A Constructive Perspective on Signcryption Security

Christian Badertscher(✉), Fabio Banfi, and Ueli Maurer

Department of Computer Science, ETH Zurich, 8092 Zürich, Switzerland
{badi,fbanfi,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.

D. Catalano and R. De Prisco (Eds.): SCN 2018, LNCS 11035, pp. 102–120, 2018.
https://doi.org/10.1007/978-3-319-98113-0_6

1 Introduction

1.1 Motivation and Background

Signcryption is a public-key cryptographic primitive introduced by Zheng [35] 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.

Since its introduction, several concrete schemes have emerged in the literature based on different hardness assumptions [20,21,31,35,36]. Also, new properties beyond the basic security goals have been introduced recently, such as identity-based [8,20,22,23,29,30], hybrid [13], KEM-DEM-based [7], certificateless [5], verifiable [29], attribute-based [11,27], functional [12], or key invisible [33] 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 [4]. While (symmetric) authenticated encryption was put on solid security definitions directly from the start (cf. [6]), the basic security notions for signcryption have had a more difficult path and converged to a set of commonly agreed notions only recently [34] and only thanks to the merits of a sequence of foundational works [1,2,4] that formally introduced what is now known as the *outsider security model*—the model that captures network attackers or an adversarial entity that registers public keys with a certificate authority—and the *insider security model*—the model that captures attacks of corrupted users, for example an a priori legitimate user whose private key got compromised.

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 [16] 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 [9].

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 [4,16,32], recent developments in practical lattice-based schemes [17], and one of the main references on the basic notions [34, p. 29 and 46], that assign too little credit to the relevance of insider security. In this paper, we take a step towards clarifying this situation by systematically identifying which basic notion a signcryption scheme should fulfill and why. We believe that our analysis provides sufficient evidence to call insider security the standard notion for signcryption and to pinpoint which proposed variants of insider security are practically relevant. 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.

Our main technical result is the proof of the following theorem.

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 Definitions 3, 4 and 5, 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 Definition 3, then the secure network is constructed if no key of legitimate users is compromised.

1.3 Contributions

The Preferred Insider Security Notion. Our analysis identifies the notions that imply the above construction and thereby provides confidence that the security games that we formally describe in Figs. 2 and 3 in Sect. 3 are an adequate choice to model game-based insider security. The notions we use are in particular implied by what is denoted in [34] 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.

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.

Enabling Comparisons 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 [14], 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 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. The schemes are in addition arguably practically efficient [10]. 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 [15] and authenticated encryption [18] 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 key compromises.

Modeling Partial Corruptions. Our composable security analysis considers the so-called *static corruption* model which is the typical model when analyzing communication protocols that involve standard encryption techniques. A discussion of adaptive corruptions and forward secrecy is found in the full version [3]. Since in our setting the only secret information of a party is its secret key, compromising the key fully corrupts a party as it allows the attacker to entirely impersonate and control the party (sending, reading, and delivering messages).

Our approach thereby introduces a conceptual contribution: we make *partial corruptions* explicit in the model and we refrain from letting compromised parties be formally absorbed by the adversary (i.e., partially corrupted parties are still operational as protocol machines). Still, as explained above, our statements contain the *full corruption* case. We believe that identifying reasonable partial corruption scenarios seems to be crucial in building formal models that are able to capture *a range* of real-world threats and to precisely express which security guarantees can still be retained in the presence of such threats.

2 Preliminaries

2.1 Notation for Systems and Games

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., $(\texttt{send}, m, \mathsf{ID}_r)$ to send a message m to a receiver with identity ID_r). We ignore keywords in writing the domains of arguments, e.g., $(\texttt{send}, m, \mathsf{ID}_r) \in \mathcal{M} \times \{0,1\}^*$ 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 known) integer $k > 0$, and we do not write the security parameter as an explicit input to functions and algorithms.

2.2 Definition of Signcryption Schemes

We present the formal syntactic definition of Signcryption from [4]. For convenience, we do not make domain parameters and their generation explicit in our notation.

Definition 1 (Signcryption Scheme). *A signcryption scheme* $\Psi = (\mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Signcrypt}, \mathsf{Unsigncrypt})$ *for* key space $\mathcal{K}$*,* 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 scheme is correct *if 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* $m \in \mathcal{M}$ *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 [24], 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.

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* `in` and the outer interface `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 $\overline{\mathcal{P}} := \mathcal{I} \backslash \mathcal{P}$. 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. (The composition of resources with different interface sets arises naturally by introducing dummy interfaces.)

In this paper, we make statements about resources with interfaces from the set $\mathcal{I} = \{\mathtt{P}_1, \ldots, \mathtt{P}_n, \mathtt{M}_1, \ldots, \mathtt{M}_n, \mathtt{E}\}$. Interface $\mathtt{P}_i$ can be thought of as being the access point of the ith honest party to the system. Interface $\mathtt{M}_i$ is the access point of an intruder (i.e., a hypothetical attacker entity like Mallory), and $\mathtt{E}$ 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 $\mathtt{E}$ 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 [26] for more details and briefly mention that this concept has turned out to be useful in modeling cryptographic problems [19].

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 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.[1]

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

[1] 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 ideal functionality from the given ones. We directly state the central definition of a construction of [26] 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 $\mathcal{U} \subseteq \mathcal{I}$ 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*

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

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 for any distinguisher $\mathbf{D}$

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

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 [25,26] for a proof. For readers more familiar with Canetti's UC Framework [9], we refer to [19] for explanations of how the above concepts relate to similar concepts in UC. We refer to Fig. 4 (in Sect. 4.2) 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 $\rho(\cdot)$ stands for an efficient black-box construction of such an adversary $\mathcal{A}$ from a distinguisher $\mathbf{D}$.

3 An Overview of Signcyrption Security

Our analysis of signcryption focuses on the multi-user model extensively studied by Baek, Steinfield, and Zheng in [4]. We now present the relevant security games.

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*

```
$\mathbf{Real}_{\Psi}^{\mathsf{MOS}}$

Initialization
    $(sk_S^\star, pk_S^\star) \leftarrow \mathsf{Gen}_S$
    $(sk_R^\star, pk_R^\star) \leftarrow \mathsf{Gen}_R$
    $S \leftarrow \emptyset$
    return $(pk_S^\star, pk_R^\star)$

Oracle Scr
Input: $(pk_R, m) \in \mathcal{K} \times \mathcal{M}$
    $s \leftarrow \mathsf{Signcrypt}(sk_S^\star, pk_R, m)$
    if $pk_R = pk_R^\star$ then
        $S \leftarrow S \cup \{s\}$
    return $s$

Oracle Usc
Input: $(pk_S, s) \in \mathcal{K} \times \mathcal{S}$
    if $pk_S = pk_S^\star \wedge s \in S$ then
        $m \leftarrow \perp$
    else
        $m \leftarrow \mathsf{Unsigncrypt}(sk_R^\star, pk_S, s)$
    return $m$
```

```
$\mathbf{Ideal}_{\Psi}^{\mathsf{MOS}}$

Initialization
    $(sk_S^\star, pk_S^\star) \leftarrow \mathsf{Gen}_S$
    $(sk_R^\star, pk_R^\star) \leftarrow \mathsf{Gen}_R$
    return $(pk_S^\star, pk_R^\star)$

Oracle Scr
Input: $(pk_R, m) \in \mathcal{K} \times \mathcal{M}$
    if $pk_R = pk_R^\star$ then
        $m \twoheadleftarrow \mathcal{M}$
    $s \leftarrow \mathsf{Signcrypt}(sk_S^\star, pk_R, m)$
    return $s$

Oracle Usc
Input: $(pk_S, s) \in \mathcal{K} \times \mathcal{S}$
    if $pk_S = pk_S^\star$ then
        $m \leftarrow \perp$
    else
        $m \leftarrow \mathsf{Unsigncrypt}(sk_R^\star, pk_S, s)$
    return $m$
```

Fig. 1. The games $\mathbf{Real}_{\Psi}^{\mathsf{MOS}}$ and $\mathbf{Ideal}_{\Psi}^{\mathsf{MOS}}$.

of signcryptexts under a chosen-signcryptext attack by an outsider adversary (MOS-Conf) and *strong unforgeability of signcryptexts* (also called *integrity of signcryptexts*) *under a chosen-message attack by an outsider adversary* (MOS-Auth). 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 [28]. The all-in-one version is equivalent to the combination of the two mentioned separate security notions which is proven in the full version [3]. 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 $\mathbf{G}$.

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 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 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* $\mathcal{A}$*.*

3.2 Multi-user Insider Security

For insider security, the two basic requirements are *indistinguishability of signcryptexts under a chosen-signcryptext attack by an insider adversary* (MIS-Conf) and *strong unforgeability of signcryptexts* (also called *integrity of signcryptexts*) *under a chosen-message attack by an insider adversary* (MIS-Auth).

Confidentiality. The games capturing MIS-Conf (using the real-or-random paradigm) are given in Fig. 2. We specify two variants of different strengths: the games that include the **Gen** oracle and the boxed statements constitute the weaker version which we use in this work. Intuitively, the weaker game does not allow the adversary to choose the randomness to generate keys. However, in both variants whenever the adversary makes an oracle call, he has to provide a *valid* key-pair. As commonly known, enforcing this 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. We now state the formal definition:

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. The forgery game $\mathbf{Auth}_\Psi^{\mathsf{MIS}}$ is given in Fig. 3. We again give two variants as for confidentiality before. We directly state the relevant definition:

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).*

4 Constructive Analysis

4.1 Real World: Assumed Resources and Converters

We now describe the assumed resources and the converters. The formal specifications as pseudo-code are given in the full version [3].

```
$\mathbf{Real}_{\Psi}^{\mathsf{MIS\text{-}Conf}}$

Initialization
    $(sk_R^\star, pk_R^\star) \leftarrow \mathsf{Gen}_R$
    $S \leftarrow \emptyset$
    $\boxed{K \leftarrow \emptyset}$
    return $pk_R^\star$

Oracle Gen
    $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S$
    $K \leftarrow K \cup \{(sk_S, pk_S)\}$
    return $(sk_S, pk_S)$

Oracle Scr
Input: $((sk_S, pk_S), pk_R, m) \in$
       $\mathrm{supp}(\mathsf{Gen}_S) \times \mathcal{K} \times \mathcal{M}$
    $\boxed{\text{if } (sk_S, pk_S) \notin K \text{ then}}$
    $\boxed{\quad \text{return } \bot}$
    $s \leftarrow \mathsf{Signcrypt}(sk_S, pk_R, m)$
    if $pk_R = pk_R^\star$ then
        $S \leftarrow S \cup \{(pk_S, s)\}$
    return $s$

Oracle Usc
Input: $(pk_S, s) \in \mathcal{K} \times \mathcal{S}$
    if $(pk_S, s) \in S$ then
        $m \leftarrow \bot$
    else
        $m \leftarrow \mathsf{Unsigncrypt}(sk_R^\star, pk_S, s)$
    return $m$
```

```
$\mathbf{Ideal}_{\Psi}^{\mathsf{MIS\text{-}Conf}}$

Initialization
    $(sk_R^\star, pk_R^\star) \leftarrow \mathsf{Gen}_R$
    $S \leftarrow \emptyset$
    $\boxed{K \leftarrow \emptyset}$
    return $pk_R^\star$

Oracle Gen
    $(sk_S, pk_S) \leftarrow \mathsf{Gen}_S$
    $K \leftarrow K \cup \{(sk_S, pk_S)\}$
    return $(sk_S, pk_S)$

Oracle Scr
Input: $((sk_S, pk_S), pk_R, m) \in$
       $\mathrm{supp}(\mathsf{Gen}_S) \times \mathcal{K} \times \mathcal{M}$
    $\boxed{\text{if } (sk_S, pk_S) \notin K \text{ then}}$
    $\boxed{\quad \text{return } \bot}$
    if $pk_R = pk_R^\star$ then
        $m \twoheadleftarrow \mathcal{M}$
    $s \leftarrow \mathsf{Signcrypt}(sk_S, pk_R, m)$
    if $pk_R = pk_R^\star$ then
        $S \leftarrow S \cup \{(pk_S, s)\}$
    return $s$

Oracle Usc
Input: $(pk_S, s) \in \mathcal{K} \times \mathcal{S}$
    if $(pk_S, s) \in S$ then
        $m \leftarrow \bot$
    else
        $m \leftarrow \mathsf{Unsigncrypt}(sk_R^\star, pk_S, s)$
    return $m$
```

Fig. 2. The games $\mathbf{Real}_{\Psi}^{\mathsf{MIS\text{-}Conf}}$ and $\mathbf{Ideal}_{\Psi}^{\mathsf{MIS\text{-}Conf}}$. The games that additionally includes the boxed statements (and the oracle **Gen**) constitute the weaker versions.

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. 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 $\phi^{\mathrm{net}} := (\mathbf{1}, \ldots, \mathbf{1}, \mathsf{dlv})$ 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).

```
$\mathbf{Auth}_{\Psi}^{\mathsf{MIS}}$

Initialization
    $(sk_S^\star, pk_S^\star) \leftarrow \mathsf{Gen}_S$
    $S \leftarrow \emptyset$
    $\boxed{K \leftarrow \emptyset}$
    $\mathsf{win} \leftarrow 0$
    return $pk_S^\star$

Oracle Gen
    $(sk_R, pk_R) \leftarrow \mathsf{Gen}_R$
    $K \leftarrow K \cup \{(sk_R, pk_R)\}$
    return $(sk_R, pk_R)$

Oracle Scr
Input: $(pk_R, m) \in \mathcal{K} \times \mathcal{M}$
    $s \leftarrow \mathsf{Signcrypt}(sk_S^\star, pk_R, m)$
    $S \leftarrow S \cup \{(pk_R, s)\}$
    return $s$

Oracle Usc
Input: $((sk_R, pk_R), pk_S, s) \in \mathrm{supp}(\mathsf{Gen}_R) \times \mathcal{K} \times \mathcal{S}$
    $\boxed{\begin{array}{l}\textbf{if } (sk_R, pk_R) \notin K \textbf{ then}\\ \quad \textbf{return } \bot\end{array}}$
    $m \leftarrow \mathsf{Unsigncrypt}(sk_R^\star, pk_S, s)$
    if $pk_S = pk_S^\star \land m \neq \bot \land (pk_R, s) \notin S$ then
        $\mathsf{win} \leftarrow 1$
    return $m$
```

Fig. 3. The forgery game $\mathbf{Auth}_{\Psi}^{\mathsf{MIS}}$. The game that includes the boxed statements (and the oracle **Gen**) constitutes the weaker version.

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 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. 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

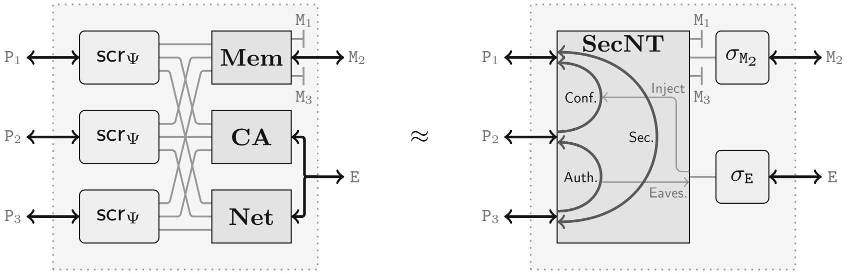

Fig. 4. Illustration of the construction notion. Left (real world): Three parties running the protocol and where the second party's key got compromised. Right (ideal world): The secure network resource (with simulators) that guarantees secure communication between $\mathtt{P}_1$ and $\mathtt{P}_3$, but for example only confidential communication from party $\mathtt{P}_2$ to party $\mathtt{P}_1$, and only authentic communication from party $\mathtt{P}_3$ to party $\mathtt{P}_2$.

$(\mathtt{fetch}, \mathsf{ID})$ 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 capabilities at interface $\mathtt{E}$ are again not guaranteed and will be filtered as in the case of the network.

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 $\mathtt{P}_i$. 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 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 **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.

4.2 Ideal World: 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. 5. This ideal network is basically a secure network. To see this, imagine there was no interface $\mathtt{M}_i$: 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 $\mathtt{P}_i$ registers its identity successfully, then its associated identity is only added to the special set S if there was no input `reveal` at interface $\mathtt{M}_i$. 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 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 $\mathtt{M}_i$. On a technical level, shared randomness is needed to achieve a consistent simulation.

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. Figure 4 illustrates an example instantiation of the real and ideal worlds which should help clarifying the above descriptions.

4.3 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 $\mathtt{E}$ 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

```
Resource $\mathbf{SecNT}_n$

Initialization
  $r_1||r_2||\ldots||r_n \twoheadleftarrow (\{0,1\}^{2\kappa})^n$
                                        ▷ Common randomness
  $I_1, I_2 \leftarrow$ empty tables
                        ▷ Mapping interfaces and identities
  $J, S \leftarrow \emptyset$                ▷ Registered 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 $(m, \mathsf{ID}_s)$ at 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 $\mathtt{P} \neq \bot$ and $\mathsf{ID}_s \notin S$ then
      output $(m, \mathsf{ID}_s)$ at 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 $\mathsf{ID} \notin J \ \wedge \ I_1[\mathtt{P}_i] = \bot$ 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 $\mathsf{ID}_s \neq \bot$ and $\mathsf{ID} \in J$ 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. 5. The (unfiltered) behavior of the constructed resource.

(interface E), as described above and thus is equal to the filter ϕ^{ideal}. The following theorem says that if the signcryption scheme is secure in the respective multi-user, outsider-security and insider-security model, then we achieve the desired construction. The proof is found in the full version [3].

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. 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.,*

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

for $\varepsilon(\mathbf{D}) := n^2 \cdot \mathsf{Adv}^{\mathsf{MOS}}_{\Psi,\rho_1(\mathbf{D})} + n \cdot \mathsf{Adv}^{\mathsf{MIS\text{-}Auth}}_{\Psi,\rho_2(\mathbf{D})} + n \cdot \mathsf{Adv}^{\mathsf{MIS\text{-}Conf}}_{\Psi,\rho_3(\mathbf{D})}$, and (efficient) black-box reductions ρ_1, ρ_2, and ρ_3.

An interesting corollary for the special case when the set of interfaces with potential dishonest behavior is just {E} is the following statement: The outsider security model implies the construction of a secure network if no honest parties' keys are compromised. The formal statement and proof are given in [3].

References

1. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_6
2. An, J.H.: Authenticated encryption in the public-key setting: security notions and analyses. Cryptology ePrint Archive, Report 2001/079 (2001). http://eprint.iacr.org/2001/079
3. Badertscher, C., Banfi, F., Maurer, U.: A constructive perspective on signcryption security. Cryptology ePrint Archive, Report 2018/050 (2018). https://eprint.iacr.org/2018/050
4. Baek, J., Steinfeld, R., Zheng, Y.: Formal proofs for the security of signcryption. J. Cryptol. **20**(2), 203–235 (2007)
5. Barbosa, M., Farshim, P.: Certificateless signcryption. In: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, pp. 369–372. ACM (2008)
6. Bellare, M., Namprempre, C.: Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44448-3_41
7. Bjørstad, T.E., Dent, A.W.: Building better signcryption schemes with tag-KEMs. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 491–507. Springer, Heidelberg (2006). https://doi.org/10.1007/11745853_32
8. Boyen, X.: Multipurpose identity-based signcryption. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 383–399. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_23
9. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: Proceedings of the 42nd Symposium on Foundations of Computer Science, pp. 136–145. IEEE (2001)
10. Çapar, Ç., Goeckel, D., Paterson, K.G., Quaglia, E.A., Towsley, D., Zafer, M.: Signal-flow-based analysis of wireless security protocols. Inf. Comput. **226**, 37–56 (2013)
11. Datta, P., Dutta, R., Mukhopadhyay, S.: Compact attribute-based encryption and signcryption for general circuits from multilinear maps. In: Biryukov, A., Goyal, V. (eds.) INDOCRYPT 2015. LNCS, vol. 9462, pp. 3–24. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26617-6_1
12. Datta, P., Dutta, R., Mukhopadhyay, S.: Functional signcryption: notion, construction, and applications. In: Au, M.-H., Miyaji, A. (eds.) ProvSec 2015. LNCS, vol. 9451, pp. 268–288. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26059-4_15
13. Dent, A.W.: Hybrid signcryption schemes with insider security. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 253–266. Springer, Heidelberg (2005). https://doi.org/10.1007/11506157_22
14. Freire, E.S.V., Hesse, J., Hofheinz, D.: Universally composable non-interactive key exchange. In: Abdalla, M., De Prisco, R. (eds.) SCN 2014. LNCS, vol. 8642, pp. 1–20. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10879-7_1

15. Freire, E.S.V., Hofheinz, D., Kiltz, E., Paterson, K.G.: Non-interactive key exchange. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 254–271. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_17
16. Gjøsteen, K., Kråkmo, L.: Universally composable signcryption. In: Lopez, J., Samarati, P., Ferrer, J.L. (eds.) EuroPKI 2007. LNCS, vol. 4582, pp. 346–353. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73408-6_26
17. Gérard, F., Merckx, K.: Post-quantum signcryption from lattice-based signatures. Cryptology ePrint Archive, Report 2018/056 (2018)
18. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 15–44. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_2
19. Hofheinz, D., Matt, C., Maurer, U.: Idealizing identity-based encryption. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 495–520. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_21
20. Libert, B., Quisquater, J.J.: A new identity based signcryption scheme from pairings. In: 2003 Proceedings of the Information Theory Workshop, pp. 155–158. IEEE (2003)
21. Libert, B., Quisquater, J.-J.: Efficient signcryption with key privacy from gap Diffie-Hellman groups. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 187–200. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24632-9_14
22. Liu, J.K., Baek, J., Zhou, J.: Online/offline identity-based signcryption revisited. In: Lai, X., Yung, M., Lin, D. (eds.) Inscrypt 2010. LNCS, vol. 6584, pp. 36–51. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21518-6_3
23. Malone-Lee, J.: Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098 (2002). https://eprint.iacr.org/2002/098
24. Maurer, U.: Indistinguishability of random systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_8
25. Maurer, U.: Constructive cryptography – a new paradigm for security definitions and proofs. In: Mödersheim, S., Palamidessi, C. (eds.) TOSCA 2011. LNCS, vol. 6993, pp. 33–56. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27375-9_3
26. Maurer, U., Renner, R.: Abstract cryptography. In: Innovations in Theoretical Computer Science, pp. 1–21. Tsinghua University Press (2011)
27. Pandit, T., Pandey, S.K., Barua, R.: Attribute-based signcryption: signer privacy, strong unforgeability and IND-CCA2 security in adaptive-predicates attack. In: Chow, S.S.M., Liu, J.K., Hui, L.C.K., Yiu, S.M. (eds.) ProvSec 2014. LNCS, vol. 8782, pp. 274–290. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12475-9_19
28. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_23
29. Selvi, S.S.D., Sree Vivek, S., Pandu Rangan, C.: Identity based public verifiable signcryption scheme. In: Heng, S.-H., Kurosawa, K. (eds.) ProvSec 2010. LNCS, vol. 6402, pp. 244–260. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16280-0_17

30. Selvi, S.S.D., Sree Vivek, S., Vinayagamurthy, D., Pandu Rangan, C.: ID based signcryption scheme in standard model. In: Takagi, T., Wang, G., Qin, Z., Jiang, S., Yu, Y. (eds.) ProvSec 2012. LNCS, vol. 7496, pp. 35–52. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33272-2_4
31. Steinfeld, R., Zheng, Y.: A signcryption scheme based on integer factorization. In: Goos, G., Hartmanis, J., van Leeuwen, J., Pieprzyk, J., Seberry, J., Okamoto, E. (eds.) ISW 2000. LNCS, vol. 1975, pp. 308–322. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44456-4_23
32. Tian, Y., Peng, C.: Universally composable secure group communication. Cryptology ePrint Archive, Report 2014/647 (2014). https://eprint.iacr.org/2014/647
33. Wang, Y., Manulis, M., Au, M.H., Susilo, W.: Relations among privacy notions for signcryption and key invisible "sign-then-encrypt". In: Boyd, C., Simpson, L. (eds.) ACISP 2013. LNCS, vol. 7959, pp. 187–202. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39059-3_13
34. Young, M., Dent, A.W., Zheng, Y.: Practical Signcryption. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-89411-7
35. Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052234
36. Zheng, Y., Imai, H.: How to construct efficient signcryption schemes on elliptic curves. Inf. Process. Lett. **68**(5), 227–233 (1998)