and $\mathtt{P}_i$'s sender public key. If a value $m \neq \perp$ can be obtained this way, (m, ID) is output at $\mathtt{P}_j$ and otherwise, no output is produced.
  - If $\omega^{-1}(j) > t$ and $j \notin \mathcal{I}$, then the signcryptext s is unsigncrypted into m using $\mathtt{P}_j$'s receiver private key and $\mathtt{P}_i$'s sender public key, and (m, ID) is output at $\mathtt{P}_j$.
  - If $\omega^{-1}(j) = t$ (that is, the party $\mathtt{P}_i$ is exactly the designated sender $\mathtt{P}_S$) and $j \notin \mathcal{I}$, then if possible the corresponding value m is retrieved from table M, otherwise the signcryptext s is unsigncrypted into m using the provided unsigncryption oracle. If a value $m \neq \perp$ can be obtained this way, (m, ID) is output at $\mathtt{P}_j$ and otherwise, no output is produced.

Towards a standard hybrid argument, note that:

- $\Pr\left[\mathcal{A}_1^{\mathbf{Real}_{\Psi}^{\mathsf{MIS\text{-}Conf}}} = 1\right] = \Pr\left[\mathbf{DH}_2^{\mathcal{Z}} = 1\right]$, that is, if the reduction adversary is connected to real oracles, and it sets R as the first index in $\mathcal{I}$ according to the ordering induced by ω, then for the distinguisher $\mathbf{D}$ the view is the same as if it was connected to the real world resource $\mathbf{H}_2^{\mathcal{Z}}$, since for all parties with index $j \in \mathcal{I}$ and greater t as well as $\mathtt{P}_t$ confidentiality is only enforced for messages originating form an honest sender. Note that injecting messages in the name of corrupted senders is possible (as in a typical CCA-style game).
- $\Pr\left[\mathcal{A}_\ell^{\mathbf{Ideal}_{\Psi}^{\mathsf{MIS\text{-}Conf}}} = 1\right] = \Pr\left[\mathbf{DH}_3^{\mathcal{Z}} = 1\right]$, that is, if the reduction adversary is connected to ideal oracles, and it sets R as the last index in $\mathcal{I}$ according to the ordering induced by ω, then for the distinguisher $\mathbf{D}$ the view is the same as if it was connected to the ideal world resource $\mathbf{H}_3^{\mathcal{Z}}$, since now, for all uncorrupted parties with index $j \in \mathcal{I}$ and before R as well as $\mathtt{P}_R$ confidentiality of the message is enforced, even for messages originating from dishonest senders. However, injecting message in the name of corrupted users is still possible.

– $\Pr\left[\mathcal{A}_t^{\mathbf{Ideal}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] = \Pr\left[\mathcal{A}_{t+1}^{\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right]$,that is, if the reduction adversary $\mathcal{A}_t$ is connected to ideal oracles, then for the distinguisher $\mathbf{D}$ the view is the same as if it was being used by the reduction adversary $\mathcal{A}_{t+1}$ when connected to real oracles, since in the former case, the ideal oracles enforce confidentiality (by encryption a random message) and in the latter, this happens by definition of the reduction.

We can now conclude the proof using the hybrid argument:

$$\mathsf{Adv}^{\mathsf{MIS\text{-}Conf}}_{\Psi,\mathcal{A}} = \sum_{t=1}^{\ell} \mathsf{Adv}^{\mathsf{MIS\text{-}Conf}}_{\Psi,\mathcal{A}_t} \cdot \Pr[T=t] \tag{20}$$

$$= \frac{1}{\ell} \sum_{t=1}^{\ell} \left(\Pr\left[\mathcal{A}_t^{\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] - \Pr\left[\mathcal{A}_t^{\mathbf{Ideal}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] \right) \tag{21}$$

$$= \frac{1}{\ell} \sum_{t=1}^{\ell} \left(\Pr\left[\mathcal{A}_t^{\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] - \Pr\left[\mathcal{A}_{t+1}^{\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] \right) \tag{22}$$

$$= \frac{1}{\ell} \left(\Pr\left[\mathcal{A}_1^{\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] - \Pr\left[\mathcal{A}_{\ell+1}^{\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] \right) \tag{23}$$

$$= \frac{1}{\ell} \left(\Pr\left[\mathcal{A}_1^{\mathbf{Real}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] - \Pr\left[\mathcal{A}_\ell^{\mathbf{Ideal}_\Psi^{\mathsf{MIS\text{-}Conf}}} = 1\right] \right) \tag{24}$$

$$= \frac{1}{\ell} \left(\Pr\left[\mathbf{DH}_2^{\mathcal{Z}} = 1\right] - \Pr\left[\mathbf{DH}_3^{\mathcal{Z}} = 1\right] \right) \tag{25}$$

$$= \frac{1}{\ell} \cdot \Delta^{\mathbf{D}}(\mathbf{H}_2^{\mathcal{Z}}, \mathbf{H}_3^{\mathcal{Z}}), \tag{26}$$

where for (20) we used $\mathcal{A} = \mathcal{A}_T$ and the law of total probability, for (21) we used $\Pr[T=t] = \frac{1}{\ell}$ (for any $t \in [\ell]$), for (22), (24), and (25) we used the three equalities outlined above, for (23) we used the hybrid argument, and for (26) we used the definition of Δ. This proves that systems $\mathbf{H}_2^{\mathcal{Z}}$ and $\mathbf{H}_3^{\mathcal{Z}}$ are computationally indistinguishable, that is, for any distinguisher $\mathbf{D}$,

$$\Delta^{\mathbf{D}}(\mathbf{H}_2^{\mathcal{Z}}, \mathbf{H}_3^{\mathcal{Z}}) \leq n \cdot \mathsf{Adv}^{\mathsf{MIS\text{-}Conf}}_{\Psi,\rho_{\mathcal{Z}}(\mathbf{D})},$$

since $\ell \leq n$. □

C Formal Specification of the Hybrid Systems

On the following pages, we provide the formal specifications underlying our game-hopping argument.

On the notation. In the title of each box that depicts two hybrids at once, there are typically two names surrounded by solid or dashed boxes such as $\boxed{\mathbf{H}_0^{\mathcal{Z}}}$ and $\boxed{\mathbf{H}_1^{\mathcal{Z}}}$. This means that all code specifically surrounded by a dashed line is executed in $\mathbf{H}_0^{\mathcal{Z}}$, but not $\mathbf{H}_1^{\mathcal{Z}}$. Similarly, all code specifically surrounded by a solid line is executed in $\mathbf{H}_1^{\mathcal{Z}}$, but not in $\mathbf{H}_0^{\mathcal{Z}}$. All remaining code is executed in both systems.

In cases where the box represents just one hybrid system, we might draw boxes to highlight certain parts of the code. The descriptions of the hybrid systems are given on the following pages.

Resource $\mathbf{H}_0^{\mathcal{Z}}$

Initialization

$J_{ca}, J_{net}, S \leftarrow \emptyset$
$I_1^{ca}, I_2^{ca}, I_1^{net}, I_2^{net}, T, L \leftarrow$ empty tables

Interface $\mathtt{P}_i$

Input: $(\mathtt{register}, \mathsf{ID})$
  if $\mathsf{ID}_{\mathtt{P}_i} = \bot$ **then**
    $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S$
    $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R$
    if $\mathsf{ID} \notin J_{ca}$ **then**
      $J_{ca} \leftarrow J_{ca} \cup \{\mathsf{ID}\}$
      $I_1^{ca}[\mathtt{P}_i] \leftarrow \mathsf{ID}$
      $I_2^{ca}[\mathsf{ID}] \leftarrow \mathtt{P}_i$
      $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
      if $\mathsf{ID} \notin J_{net} \wedge I_1[\mathtt{P}_i] = \bot$ **then**
        $J_{net} \leftarrow J_{net} \cup \{\mathsf{ID}\}$
        $I_1^{net}[\mathtt{P}_i] \leftarrow \mathsf{ID}$
        $I_2^{net}[\mathsf{ID}] \leftarrow \mathtt{P}_i$
        $\mathsf{ID}_{\mathtt{P}_i} \leftarrow \mathsf{ID}$
        $val_{\mathtt{P}_i} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
        if $\mathtt{M}_i \notin \mathcal{Z}$ **then**
          $S \leftarrow S \cup \{\mathsf{ID}\}$
        output Success at $\mathtt{P}_i$
      else
        output Fail at $\mathtt{P}_i$
    else
      output Fail at $\mathtt{P}_i$

Input: $(\mathtt{send}, m, \mathsf{ID})$
  if $\mathsf{ID}_{\mathtt{P}_i} \neq \bot$ **then**
    Let $val_{\mathtt{P}_i} = [sk_S, pk_S, sk_R, pk_R]$
    if $T[\mathsf{ID}] \neq \bot$ **then**
      Parse $T[\mathsf{ID}]$ as (pk'_S, pk'_R)
      if $(pk'_S, pk'_R) \neq (\bot, \bot)$ **then**
        $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m)$
        $L[s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}] \leftarrow m$
        output $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ at $\mathtt{E}$

Interface $\mathtt{E}$ of Net

Input: $(\mathtt{inject}, s, \mathsf{ID}_s, \mathsf{ID}_r)$
  if $\mathsf{ID}_r \in J_{net}$ **then**
    $\mathtt{P}_i \leftarrow I_2^{net}[\mathsf{ID}_r]$
    Let $val_{\mathtt{P}_i} = [sk_S, pk_S, sk_R, pk_R]$
    if $T[\mathsf{ID}_s] \neq \bot$ **then**
      Parse $T[\mathsf{ID}_s]$ as (pk'_S, pk'_R)
      if $(pk'_S, pk'_R) \neq (\bot, \bot)$ **then**
        $m \leftarrow \mathsf{Unsigncrypt}(sk_R, pk'_S, s)$
        if $(\mathsf{ID}_s, \mathsf{ID}_r) \in S \times S$ **then**
          $\mathsf{bad}_1 \leftarrow (m \neq \bot \wedge L[s, \mathsf{ID}_s, \mathsf{ID}_r] = \bot)$
        if $(\mathsf{ID}_s, \mathsf{ID}_r) \in S \times (J_{net} \setminus S)$ **then**
          $\mathsf{bad}_2 \leftarrow (m \neq \bot \wedge L[s, \mathsf{ID}_s, \mathsf{ID}_r] = \bot)$
        if $m \neq \bot$ **then**
          output (m, ID_s) at $\mathtt{P}_i$

Interface $\mathtt{E}$ of CA

Input: $(\mathtt{register}, \mathsf{ID}, val)$
  if $\mathsf{ID} \notin J$ **then**
    $T[\mathsf{ID}] \leftarrow val$
    $J_{ca} \leftarrow J_{ca} \cup \{\mathsf{ID}\}$
    output Success at $\mathtt{E}$
  else
    output Fail at $\mathtt{E}$

Input: `fetchAll`
  output (J, T) at $\mathtt{E}$

Interface $\mathtt{M}_i \in \mathcal{Z}$

Input: `reveal`
  output $val_{\mathtt{P}_i}$ at $\mathtt{M}_i$

Interface $\mathtt{M}_i \notin \mathcal{Z}$

Input: `reveal`
  output $\bot$ at $\mathtt{M}_i$

Fig. 15. The first hybrid system represents a concise description of the real world, i.e., the behavior when the converters are attached at the respective interfaces of the assumed resources. The boxed statements already introduce some further notation that does not affect the behavior of this system.

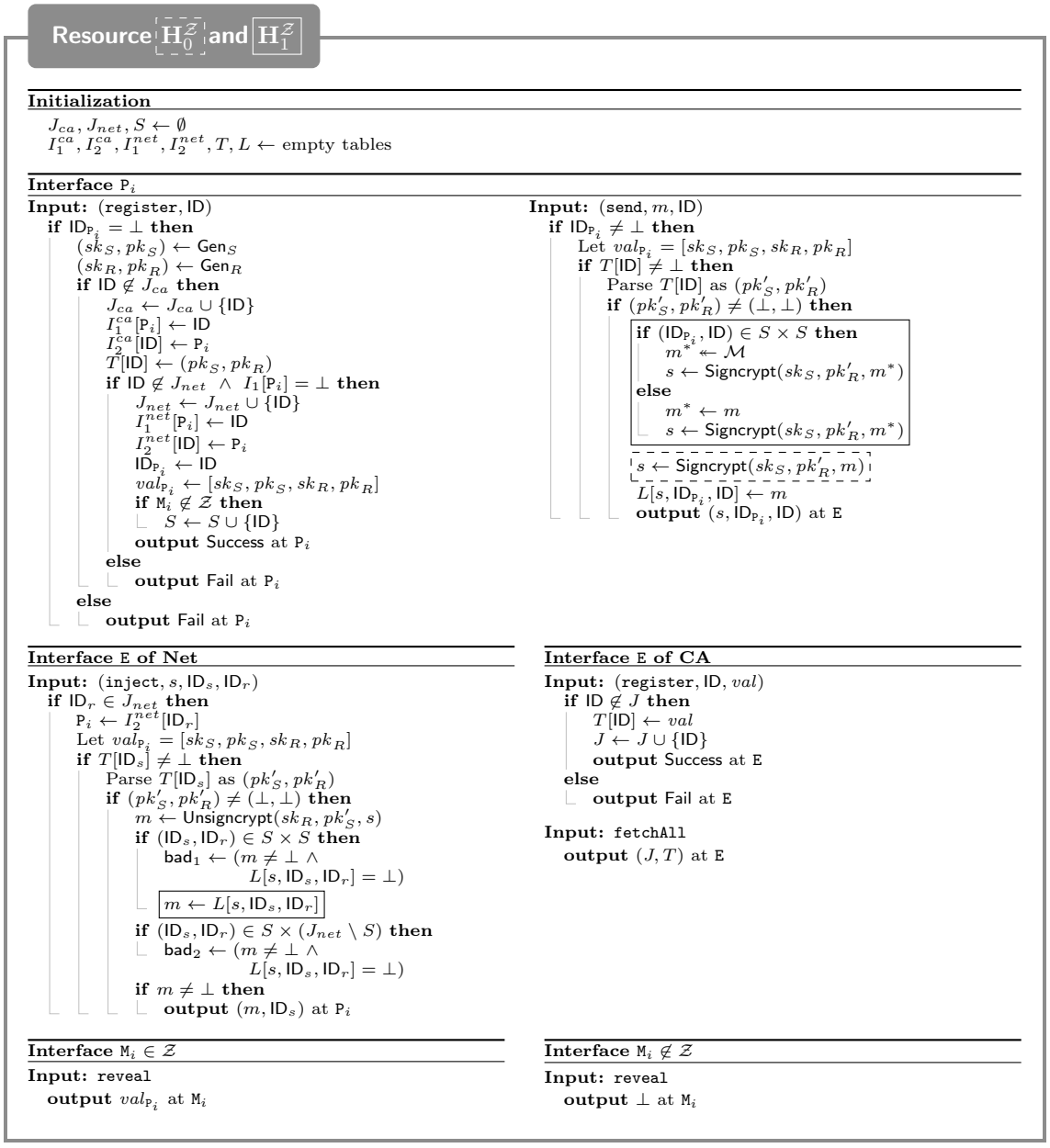

Fig. 16. The second hybrid system replaces encryptions of messages between honest parties by encryptions of uniform random messages. In addition, a message between two honest users is only delivered if it was recorded in L with the matching identities. The difference of these two hybrids can be bounded by the real-or-random game of Signcryption in the outsider security model.

Resource $\mathbf{H}_1^{\mathcal{Z}}$ and $\boxed{\mathbf{H}_2^{\mathcal{Z}}}$

Initialization

$J_{ca}, J_{net}, S \leftarrow \emptyset$
$I_1^{ca}, I_2^{ca}, I_1^{net}, I_2^{net}, T, L \leftarrow$ empty tables

Interface $\mathtt{P}_i$

Input: $(\mathtt{register}, \mathsf{ID})$
  if $\mathsf{ID}_{\mathtt{P}_i} = \bot$ **then**
    $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S$
    $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R$
    if $\mathsf{ID} \notin J_{ca}$ **then**
      $J_{ca} \leftarrow J_{ca} \cup \{\mathsf{ID}\}$
      $I_1^{ca}[\mathtt{P}_i] \leftarrow \mathsf{ID}$
      $I_2^{ca}[\mathsf{ID}] \leftarrow \mathtt{P}_i$
      $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
      if $\mathsf{ID} \notin J_{net} \wedge I_1[\mathtt{P}_i] = \bot$ **then**
        $J_{net} \leftarrow J_{net} \cup \{\mathsf{ID}\}$
        $I_1^{net}[\mathtt{P}_i] \leftarrow \mathsf{ID}$
        $I_2^{net}[\mathsf{ID}] \leftarrow \mathtt{P}_i$
        $\mathsf{ID}_{\mathtt{P}_i} \leftarrow \mathsf{ID}$
        $val_{\mathtt{P}_i} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
        if $\mathtt{M}_i \notin \mathcal{Z}$ **then**
          $S \leftarrow S \cup \{\mathsf{ID}\}$
        output Success at $\mathtt{P}_i$
      else
        output Fail at $\mathtt{P}_i$
    else
      output Fail at $\mathtt{P}_i$

Input: $(\mathtt{send}, m, \mathsf{ID})$
  if $\mathsf{ID}_{\mathtt{P}_i} \neq \bot$ **then**
    Let $val_{\mathtt{P}_i} = [sk_S, pk_S, sk_R, pk_R]$
    if $T[\mathsf{ID}] \neq \bot$ **then**
      Parse $T[\mathsf{ID}]$ as (pk'_S, pk'_R)
      if $(pk'_S, pk'_R) \neq (\bot, \bot)$ **then**
        if $(\mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}) \in S \times S$ **then**
          $m^* \twoheadleftarrow \mathcal{M}$
          $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m^*)$
        else
          $m^* \leftarrow m$
          $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m^*)$
        $L[s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}] \leftarrow m$
        output $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ at $\mathtt{E}$

Interface $\mathtt{E}$ of Net

Input: $(\mathtt{inject}, s, \mathsf{ID}_s, \mathsf{ID}_r)$
  if $\mathsf{ID}_r \in J_{net}$ **then**
    $\mathtt{P}_i \leftarrow I_2^{net}[\mathsf{ID}_r]$
    Let $val_{\mathtt{P}_i} = [sk_S, pk_S, sk_R, pk_R]$
    if $T[\mathsf{ID}_s] \neq \bot$ **then**
      Parse $T[\mathsf{ID}_s]$ as (pk'_S, pk'_R)
      if $(pk'_S, pk'_R) \neq (\bot, \bot)$ **then**
        $m \leftarrow \mathsf{Unsigncrypt}(sk_R, pk'_S, s)$
        if $(\mathsf{ID}_s, \mathsf{ID}_r) \in S \times S$ **then**
          $\mathsf{bad}_1 \leftarrow (m \neq \bot \wedge L[s, \mathsf{ID}_s, \mathsf{ID}_r] = \bot)$
          $m \leftarrow L[s, \mathsf{ID}_s, \mathsf{ID}_r]$
        if $(\mathsf{ID}_s, \mathsf{ID}_r) \in S \times (J_{net} \setminus S)$ **then**
          $\mathsf{bad}_2 \leftarrow (m \neq \bot \wedge L[s, \mathsf{ID}_s, \mathsf{ID}_r] = \bot)$
          $\boxed{m \leftarrow L[s, \mathsf{ID}_s, \mathsf{ID}_r]}$
        if $m \neq \bot$ **then**
          output (m, ID_s) at $\mathtt{P}_i$

Interface $\mathtt{E}$ of CA

Input: $(\mathtt{register}, \mathsf{ID}, val)$
  if $\mathsf{ID} \notin J$ **then**
    $T[\mathsf{ID}] \leftarrow val$
    $J \leftarrow J \cup \{\mathsf{ID}\}$
    output Success at $\mathtt{E}$
  else
    output Fail at $\mathtt{E}$

Input: `fetchAll`
  output (J, T) at $\mathtt{E}$

Interface $\mathtt{M}_i \in \mathcal{Z}$

Input: `reveal`
  output $val_{\mathtt{P}_i}$ at $\mathtt{M}_i$

Interface $\mathtt{M}_i \notin \mathcal{Z}$

Input: `reveal`
  output $\bot$ at $\mathtt{M}_i$

Fig. 17. The third hybrid system enforces that no message of an honest sender can be forged. This intuitively follows from insider security of signcryption.

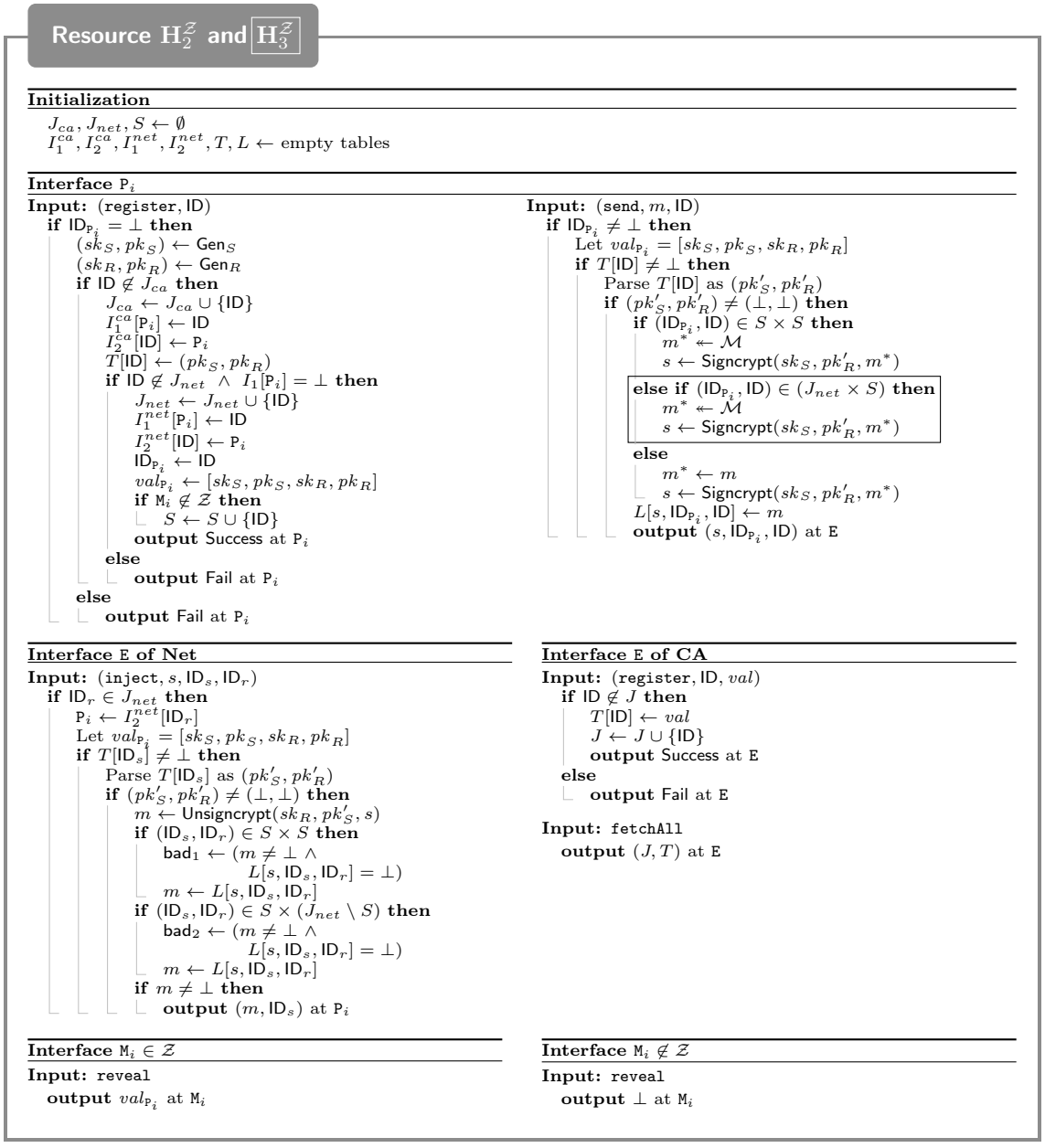

Fig. 18. The fourth hybrid ensures that no information about a message to an honest receiver is leaked to the adversary. This follows intuitively from the insider security model of signcryption.

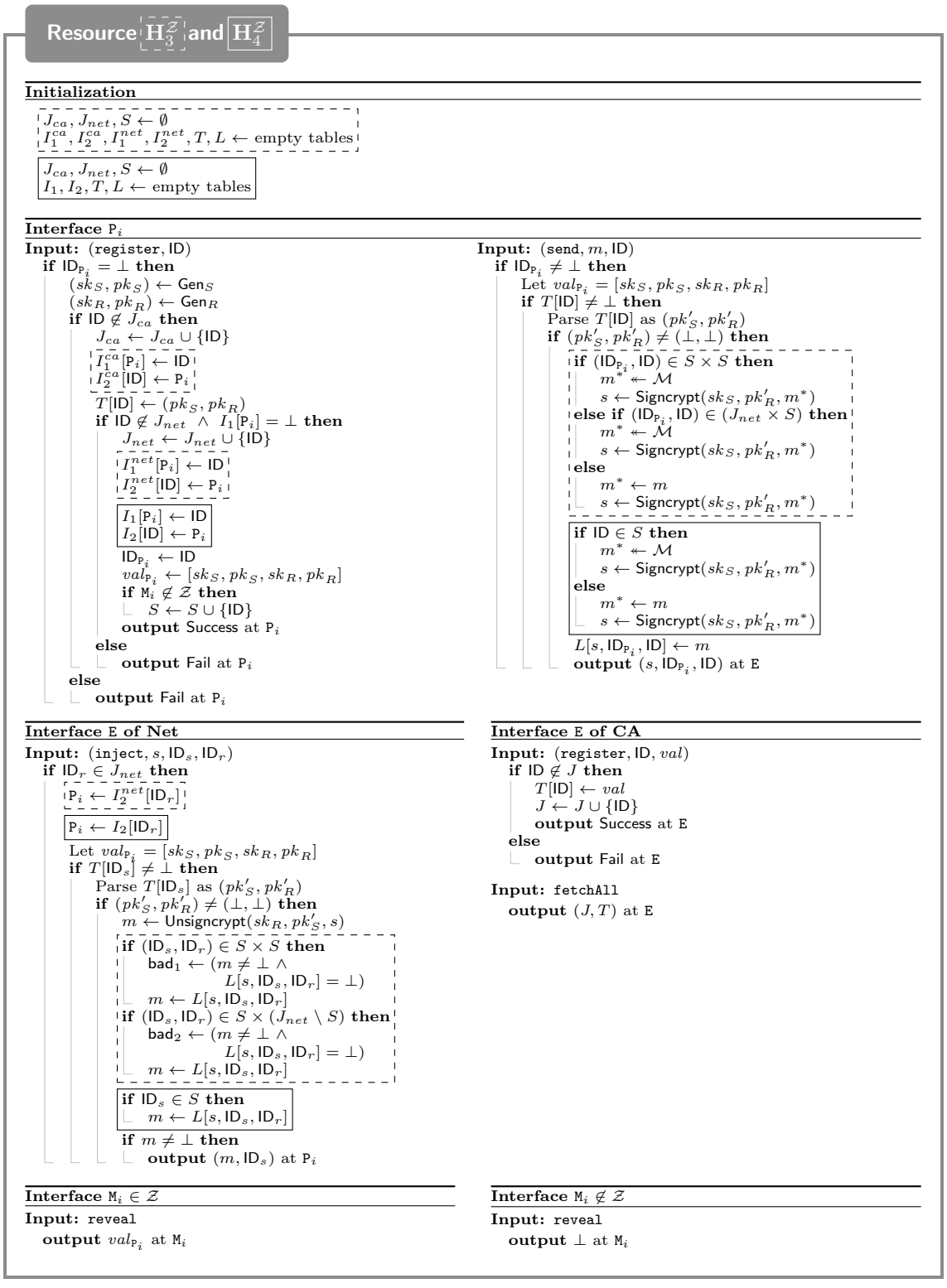

Fig. 19. The fifth hybrid system is equivalent to the fourth. Only syntactic manipulations are made that do not affect the behavior.

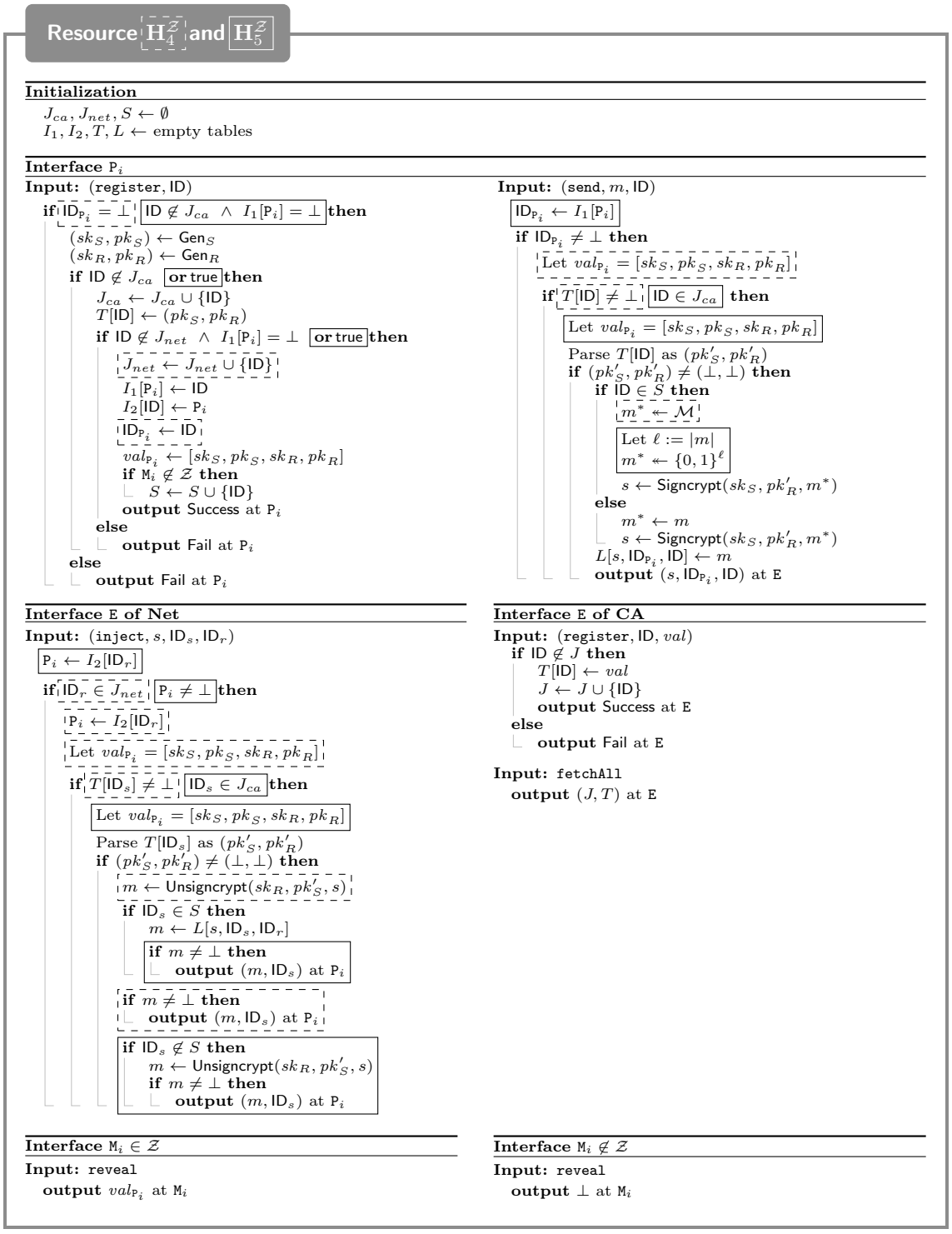

Fig. 20. The sixth hybrid system is equivalent to the fifth.

Resource $\mathbf{H}_5^{\mathcal{Z}}$ and $\mathbf{H}_6^{\mathcal{Z}}$

Initialization

J_{ca} J J_{net} $S \leftarrow \emptyset$ $j \leftarrow 0$

I_1, I_2, T, L $L_{sim}, L_{res} \leftarrow$ empty tables

$(r_1^1, r_1^2)||(r_2^1, r_2^2)||\ldots||(r_n^1, r_n^2) \twoheadleftarrow \{0,1\}^{n\cdot(2\kappa)}$ ▷ Common randomness

Interface $\mathtt{P}_i$

Input: $(\mathtt{register}, \mathsf{ID})$
- **if** $\mathsf{ID} \notin J_{ca}$ $J \wedge\ I_1[\mathtt{P}_i] = \perp$ **then**
  - $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S$
  - $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R$
  - J_{ca} $J \leftarrow J_{ca}$ J $\cup \{\mathsf{ID}\}$
  - $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
  - $I_1[\mathtt{P}_i] \leftarrow \mathsf{ID}$
  - $I_2[\mathsf{ID}] \leftarrow \mathtt{P}_i$
  - $val_{\mathtt{P}_i} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
  - **if** $\mathtt{M}_i \notin \mathcal{Z}$ **then**
    - $S \leftarrow S \cup \{\mathsf{ID}\}$
  - **output** Success at $\mathtt{P}_i$
- **else**
  - **output** Fail at $\mathtt{P}_i$

Input: $(\mathtt{send}, m, \mathsf{ID})$
- $\mathsf{ID}_{\mathtt{P}_i} \leftarrow I_1[\mathtt{P}_i]$
- **if** $\mathsf{ID}_{\mathtt{P}_i} \neq \perp$ **then**
  - **if** $\mathsf{ID} \in J_{ca}$ J **then**
    - **for** each $\mathsf{ID} \in J$ **do**
      - **if** $I_2[\mathsf{ID}] \neq \perp\ \wedge\ T[\mathsf{ID}] = \perp$ **then**
        - Let $\mathtt{P}_k \leftarrow I_2[\mathsf{ID}]$
        - **if** $\mathtt{M}_k \notin \mathcal{Z}$ **then**
          - $(r_k^1, r_k^2) \twoheadleftarrow \{0,1\}^{2\kappa}$
        - $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_k^1)$
        - $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_k^2)$
        - $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
        - $val_{\mathtt{P}_k} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
    - Parse $val_{\mathtt{P}_i}$ as $[sk_S, pk_S, sk_R, pk_R]$
    - Parse $T[\mathsf{ID}]$ as (pk'_S, pk'_R)
    - **if** $(pk'_S, pk'_R) \neq (\perp, \perp)$ **then**
      - **if** $\mathsf{ID} \in S$ **then**
        - Let $\ell := |m|$
        - $m^* \twoheadleftarrow \{0,1\}^\ell$
        - $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m^*)$
      - **else**
        - $m^* \leftarrow m$
        - $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m^*)$
      - $L[s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}] \leftarrow m$
      - $L_{sim}[c, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}] \leftarrow j$
      - $L_{res}[j] \leftarrow (m, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$
      - $j \leftarrow j + 1$
      - **output** $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ at $\mathtt{E}$

Interface $\mathtt{E}$ of Net

Input: $(\mathtt{inject}, s, \mathsf{ID}_s, \mathsf{ID}_r)$
- **for** each $\mathsf{ID} \in J$ **do**
  - **if** $I_2[\mathsf{ID}] \neq \perp\ \wedge\ T[\mathsf{ID}] = \perp$ **then**
    - Let $\mathtt{P}_k \leftarrow I_2[\mathsf{ID}]$
    - **if** $\mathtt{M}_k \notin \mathcal{Z}$ **then** $(r_k^1, r_k^2) \twoheadleftarrow \{0,1\}^{2\kappa}$
    - $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_k^1)$
    - $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_k^2)$
    - $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
    - $val_{\mathtt{P}_k} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
- $\mathtt{P}_i \leftarrow I_2[\mathsf{ID}_r]$
- **if** $\mathtt{P}_i \neq \perp$ **then**
  - **if** $\mathsf{ID}_s \in J_{ca}$ J **then**
    - Let $val_{\mathtt{P}_i} = [sk_S, pk_S, sk_R, pk_R]$
    - Parse $T[\mathsf{ID}_s]$ as (pk'_S, pk'_R)
    - **if** $(pk'_S, pk'_R) \neq (\perp, \perp)$ **then**
      - **if** $\mathsf{ID}_s \in S$ **then**
        - $m \leftarrow L[s, \mathsf{ID}_s, \mathsf{ID}_r]$
        - **if** $m \neq \perp$ **then**
          - **output** (m, ID_s) at $\mathtt{P}_i$
        - **if** $\exists j : L_{sim}[c, \mathsf{ID}_s, \mathsf{ID}_r] = j\}$ **then**
          - $j \leftarrow L_{sim}[c, \mathsf{ID}_s, \mathsf{ID}_r]$
          - Parse $L_{res}[j]$ as $(m, \mathsf{ID}_s, \mathsf{ID}_r)$
          - **output** (m, ID_s) at $\mathtt{P}_i$
      - **if** $\mathsf{ID}_s \notin S$ **then**
        - $m \leftarrow \mathsf{Unsigncrypt}(sk_R, pk'_S, s)$
        - **if** $m \neq \perp$ **then**
          - **output** (m, ID_s) at $\mathtt{P}_i$

Interface $\mathtt{E}$ of CA

Input: $(\mathtt{register}, \mathsf{ID}, val)$
- **if** $\mathsf{ID} \notin J$ **then**
  - $T[\mathsf{ID}] \leftarrow val$
  - J_{ca} $J \leftarrow J_{ca}$ $J \cup \{\mathsf{ID}\}$
  - **output** Success at $\mathtt{E}$
- **else**
  - **output** Fail at $\mathtt{E}$

Input: `fetchAll`
- **for** each $\mathsf{ID} \in J$ **do**
  - **if** $I_2[\mathsf{ID}] \neq \perp\ \wedge\ T[\mathsf{ID}] = \perp$ **then**
    - Let $\mathtt{P}_k \leftarrow I_2[\mathsf{ID}]$
    - **if** $\mathtt{M}_k \notin \mathcal{Z}$ **then**
      - $(r_k^1, r_k^2) \twoheadleftarrow \{0,1\}^{2\kappa}$
    - $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_k^1)$
    - $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_k^2)$
    - $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
    - $val_{\mathtt{P}_k} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
- **output** (J, T) at $\mathtt{E}$

Interface $\mathtt{M}_i \in \mathcal{Z}$

Input: `reveal`
- **if** $I_1[\mathtt{P}_i] \neq \perp$ **and** $val_{\mathtt{P}_i} = \perp$ **then**
  - $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_i^1)$
  - $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_i^2)$
  - $val_{\mathtt{P}_i} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
- **output** $val_{\mathtt{P}_i}$ at $\mathtt{M}_i$

Interface $\mathtt{M}_i \notin \mathcal{Z}$

Input: `reveal`
- **output** $\perp$ at $\mathtt{M}_i$

Fig. 21. The seventh hybrid system is equivalent to the sixth.

Resource $\mathbf{H}_6^{\mathcal{Z}}$

Initialization

$J, S \leftarrow \emptyset$ $j \leftarrow 0$

I_1, I_2, L_{res}, $T, L_{sim} \leftarrow$ empty tables

$(r_1^1, r_1^2)||(r_2^1, r_2^2)||\ldots||(r_n^1, r_n^2) \twoheadleftarrow \{0,1\}^{n\cdot(2\kappa)}$ ▷ Common randomness

Interface $\mathtt{P}_i$

Input: $(\mathtt{register}, \mathsf{ID})$

if $\mathsf{ID} \notin J \ \wedge \ I_1[\mathtt{P}_i] = \bot$ **then**
  $J \leftarrow J \cup \{\mathsf{ID}\}$
  $I_1[\mathtt{P}_i] \leftarrow \mathsf{ID}$
  $I_2[\mathsf{ID}] \leftarrow \mathtt{P}_i$
  if $\mathtt{M}_i \notin \mathcal{Z}$ **then**
    $S \leftarrow S \cup \{\mathsf{ID}\}$
  output Success at $\mathtt{P}_i$
else
  output Fail at $\mathtt{P}_i$

Input: $(\mathtt{send}, m, \mathsf{ID})$

$\mathsf{ID}_{\mathtt{P}_i} \leftarrow I_1[\mathtt{P}_i]$
if $\mathsf{ID}_{\mathtt{P}_i} \neq \bot$ **then**
  if $\mathsf{ID} \in J$ **then**
    for each $\mathsf{ID} \in J$ **do**
      if $I_2[\mathsf{ID}] \neq \bot \ \wedge \ T[\mathsf{ID}] = \bot$ **then**
        Let $\mathtt{P}_k \leftarrow I_2[\mathsf{ID}]$
        if $\mathtt{M}_k \notin \mathcal{Z}$ **then**
          $(r_k^1, r_k^2) \twoheadleftarrow \{0,1\}^{2\kappa}$
        $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_k^1)$
        $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_k^2)$
        $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
        $val_{\mathtt{P}_k} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
    Parse $val_{\mathtt{P}_i}$ as $[sk_S, pk_S, sk_R, pk_R]$
    Parse $T[\mathsf{ID}]$ as (pk'_S, pk'_R)
    if $(pk'_S, pk'_R) \neq (\bot, \bot)$ **then**
      if $\mathsf{ID} \in S$ **then**
        Let $\ell := |m|$
        $m^* \twoheadleftarrow \{0,1\}^\ell$
        $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m^*)$
      else
        $m^* \leftarrow m$
        $s \leftarrow \mathsf{Signcrypt}(sk_S, pk'_R, m^*)$
      $L_{sim}[c, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID}] \leftarrow j$
      $L_{res}[j] \leftarrow (m, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$
      $j \leftarrow j + 1$
      output $(s, \mathsf{ID}_{\mathtt{P}_i}, \mathsf{ID})$ at $\mathtt{E}$

Interface $\mathtt{E}$ of Net

Input: $(\mathtt{inject}, s, \mathsf{ID}_s, \mathsf{ID}_r)$

for each $\mathsf{ID} \in J$ **do**
  if $I_2[\mathsf{ID}] \neq \bot \ \wedge \ T[\mathsf{ID}] = \bot$ **then**
    Let $\mathtt{P}_k \leftarrow I_2[\mathsf{ID}]$
    if $\mathtt{M}_k \notin \mathcal{Z}$ **then** $(r_k^1, r_k^2) \twoheadleftarrow \{0,1\}^{2\kappa}$
    $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_k^1)$
    $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_k^2)$
    $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
    $val_{\mathtt{P}_k} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
$\mathtt{P}_i \leftarrow I_2[\mathsf{ID}_r]$
if $\mathtt{P}_i \neq \bot$ **then**
  if $\mathsf{ID}_s \in J$ **then**
    Let $val_{\mathtt{P}_i} = [sk_S, pk_S, sk_R, pk_R]$
    Parse $T[\mathsf{ID}_s]$ as (pk'_S, pk'_R)
    if $(pk'_S, pk'_R) \neq (\bot, \bot)$ **then**
      if $\mathsf{ID}_s \in S$ **then**
        if $\exists j : L_{sim}[c, \mathsf{ID}_s, \mathsf{ID}_r] = j\}$ **then**
          $j \leftarrow L_{sim}[c, \mathsf{ID}_s, \mathsf{ID}_r]$
          Parse $L_{res}[j]$ as $(m, \mathsf{ID}_s, \mathsf{ID}_r)$
          output (m, ID_s) at $\mathtt{P}_i$
      if $\mathsf{ID}_s \notin S$ **then**
        $m \leftarrow \mathsf{Unsigncrypt}(sk_R, pk'_S, s)$
        if $m \neq \bot$ **then**
          output (m, ID_s) at $\mathtt{P}_i$

Interface $\mathtt{E}$ of CA

Input: $(\mathtt{register}, \mathsf{ID}, val)$

if $\mathsf{ID} \notin J$ **then**
  $T[\mathsf{ID}] \leftarrow val$
  $J \leftarrow J \cup \{\mathsf{ID}\}$
  output Success at $\mathtt{E}$
else
  output Fail at $\mathtt{E}$

Input: `fetchAll`

for each $\mathsf{ID} \in J$ **do**
  if $I_2[\mathsf{ID}] \neq \bot \ \wedge \ T[\mathsf{ID}] = \bot$ **then**
    Let $\mathtt{P}_k \leftarrow I_2[\mathsf{ID}]$
    if $\mathtt{M}_k \notin \mathcal{Z}$ **then** $(r_k^1, r_k^2) \twoheadleftarrow \{0,1\}^{2\kappa}$
    $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_k^1)$
    $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_k^2)$
    $T[\mathsf{ID}] \leftarrow (pk_S, pk_R)$
    $val_{\mathtt{P}_k} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
output (J, T) at $\mathtt{E}$

Interface $\mathtt{M}_i \in \mathcal{Z}$

Input: `reveal`

if $I_1[\mathtt{P}_i] \neq \bot$ **and** $val_{\mathtt{P}_i} = \bot$ **then**
  $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S(r_i^1)$
  $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R(r_i^2)$
  $val_{\mathtt{P}_i} \leftarrow [sk_S, pk_S, sk_R, pk_R]$
output $val_{\mathtt{P}_i}$ at $\mathtt{M}_i$

Interface $\mathtt{M}_i \notin \mathcal{Z}$

Input: `reveal`

output $\bot$ at $\mathtt{M}_i$

Fig. 22. Illustration that $\mathbf{H}_6^{\mathcal{Z}}$ is equivalent to the ideal world. Dashed/Green: code of secure network; Solid/Blue: code of simulators.