Robust Authenticated Encryption and the Limits of Symmetric Cryptography*

Christian Badertscher[1], Christian Matt[1], Ueli Maurer[1],
Phillip Rogaway[2], and Björn Tackmann[3]

[1] Department of Computer Science, ETH Zurich, Switzerland
{badi, mattc, maurer}@inf.ethz.ch
[2] Department of Computer Science, University of California, Davis, USA
rogaway@cs.ucdavis.edu
[3] Department of Computer Science & Engineering,
University of California, San Diego, USA
btackmann@eng.ucsd.edu

Abstract. Robust authenticated encryption (RAE) is a primitive for symmetric encryption that allows to flexibly specify the ciphertext expansion, i.e., how much longer the ciphertext is compared to the plaintext. For every ciphertext expansion, RAE aims at providing the best-possible authenticity and confidentiality. To investigate whether this is actually achieved, we characterize exactly the guarantees symmetric cryptography can provide for any given ciphertext expansion. Our characterization reveals not only that RAE reaches the claimed goal, but also, contrary to prior belief, that one cannot achieve full confidentiality without ciphertext expansion. This provides new insights into the limits of symmetric cryptography.
Moreover, we provide a rigorous treatment of two previously only informally stated additional features of RAE; namely, we show how redundancy in the message space can be exploited to improve the security and we analyze the exact security loss if multiple messages are encrypted with the same nonce.

1 Introduction

Authenticity and confidentiality are arguably among the most important cryptographic objectives. Authenticated encryption is a symmetric primitive that aims to achieve both at the same time, allowing efficiency gains and reducing the risk of misuse compared to combined schemes. Several notions of authenticated encryption have emerged over a series of works [2,3,5,7,8,13], including authenticated encryption with associated data [4,12] and misuse-resistant authenticated encryption [14]. In this development, *robust authenticated encryption (RAE)*, introduced by Hoang, Krovetz, and Rogaway [6], is the latest and most ambitious notion.

* This is the full version of a paper due to appear at the 15th IMA International Conference on Cryptography and Coding (2015). The final publication will be available at link.springer.com.

Robust authenticated encryption allows to specify the ciphertext expansion λ that determines how much longer ciphertexts are compared to the corresponding plaintexts. Its self-declared goal in [6] is to provide the best-possible authenticity and confidentiality for every choice of λ. This raises the question of what best-possible authenticity and confidentiality is, and whether RAE actually achieves it. We provide a formal model that allows us to investigate this question and answer it in the affirmative. We further show how to use verifiable redundancy to improve security, and we show what security guarantees remain if values intended as nonces are reused. Both questions were addressed in [6] but not proven formally.

1.1 Robust Authenticated Encryption

An RAE scheme consists of a key distribution $\mathcal{K}$, a deterministic encryption algorithm $\mathcal{E}$, and a deterministic decryption algorithm $\mathcal{D}$. The encryption algorithm takes as input a key K, a nonce N, associated data A, the ciphertext expansion λ, and a message M. It outputs a ciphertext C. The decryption algorithm takes as input K, N, A, λ, and C, and returns the corresponding message M (or $\perp$ if C is an invalid ciphertext). In [6], the security of an RAE scheme is defined via a game in which an adversary has access to two oracles and has to distinguish between two possible settings. In the first setting, the oracles correspond to the encryption and decryption algorithm of the RAE scheme, where the key is fixed in the beginning and chosen according to $\mathcal{K}$. In the second setting, the first oracle chooses for each N, A, λ, and message length ℓ an injective function that maps strings of length ℓ to strings of length $\ell + \lambda$. On input (N, A, λ, M), the oracle answers by evaluating the corresponding function. The second oracle corresponds to the partially defined inverse of that function that answers $\perp$ if the given value has no preimage. An RAE scheme is secure if these two settings are indistinguishable for efficient adversaries. While this seems to be a strong guarantee, it is not clear which security such a scheme actually provides in a specific application and whether it is best-possible.

1.2 Security Definitions and Constructive Cryptography

Since game-based security definitions only capture what an adversary can do in a specific attack-scenario, they inherently fall short of providing guarantees that hold in any possible application of the scheme. To capture what RAE schemes achieve, we formulate our results using the constructive cryptography framework by Maurer and Renner [9,10]. The central idea of this framework is that the resources available to the parties, such as communication channels or shared randomness like cryptographic keys, are made explicit. The goal of a cryptographic protocol is then to construct, from certain existing resources, another resource that can again be used by higher-level protocols or applications. For example, the goal of an authentication scheme can be formalized as constructing an authenticated channel from a shared secret key and an insecure channel. The insecure channel allows a sender, say Alice, to send messages to a receiver, say Bob, but entirely leaks the transmitted messages to the adversary and additionally allows the

adversary to delete messages and inject arbitrary messages; an authenticated channel still leaks the messages but only allows the adversary to delete messages and to deliver the messages originally sent. A conventional encryption scheme is supposed to construct a secure channel from a shared secret key and an authenticated channel, where the secure channel restricts the leakage to the length of the transmitted messages. The composition theorem of constructive cryptography guarantees that if two protocols achieve these constructions, the composed protocol constructs a secure channel from two shared secret keys and an insecure channel, i.e, the security of the overall construction follows from the security of the individual construction steps. On the other hand, authenticated encryption directly achieves the overall construction.

1.3 Our Contributions

In the vein of [1] and accounting for the associated data RAE schemes support, we formalize the goal of RAE as constructing an *augmented secure channel (ASC)* from a shared secret key and an insecure channel. An ASC takes as input from the sender a tuple (A, M), leaks A and the length of M to the adversary, and allows the adversary to either deliver the pair (A, M) or to terminate the channel. This channel provides authenticity for both A and M, but confidentiality is only guaranteed for the message M. The value A can for example be used to authenticate non-private header information; see [1] for an application of ASC in the context of TLS 1.3.

Uniform random injection resource. Instead of directly constructing channels from a shared secret key and an insecure channel, we introduce an intermediate system *URI (uniform random injection)* that provides the sender and receiver access to the same uniform random injections and their inverses chosen as follows: For each combination of N, A, λ, and message length ℓ, an injective function that maps strings of length ℓ to strings of length $\ell + \lambda$ is chosen uniformly at random.

As we shall see, this resource can be constructed from a shared secret key using an RAE scheme in a straightforward manner. We then construct several channels from URI and an insecure channel. The advantage of this approach is that all further constructions in this paper are information-theoretic, i.e., we do not have to relate the security of each construction step to the RAE security game. Instead, we can rely on the composition theorem to guarantee the security of the overall construction.

Random injection channel. We show that one can construct a channel we call *RIC (random injection channel)* from URI and an insecure channel by fixing λ and using a counter as the nonce. RIC can be seen as a further intermediate step towards constructing ASC, that in addition allows us to analyze best-possible security.

The channel RIC takes as input a pair (A, M) from the sender and leaks A and the length of M to the adversary. The adversary can deliver the pair (A, M), and further at any point in time try to inject a new message of length ℓ and some value A. The probability with which such an injection is successful depends on λ and ℓ. In case of a success, an almost uniform message of length ℓ from the message space together with A is delivered to the receiver. If an injection was successful and the tuple (A, M) was received, and if the sender subsequently sends exactly the pair (A, M), then the adversary is notified about this repetition.

Best possible authenticity and confidentiality. If ASC is considered as the ultimate goal of RAE and authenticated encryption in general, the only shortcomings of RIC are that it is possible to inject messages with positive probability and that, if an attempted message injection was successful, the channel leaks a certain repetition to the adversary. While the first shortcoming is a lack of authenticity, the second one is a lack of confidentiality. While the type of leakage violating confidentiality might seem artificial, we describe an application in which such leakage might be problematic. Briefly, the leakage can reveal hidden information flow from the receiver to the sender.

We then analyze whether RAE really achieves the best-possible authenticity and confidentiality by bounding the probabilities of successful message injections and of leaking this particular repetition pattern for arbitrary schemes for achieving authenticity and confidentiality. While it is straightforward to see that authenticity requires redundancy and therefore a large ciphertext expansion, one might hope that the repetition leakage can be avoided. We prove that this is not the case, i.e., we show that the probability of an adversary being able to observe such a repetition is at least as high as in RIC, no matter what scheme is used or which setup assumptions are made.

To illustrate this lack of confidentiality for a concrete scheme, consider the following scenario in which the one-time pad is used over an insecure channel: Assume the attacker injects a ciphertext to Bob who decrypts it using the shared secret key and outputs the resulting message. Further assume Alice afterwards sends a message to Bob which results in the attacker seeing the same ciphertext. In that case, the attacker learns the fact that the message sent by Alice equals the message output by Bob. This contradicts the understanding of confidentiality as revealing nothing except the length of the transmitted message.[4] Our results generalize this observation to arbitrary schemes. We thereby refine the understanding of what symmetric cryptography can and cannot achieve by showing that confidentiality, quite surprisingly, also requires redundancy in the ciphertexts when only insecure channels and an arbitrary setup are assumed, even if the protocol can keep state.

[4] This also contradicts a prior result in [11] that claims that the one-time pad constructs a certain (fully) confidential channel, a so-called XOR-malleable channel, from an insecure channel and a shared key. The proof in that paper is flawed in that the simulation fails if more ciphertexts are injected than messages sent.

Augmented secure channels and message redundancy. Since the probability of successful message injections decreases exponentially with λ, RIC is indistinguishable from ASC for large λ. We further provide a construction that incorporates an idea from [6] to exploit the redundancy in messages to achieve a better bound. Our construction reveals the exact trade-off between ciphertext expansion and redundancy to achieve a required security level.

Nonce-reuse resistance. It was claimed in [6] that reusing nonces only results in leaking the repetition pattern of messages, but does not compromise security beyond that. However, the claim was neither formalized nor proven. We fill this gap by introducing the channel resource *RASC (Repetition ASC)* that, aside of the length of each message, leaks the repetition pattern of the transmitted messages to the adversary. Furthermore, the adversary can deliver messages out-of-order and arbitrarily replay messages. We show that RASC can be constructed from URI and an insecure channel if the used nonce is always the same. This confirms the informal claim from [6] and makes explicit that some authenticity is lost by allowing the adversary to reorder messages.

2 Preliminaries

2.1 Notation for Systems and Algorithms

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 $\mathcal{X}$ over some set, $x \twoheadleftarrow \mathcal{X}$ denotes sampling x according to $\mathcal{X}$. For a finite set X, $x \twoheadleftarrow X$ denotes assigning to x a uniformly random value in X.

We denote the empty list by $[\,]$ and for a list L, $L \parallel x$ denotes the list L with x appended. Furthermore, $|L|$ denotes the number of elements in L and the ith element in L is denoted by $L[i]$ for $i \in \{1, \ldots, |L|\}$. For a FIFO queue $\mathcal{Q}$, we write $\mathcal{Q}.\texttt{enqueue}(x)$ to insert x into the queue and $\mathcal{Q}.\texttt{dequeue}()$ to retrieve (and remove) the element of the queue that was inserted first among all remaining elements.

For $n, m \in \mathbb{N}$, $\mathrm{Inj}\,(\Sigma^n, \Sigma^m)$ denotes the set of injective functions $\Sigma^n \to \Sigma^m$. For an injective function $f\colon X \to Y$, we denote by f^{-1} the function $Y \to X \cup \{\bot\}$ that maps y to the preimage of y under f if existing, and to the distinct element $\bot$ otherwise.

Typically queries to systems consist of a suggestive keyword and a list of arguments (e.g., (send, M) to send the message M). We ignore keywords in writing the domains of arguments, e.g., $(\mathsf{send}, M) \in \mathcal{M}$ indicates that $M \in \mathcal{M}$.

2.2 Constructive Cryptography

Constructive cryptography makes statements about *constructions* of *resources* from other resources. A resource is a system with interfaces via which the resource interacts with its environment and which can be thought of as being assigned to

parties. All resources in this paper have an interface A for the sender (Alice), an interface B for the receiver (Bob), and an interface E for the adversary (Eve). In our security statements, we are interested in the advantage of a distinguisher $\mathbf{D}$ in distinguishing two resources, say $\mathbf{R}$ and $\mathbf{S}$ which is defined as

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

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

Converters are systems that can be attached to an interface of a resource to change the inputs and outputs at that interface, which yields another resource. A converter is a system with two interfaces: the *inner interface* in is connected to an interface of a resource and the *outer interface* out, becomes the new connection point of that resource towards the environment. The protocols of the honest parties and simulators correspond to converters.

We directly state the central definition of a construction of [9] and briefly explain the relevant conditions.

Definition 1. *Let* $\mathbf{R}$ *and* $\mathbf{S}$ *be resources and let* $\mathsf{noAtck}_{\mathbf{R}}$ *and* $\mathsf{noAtck}_{\mathbf{S}}$ *be converters that describe the default behavior at interface* E *when no attacker is present. Let* ε *be a function that maps distinguishers to a value in* $[-1, 1]$ *and let* sim *be a converter (the simulator). A protocol, i.e., a pair* $(\mathsf{conv}_1, \mathsf{conv}_2)$ *of converters, constructs resource* $\mathbf{S}$ *from resource* $\mathbf{R}$ *within* ε *and with respect to the pair* $(\mathsf{noAtck}_{\mathbf{R}}, \mathsf{noAtck}_{\mathbf{S}})$ *and the simulator* sim*, if for all distinguishers* $\mathbf{D}$*,*

$$\Delta^{\mathbf{D}}\left(\mathsf{conv}_1{}^{\mathsf{A}}\mathsf{conv}_2{}^{\mathsf{B}}\mathsf{noAtck}_{\mathbf{R}}{}^{\mathsf{E}}\,\mathbf{R}, \mathsf{noAtck}_{\mathbf{S}}{}^{\mathsf{E}}\,\mathbf{S}\right) \leq \varepsilon(\mathbf{D}) \qquad \textit{(Availability)}$$

$$\Delta^{\mathbf{D}}\left(\mathsf{conv}_1{}^{\mathsf{A}}\mathsf{conv}_2{}^{\mathsf{B}}\,\mathbf{R}, \mathsf{sim}^{\mathsf{E}}\,\mathbf{S}\right) \leq \varepsilon(\mathbf{D}). \qquad \textit{(Security)}$$

The first condition ensures that the protocol implements the required functionality if there is no attacker. For example, for communication channels, all sent messages have to be delivered when no attacker interferes with the protocol.

The second condition ensures that whatever Eve can do with the assumed resource, she could do as well with the constructed resource by using the simulator sim. Turned around, if the constructed resource is secure by definition, 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. This is proven in [9].

2.3 Robust Authenticated Encryption

Let Σ be an alphabet (a finite nonempty set). Typically an element of Σ is a bit ($\Sigma = \{0,1\}$) or a byte ($\Sigma = \{0,1\}^8$). For a string $x \in \Sigma^*$, $|x|$ denotes its length. We define the syntax of a robust authenticated encryption scheme following [6].

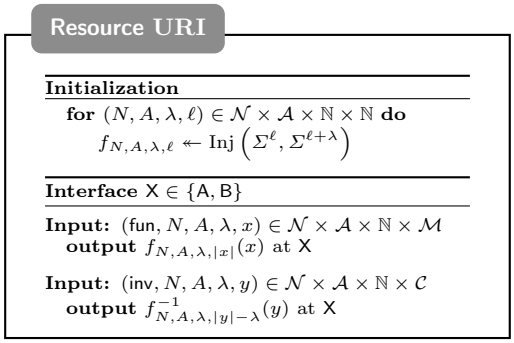

Fig. 1. Uniform random injection resource. Interface E remains inactive.

Definition 2. *A* robust authenticated encryption (RAE) scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *consists of a key distribution* $\mathcal{K}$*, a deterministic encryption algorithm* $\mathcal{E}$ *that maps a key* $K \in \mathcal{K}$*, a nonce* $N \in \mathcal{N}$*, associated data* $A \in \mathcal{A}$*, ciphertext expansion* $\lambda \in \mathbb{N}$*, and a message* $M \in \mathcal{M}$ *to a ciphertext* $C \in \mathcal{C}$*, and a deterministic decryption algorithm* $\mathcal{D}$ *that maps* (K, N, A, λ, C) *to an element in* $\mathcal{M} \cup \{\perp\}$*. We assume the domains* $\mathcal{N}$*,* $\mathcal{A}$*,* $\mathcal{M}$*, and* $\mathcal{C}$ *are equal to* Σ^**. We write* $\mathcal{E}_K^{N,A,\lambda}$ *and* $\mathcal{D}_K^{N,A,\lambda}$ *for the functions* $\mathcal{E}(K, N, A, \lambda, \cdot)$ *and* $\mathcal{D}(K, N, A, \lambda, \cdot)$*, respectively. We require that* $\mathcal{D}_K^{N,A,\lambda}\left(\mathcal{E}_K^{N,A,\lambda}(M)\right) = M$ *for all* K, N, A, λ, M*.*

3 Shared Uniform Random Injections and RAE Security

In this section, we describe the resource **URI** that grants access to shared uniform random injections and their inverses at interfaces A and B, and no access at interface E. We then use **URI** to define the security of RAE schemes and show that any RAE scheme that satisfies this definition can be used to construct **URI** from a shared secret key. Though syntactically different, it is easy to see that our definition is equivalent to the security definition from [6]. We recall that definition and prove the equivalence in Appendix A.

We first give a definition for the uniform random injection system **URI**.

Definition 3. *The resource* **URI** *has interfaces* A*,* B*, and* E *and takes inputs of the form* $(\mathsf{fun}, N, A, \lambda, x)$ *and* $(\mathsf{inv}, N, A, \lambda, y)$ *at interfaces* A *and* B *for* $N \in \mathcal{N}$*,* $A \in \mathcal{A}$*,* $\lambda \in \mathbb{N}$*,* $x \in \mathcal{M}$*, and* $y \in \mathcal{C}$*. Any input at interface* E *is ignored. We assume the domains* $\mathcal{N}$*,* $\mathcal{A}$*,* $\mathcal{M}$*, and* $\mathcal{C}$ *are equal to* Σ^**. On input* $(\mathsf{fun}, N, A, \lambda, x)$ *at interface* A *or* B*, it returns* $f_{N,A,\lambda,|x|}(x)$ *at the same interface. On input* $(\mathsf{inv}, N, A, \lambda, y)$*, it returns* $f^{-1}_{N,A,\lambda,|y|-\lambda}(y)$ *if* $|y| > \lambda$*, and* $\perp$ *otherwise. The function* $f_{N,A,\lambda,\ell}$ *is chosen uniformly at random from the set* $\mathrm{Inj}\left(\Sigma^{\ell}, \Sigma^{\ell+\lambda}\right)$ *when needed for the first time and reused for later inputs. See Fig. 1 for pseudocode.*

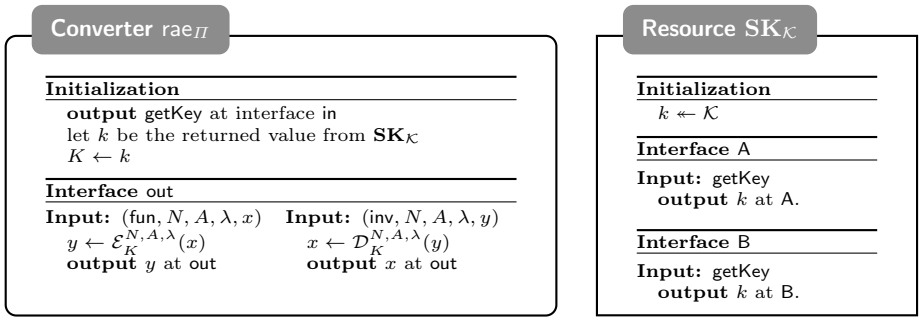

Fig. 2. Protocol that constructs **URI** from a shared secret key (left) and the shared secret key resource (right). For the shared key resource, interface E remains inactive.

3.1 Definition of RAE Security and Construction of URI

We define a shared key resource $\mathbf{SK}_{\mathcal{K}}$ for some key distribution $\mathcal{K}$. The resource initially chooses a key according to $\mathcal{K}$ and outputs this key to interfaces A and B while interface E remains inactive, see Fig. 2. Slightly abusing notation, we will also refer to the key space by $\mathcal{K}$ whenever no confusion can arise. We further define the converter rae_{Π} that is based on an RAE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. First, rae_{Π} requests the key from $\mathbf{SK}_{\mathcal{K}}$. For any input at the outer interface, it evaluates $\mathcal{E}$ or $\mathcal{D}$ using that key (and the arguments provided in the input) and returns the result. The code is given in Fig. 2.

We consider an RAE scheme secure if all efficient distinguishers have poor advantage with respect to the following definition.

Definition 4. *The advantage of a distinguisher* $\mathbf{D}$ *for an RAE scheme* Π *is quantified as*

$$\mathbf{Adv}^{\mathrm{rae}}_{\Pi}(\mathbf{D}) \coloneqq \Delta^{\mathbf{D}}\left(\mathsf{rae}_{\Pi}{}^{\mathsf{A}}\,\mathsf{rae}_{\Pi}{}^{\mathsf{B}}\,\mathbf{SK}_{\mathcal{K}}, \mathbf{URI}\right).$$

It is straightforward to see that the definition implies the following construction statement, where the converters sim and noAtck are defined as the converter that blocks any interaction at the interface it is connected to.

Lemma 1. *The protocol* $(\mathsf{rae}_{\Pi}, \mathsf{rae}_{\Pi})$ *constructs* **URI** *from* $\mathbf{SK}_{\mathcal{K}}$ *within* $\mathbf{Adv}^{\mathrm{rae}}_{\Pi}$ *with respect to* $(\mathsf{noAtck}, \mathsf{noAtck})$ *and simulator* sim *defined above.*

Proof. Since interface E of $\mathbf{SK}_{\mathcal{K}}$ and **URI** are inactive, the converters sim and noAtck have no effect when connected to that interface, i.e., $\mathsf{noAtck}^{\mathsf{E}}\,\mathbf{SK}_{\mathcal{K}} = \mathbf{SK}_{\mathcal{K}}$ and $\mathsf{noAtck}^{\mathsf{E}}\,\mathbf{URI} = \mathsf{sim}^{\mathsf{E}}\,\mathbf{URI} = \mathbf{URI}$. Thus, both the availability and the security condition of the construction are equivalent to

$$\Delta^{\mathbf{D}}\left(\mathsf{rae}_{\Pi}{}^{\mathsf{A}}\,\mathsf{rae}_{\Pi}{}^{\mathsf{B}}\,\mathbf{SK}_{\mathcal{K}}, \mathbf{URI}\right) \leq \mathbf{Adv}^{\mathrm{rae}}_{\Pi}(\mathbf{D})$$

for all distinguishers $\mathbf{D}$, which trivially holds by definition of $\mathbf{Adv}^{\mathrm{rae}}_{\Pi}$. □

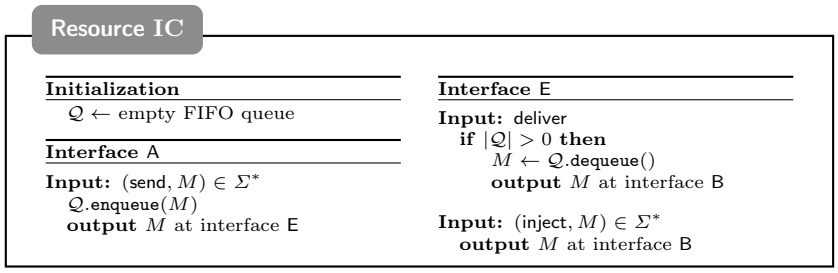

Fig. 3. The insecure channel resource.

4 Random Injection Channels: Security for any Expansion

The goal of the current section is to examine the exact security achieved by RAE schemes when used to protect communication. We present constructions of specific secure channels from insecure channels and resource **URI** where each type of secure channel precisely captures the amount of leakage to an eavesdropper and the possible influence of an adversary interfering with the protocol execution. As an additional result, we are able to answer what best-possible communication security is and observe that RAE schemes achieve this level of security.

The insecure channel **IC** allows messages $m \in \mathcal{M}$ to be input repeatedly at interface A. Each message is subsequently leaked at the E-interface. At interface E, arbitrary messages (including those that were previously input at interface A) can be injected such that they are delivered to B. This channel does not give any security guarantees to Alice and Bob. A formal description is provided in Fig. 3. For the rest of this paper, the message space of the insecure channel is Σ^*.

4.1 Constructing Random Injection Channels

The constructed channel. The channel we construct in this section is defined in Fig. 4 and can be roughly described as follows: It allows to repeatedly send pairs (A_i, M_i) in an ordered fashion from a sender to a receiver. Each pair consists of the associated data A_i and the message M_i. The attacker is limited to seeing the associated data A_i and the length of the message $|M_i|$ of each transmitted pair. Additionally, the attacker learns whether the ith injected pair equals the one that is currently sent.

The attacker can either deliver the next legitimate pair (A_i, M_i) or try to inject a pair (A, M) that is different from (A_i, M_i). Such an injection is only successful with a certain probability. The associated data A and the length ℓ of the message are chosen by the attacker and M is a uniformly random message of length ℓ if $A \neq A_i$. Otherwise, M is a uniformly random message $M \neq M_i$ of length ℓ. If an injection attempt is not successful, the resource does not deliver messages at interface B any more and signals an error by outputting $\perp$. If the

Resource $\mathbf{RIC}_\lambda$

Initialization

$\mathcal{S} \leftarrow$ empty FIFO queue
$i \leftarrow 0$
$\mathcal{R} \leftarrow [\,]$

Interface A

Input: $(\mathsf{send}, A, M) \in \mathcal{A} \times \mathcal{M}$
$i \leftarrow i + 1$
if $i > |\mathcal{R}|$ **then**
$\mathcal{S}.\mathsf{enqueue}((A, M))$
if $i \leq |\mathcal{R}|$ **and** $\mathcal{R}[i] = (A, M)$ **then**
output repeat at interface E
else
output $(A, |M|)$ at interface E

Interface E

Input: deliver
if $|\mathcal{S}| > 0$ **and** halt $= 0$ **then**
$(A, M) \leftarrow \mathcal{S}.\mathsf{dequeue}()$
$\mathcal{R} \leftarrow \mathcal{R} \parallel (A, M)$
output (A, M) at interface B

Input: $(\mathsf{inject}, A, \ell) \in \mathcal{A} \times \mathbb{N}_{>0}$
if halt $= 0$ **then**
if $|\mathcal{S}| > 0$ **then**
$(A', M') \leftarrow \mathcal{S}.\mathsf{dequeue}()$
if $A = A'$ **and** $\ell = |M'|$ **then**
$M \leftarrow \textsc{SampleExcl}(\ell, \lambda, M')$
else
$M \leftarrow \textsc{Sample}(\ell, \lambda)$
else
$M \leftarrow \textsc{Sample}(\ell, \lambda)$
if $M \neq \perp$ **then**
$\mathcal{R} \leftarrow \mathcal{R} \parallel (A, M)$
output (A, M) at B
else
output $\perp$ at B
halt $\leftarrow 1$

function $\textsc{Sample}(\ell, \lambda)$
$B \twoheadleftarrow \text{Bernoulli}\left(|\Sigma|^{-\lambda}\right)$
if $B = 1$ **then**
$M \twoheadleftarrow \Sigma^\ell$
return M
else
return $\perp$

function $\textsc{SampleExcl}(\ell, \lambda, m)$
$B \twoheadleftarrow \text{Bernoulli}\left(\frac{|\Sigma|^\ell - 1}{|\Sigma|^{\ell+\lambda} - 1}\right)$
if $B = 1$ **then**
$M \twoheadleftarrow \Sigma^\ell \setminus \{m\}$
return M
else
return $\perp$

Fig. 4. Description of $\mathbf{RIC}_\lambda$. In the description, Bernoulli(p) denotes the distribution over $\{0, 1\}$, where 1 has probability p and 0 has probability $1 - p$.

Converter snd_λ

Initialization

$i \leftarrow 0$

Interface out

Input: $(\mathsf{send}, A, M) \in \mathcal{A} \times \mathcal{M}$
output $(\mathsf{fun}, i, A, \lambda, M)$ to **URI**
let C be the return value from **URI**
$i \leftarrow i + 1$
output $(\mathsf{send}, (A, C))$ to **IC**

Converter rcv_λ

Initialization

$\mathcal{Q} \leftarrow$ empty FIFO queue
$i \leftarrow 0$
halt $\leftarrow 0$

Interface in

Input: $(A, C) \in \mathcal{A} \times \mathcal{C}$ (from **IC**)
if halt $= 0$ **then**
output $(\mathsf{inv}, i, A, \lambda, C)$ to **URI**
let M be the return value from **URI**
$i \leftarrow i + 1$
if $M \neq \perp$ **then**
output (A, M) at out
else
output $\perp$ at out
halt $\leftarrow 1$

Fig. 5. The converters for the sender (left) and the receiver (right) to construct $\mathbf{RIC}_\lambda$.

adversary injects the ith message, the legitimate ith message cannot be delivered anymore.[5]

The success probability of an injection attempt depends on the expansion λ and the specified message length ℓ and whether the sender's queue $\mathcal{S}$ is empty or not. The exact probabilities are quantified by the two sampling functions Sample and SampleExcl. The function Sample first samples a bit according to the probability that a fixed element from $\Sigma^{\ell+\lambda}$ has a preimage under a uniform random injection $\Sigma^{\ell} \to \Sigma^{\ell+\lambda}$. If the bit is 1, a uniform random preimage is returned. The function SampleExcl essentially does the same, but the domain and codomain are both reduced by one element.[6]

Protocol. We construct resource $\mathbf{RIC}_\lambda$ from $[\mathbf{URI}, \mathbf{IC}]$ which denotes the resource that provides at each interface access to the corresponding interface of both resources. Our protocol specifies a particular but very natural usage of **URI** where the nonce is implemented as a counter value.[7] We present the protocol as pseudocode in Fig. 5. The converter for the sender, snd_λ, accepts inputs of the form (send, A, M) at its outer interface. It outputs $(\mathsf{fun}, i, A, \lambda, M)$ at the inner interface to **URI**. The nonce is implemented as a counter and λ is the parameter of the protocol. Once a ciphertext is received as a return value from **URI**, it is output together with its associated data at the inner interface for the insecure channel **IC**. The receiver converter rcv_λ receives ciphertexts together with the associated data at its inner interface from **IC** and decrypts C using parameters A, i and λ. On success, the corresponding plaintext is output at the outer interface. If decryption fails, the converter stops and signals an error by outputting $\perp$.

Construction statement. In order to show that the protocol $(\mathsf{snd}_\lambda, \mathsf{rcv}_\lambda)$ constructs $\mathbf{RIC}_\lambda$ from $[\mathbf{URI}, \mathbf{IC}]$, we prove both conditions of Definition 1. For all channels, the converter noAtck corresponds to the converter dlv that on any input at its inner interface, outputs $\mathsf{deliver}$ to the channel connected to its inner interface and blocks any interaction at its outer interface.

Theorem 1. *Let $\lambda \in \mathbb{N}$. The protocol $(\mathsf{snd}_\lambda, \mathsf{rcv}_\lambda)$ constructs resource $\mathbf{RIC}_\lambda$ from $[\mathbf{URI}, \mathbf{IC}]$ with respect to $(\mathsf{dlv}, \mathsf{dlv})$ and simulator $\mathsf{sim}_{\mathrm{RIC}}$ as defined in Fig. 6. More specifically, for all distinguishers $\mathbf{D}$*

$$\Delta^{\mathbf{D}}\left(\mathsf{snd}_\lambda^{\mathsf{A}}\mathsf{rcv}_\lambda^{\mathsf{B}}\mathsf{dlv}^{\mathsf{E}}[\mathbf{URI}, \mathbf{IC}], \mathsf{dlv}^{\mathsf{E}}\mathbf{RIC}_\lambda\right) = 0 \tag{1}$$

and

$$\Delta^{\mathbf{D}}\left(\mathsf{snd}_\lambda^{\mathsf{A}}\mathsf{rcv}_\lambda^{\mathsf{B}}[\mathbf{URI}, \mathbf{IC}], \mathsf{sim}_{\mathrm{RIC}}^{\mathsf{E}}\mathbf{RIC}_\lambda\right) = 0. \tag{2}$$

[5] This relates to the security of RAE schemes which ensures that the message cannot be decrypted using a wrong nonce. In our construction, the nonce is implemented as the sequence number.

[6] This ensures that the injected message is different from the one that the sender provided.

[7] Implementing the nonce as a counter allows to maintain the order of messages.

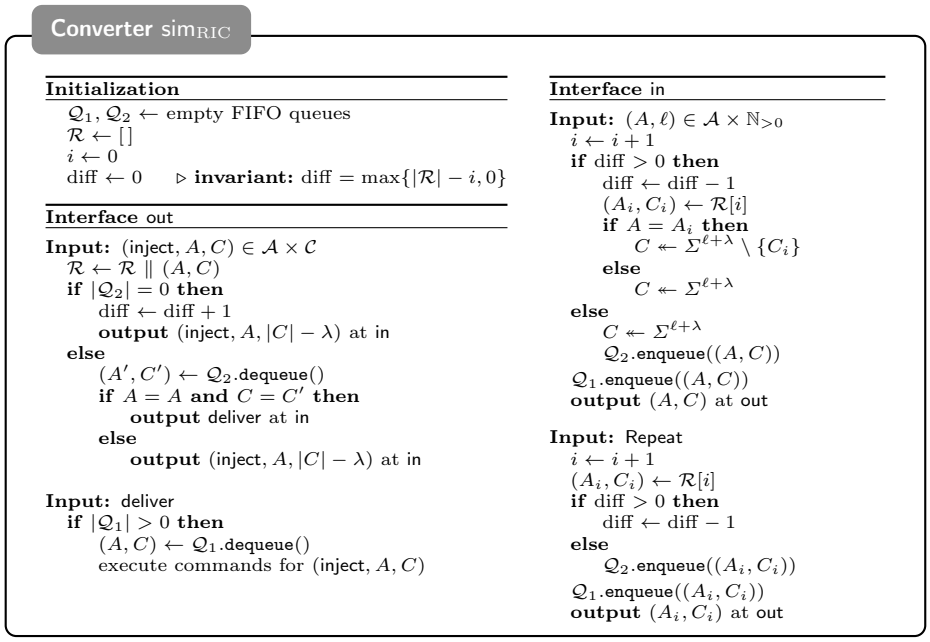

Fig. 6. Simulator for the security condition of the construction of $\mathbf{RIC}_\lambda$.

Proof. We first prove the security condition (2) by analyzing the input-output behavior of both systems involved. To this end, we consider the possible inputs at each interface.

On the ith input $(\mathsf{send}, A_i, M_i)$ **at interface** A**:** In $\mathsf{snd}_\lambda^{\mathsf{A}}\mathsf{rcv}_\lambda^{\mathsf{B}}[\mathbf{URI}, \mathbf{IC}]$, converter snd_λ evaluates $\mathbf{URI}$ on input (i, A_i, λ, M_i), where the counter i is the nonce and A_i is the associated data. The associated data is sent together with $\mathbf{URI}$'s return value C_i over $\mathbf{IC}$. If this is the first query to $\mathbf{URI}$ with parameters i and A_i, then the output C_i is distributed uniformly at random over $\Sigma^{|M_i|+\lambda}$. If there has been a query $(\mathsf{inv}, i, A_i, C_i')$ before at interface B of $\mathbf{URI}$, with $|C_i'| = |M_i| + \lambda$ that has generated the output $M_i' = M_i$, then C_i is determined to be equal to C_i'. If $M_i \neq M_i'$, the output is C_i is distributed uniformly at random over $\Sigma^{|M_i|+\lambda} \setminus \{C_i'\}$.
In system $\mathsf{sim}_{\mathrm{RIC}}^{\mathsf{E}}\mathbf{RIC}_\lambda$, (A_i, M_i) is inserted into the senders queue of $\mathbf{RIC}_\lambda$. If there has already been an ith output (A_i', M_i') with $A_i' = A_i$ and $M_i = M_i'$ at interface B, then $\mathbf{RIC}_\lambda$ outputs repeat at interface E and the simulator outputs the ciphertext and the associated data that was input before at its outer interface as the ith injection. Otherwise, $\mathbf{RIC}$ outputs the pair $(A_i, |M_i|)$ to $\mathsf{sim}_{\mathrm{RIC}}$. The simulator checks whether there exists an ith injected associated data-ciphertext pair (A_i', C_i') with $A_i' = A_i$ and $|C'| = |M_i| + \lambda$. If such a pair exists, which is the case if diff > 0, $\mathsf{sim}_{\mathrm{RIC}}$ generates a uniformly random string $C_i \in \Sigma^{|M_i|+\lambda} \setminus \{C_i'\}$ and outputs (A_i, C_i) at its outer interface. If there is no such pair, i.e., if diff $= 0$, $\mathsf{sim}_{\mathrm{RIC}}$ generates a uniformly random string

$C_i \in \Sigma^{|M_i|+\lambda}$ and outputs (A_i, C_i) at its outer interface. In both cases, $\mathsf{sim}_{\mathrm{RIC}}$ stores (A_i, C_i) in its own queues for later reference. $\mathcal{Q}_1$ simulates the queue of the insecure channel in the real world (to correctly simulate deliver-queries). $\mathcal{Q}_2$ stores the associated data-ciphertext pairs of the corresponding entries in the queue $\mathcal{S}$ of $\mathbf{RIC}_\lambda$. We observe that the case distinctions made by $\mathsf{sim}_{\mathrm{RIC}}$ correspond exactly to the cases that happen in the real system. Hence, the output distribution at interface E is identical to above.

On the ith input $(\mathsf{inject}, A_i, C_i)$ **at interface** E**:** In the following, we assume that interface B has not output $\perp$ already, as otherwise the two systems behave identically anyway.

In $\mathsf{snd}^{\mathsf{A}}_{\lambda}\mathsf{rcv}^{\mathsf{B}}_{\lambda}[\mathbf{URI}, \mathbf{IC}]$, the converter rcv_λ queries $\mathbf{URI}$ with $(\mathsf{inv}, i, A_i, \lambda, C_i)$ to receive the preimage M_i of C_i under parameters (i, A_i, λ). Let us first assume that there has been an ith input before at interface A (A'_i, M'_i) and that $\mathbf{URI}$ generated the ciphertext C'_i on that input. There are three cases to consider to determine the output distribution at interface B:

1. If $A_i = A'_i$ and $C_i = C'_i$, then $M_i = M'_i$ holds with probability 1 and $\mathsf{rcv}_{v,\lambda}$ outputs M'_i.
2. If $A_i \neq A'_i$ or $|C_i| \neq |C'_i|$, $\mathbf{URI}$ generates a fresh uniform injection $\Sigma^{|C_i|-\lambda} \to \Sigma^{|C_i|}$ and there is only an output at interface B if C_i has a preimage under the chosen injection. This is the case with probability
$$\frac{\left|\Sigma^{|C_i|-\lambda}\right|}{\left|\Sigma^{|C_i|}\right|} = |\Sigma|^{-\lambda}.$$
Given that C_i has a preimage, then the output at interface B is (A_i, M_i), where M_i is distributed uniformly over $\Sigma^{|C_i|-\lambda}$, since all preimages are equally likely. If C_i does not have a preimage, $\mathsf{rcv}_{v,\lambda}$ sets halt $\leftarrow 1$ and outputs $\perp$.
3. If $A_i = A'_i$, $C_i \neq C'_i$, and $|C_i| = |C'_i|$, $\mathbf{URI}$ has already generated an injection $f\colon \Sigma^{|C_i|-\lambda} \to \Sigma^{|C_i|}$ on the ith input (A'_i, M'_i) at interface A such that $f(M'_i) = C'_i$, and there is only an output at interface B if C_i has a preimage under f. Since there are $\left|\Sigma^{|C_i|-\lambda} \cap \mathcal{M}_v\right| - 1$ possible preimages different from M'_i and each of them is mapped to one out of $\left|\Sigma^{|C_i|}\right| - 1$ possible ciphertexts different from C'_i, the probability that C_i has a preimage is
$$\frac{\left|\Sigma^{|C_i|-\lambda}\right| - 1}{\left|\Sigma^{|C_i|}\right| - 1}.$$
If C_i has a preimage, then the output at interface B is (A_i, M_i), where M_i is distributed uniformly over $\Sigma^{|C_i|-\lambda} \setminus \{M'_i\}$, where M'_i is excluded due to injectivity and the remaining preimages are equally likely. If C_i does not have a preimage, $\mathsf{rcv}_{v,\lambda}$ sets halt $\leftarrow 1$ and outputs $\perp$.

If there has not yet been an ith input (A'_i, M'_i) at interface A (i.e., there are more messages injected than sent), the behavior is the same as in item 2 above, because $\mathbf{URI}$ generates a fresh uniform injection $\Sigma^{|C_i|-\lambda} \to \Sigma^{|C_i|}$ for each counter value i.

In $\mathsf{sim}^{\mathsf{E}}_{\mathrm{RIC}}\mathbf{RIC}_\lambda$, let us again first assume that there are more messages sent than injected, i.e., diff $= 0$. In that case, $\mathsf{sim}_{\mathrm{RIC}}$ retrieves the front element of $\mathcal{Q}_2$ which contains the simulated pair (A'_i, C'_i) of the front element of $\mathcal{S}$, say (A'_i, M'_i) (where M'_i is the next message that can be delivered reliably). If $A_i = A'_i$ and $C_i = C'_i$ then $\mathsf{sim}_{\mathrm{RIC}}$ delivers the next element in the sender queue of $\mathbf{RIC}_\lambda$ and hence the ith message M'_i is output. This corresponds to item 1 above. In the other two cases, $\mathsf{sim}_{\mathrm{RIC}}$ outputs $(\mathsf{inject}, A, |C_i| - \lambda)$. The behavior of $\mathbf{RIC}_\lambda$ is then as follows: Let $\ell := |C_i| - \lambda$. If $\ell = |M'_i|$ and $A_i = A'_i$, the output is $\perp$ with probability $1 - \frac{|\Sigma|^\ell - 1}{|\Sigma|^{\ell+\lambda} - 1}$. If the output is not $\perp$, the message is chosen uniformly at random from $\Sigma^\ell \setminus \{M_i\}$. This behavior follows from the call to the function $\textsc{SampleExcl}(\ell, \lambda, M_i)$. If $\ell \neq |M_i|$ or $A_i \neq A_i$, $\mathbf{RIC}_\lambda$ either outputs $\perp$ and halts with probability $1 - |\Sigma|^{-\lambda}$ or, conditioned on not being $\perp$, the output at interface B is (A_i, M_i), where M_i is chosen uniformly at random from Σ^ℓ. This behavior follows from the call to the function $\textsc{Sample}(\ell, \lambda)$.

Finally, if there is no element in the sender's queue $\mathcal{S}$, i.e., if there are at least as many messages injected as sent, then the output distribution is identical to the case $\ell \neq |M_i|$ or $A_i \neq A_i$ like in the real world.

We conclude that in any case, the outputs are distributed identically for the systems $\mathsf{snd}^{\mathsf{A}}_\lambda\mathsf{rcv}^{\mathsf{B}}_\lambda[\mathbf{URI}, \mathbf{IC}]$ and $\mathsf{sim}^{\mathsf{E}}_{\mathrm{RIC}}\mathbf{RIC}_\lambda$.

On the ith input $(\mathsf{deliver})$ **at interface** E**:** In $\mathsf{snd}^{\mathsf{A}}_\lambda\mathsf{rcv}^{\mathsf{B}}_\lambda[\mathbf{URI}, \mathbf{IC}]$, the front element of the sender queue (within $\mathbf{IC}$) is decrypted by rcv_λ. This behavior is perfectly simulated in $\mathsf{sim}^{\mathsf{E}}_{\mathrm{RIC}}\mathbf{RIC}_\lambda$ since the simulator retrieves the front element (E, C) of its queue $\mathcal{Q}_1$ which simulates the real-world sender queue. Next, the simulator executes the same instructions as for (inject, E, C) which lets the two systems produce identically distributed outputs.

This concludes the analysis of the security condition.

We now prove the availability condition (1). First, in $\mathsf{snd}^{\mathsf{A}}_{v,\lambda}\mathsf{rcv}^{\mathsf{B}}_{v,\lambda}\mathsf{dlv}^{\mathsf{E}}[\mathbf{URI}, \mathbf{IC}]$, the converter dlv is attached at interface E and answers any output produced by $\mathbf{IC}$ with the input $(\mathsf{deliver})$. This essentially converts $\mathbf{IC}$ into a reliable transmission channel: whatever pair (A, C) is given to $\mathbf{IC}$, it is immediately delivered to $\mathsf{rcv}_{v,\lambda}$. Therefore, if the ith input at interface A is $(\mathsf{send}, A_i, M_i)$ then the ith output at interface B is (A_i, M_i), since $\mathbf{URI}$ is queried with the exact same parameters. It is obvious that the same holds for the input-output behavior of system $\mathsf{dlv}^{\mathsf{E}}\mathbf{RIC}_\lambda$. □

4.2 What is Best-Possible Security?

We observe that $\mathbf{RIC}_\lambda$ has two undesirable properties: messages can be injected and the output at interface E leaks more than only the length of the payload in that it reveals whether Alice sends the pair (A, M) that has been output by Bob upon an adversarial injection. In contrast, a channel that only leaks the message length is considered fully confidential.

We first illustrate an application in which this lack of full confidentiality is problematic. The main purpose of our example is to show that one cannot exclude

the existence of an application where exactly this (intuitively small) difference to full confidentiality yields a security problem.

Second, we show that a successful injection followed by the undesired information leakage about the repetition is possible for any scheme, even if it is stateful and uses an arbitrary setup before starting communication, and that the probability of this is minimized if $\mathbf{RIC}_\lambda$ is used.

Sample scenario: On the difference to full confidentiality. Assume a setting in which party A is allowed to send information to party B via a fully confidential channel but not vice versa. Suppose now that B finds a possibility to send information to A via a covert channel and the two parties use the confidential channel for messages from A to B and the covert channel for messages from B to A. Suppose now that the confidential channel is in fact a channel that leaks the above repetition event instead of only the message length. This gives an investigator E a means to test for the existence of a covert channel from B to A as follows: At some point, E injects a random message M to B. Assuming information flow from B to A, party B might start a discussion about M with party A. As part of this conversation, A might send M to B, which would signal a repetition-event to E. For large message spaces, it is very unlikely that A comes up with the exact same message that was randomly injected to B before, unless there is a (hidden) flow of information. The occurrence of the event is therefore a witness for the existence of a channel from B to A. In contrast, a fully confidential channel would not reveal the existence of the covert channel.

RIC provides best-possible security. In $\mathbf{RIC}_\lambda$, an injection attempt is successful with probability at most $\frac{|\mathcal{M}|}{|\mathcal{C}|} = |\Sigma|^{-\lambda}$ and given a successful injection and that Alice subsequently sends the corresponding output of Bob, the above described leakage occurs with probability 1. Overall, the total probability that an undesired property can be observed is bounded by $|\Sigma|^{-\lambda}$.

We show that this probability is optimal and that no protocol can achieve a better bound. Hence, $\mathbf{RIC}_\lambda$ maximizes authenticity and confidentiality. We first prove the following general lemma.

Lemma 2. *Let $\mathcal{M}$ and $\mathcal{C}$ be finite nonempty sets and let E and D be random variables over functions $\mathcal{A} \times \mathcal{M} \to \mathcal{C}$ and $\mathcal{A} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$, respectively, such that*

$$\forall m \in \mathcal{M}, a \in \mathcal{A}: \quad \mathsf{Pr}[D(a, E(a, m)) = m] \geq p$$

for some $p \in [0, 1]$. We then have for all $a \in \mathcal{A}$ and any random variable C that is distributed uniformly over $\mathcal{C}$ and independent from E and D,

$$\mathsf{Pr}[D(a, C) \neq \perp \wedge E(a, D(a, C)) = C] \geq p \cdot \frac{|\mathcal{M}|}{|\mathcal{C}|}.$$

Proof. We have for all $a \in \mathcal{A}$

$$\begin{aligned}
&\mathsf{Pr}[D(a,C) \neq \perp \wedge E(a,D(a,C)) = C] \\
&\qquad = \sum_{m \in \mathcal{M}} \sum_{c \in \mathcal{C}} \mathsf{Pr}[D(a,c) = m \wedge E(a,m) = c \wedge C = c] \\
&\qquad = \frac{1}{|\mathcal{C}|} \sum_{m \in \mathcal{M}} \underbrace{\sum_{c \in \mathcal{C}} \mathsf{Pr}[D(a,c) = m \wedge E(a,m) = c]}_{= \mathsf{Pr}[D(a,E(a,m)) = m] \;\geq\; p} \\
&\qquad \geq \; p \cdot \frac{|\mathcal{M}|}{|\mathcal{C}|},
\end{aligned}$$

where we used in the second step that C is distributed uniformly over $\mathcal{C}$ and independent from E and D. □

Lemma 2 can be applied to our usual setting $\mathsf{enc}_\lambda^{\mathsf{A}} \mathsf{dec}_\lambda^{\mathsf{B}}[\mathbf{SK}_{\mathcal{K}}, \mathbf{IC}]$ for a generic protocol $(\mathsf{enc}_\lambda, \mathsf{dec}_\lambda)$ in a straightforward manner: we only have to observe that for the ith input $(\mathsf{send}, A_i, M_i)$, for all $i \in \mathbb{N}$, converter enc_λ is characterized by a probabilistic map $\mathcal{A} \times \mathcal{M} \to \mathcal{C}$, that may depend on previous inputs and outputs and on the key k. Similarly, the converter dec_λ is characterized by a probabilistic map $\mathcal{A} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$ for any $i \in \mathbb{N}$.

Correctness of the protocol implies that if $(\mathsf{send}, A_i, M_i)$ is input to enc_λ as the ith query and yields ciphertext C_i, then the probability that on the ith input (A_i, C_i) to dec_λ, and if dec_λ has not halted yet, the converter decrypts the ciphertext to M_i with probability p; note that $p = 1$ for RAE schemes. Hence, Lemma 2 implies that the probability that any of the two undesirable properties can be observed during protocol execution is at least $|\Sigma|^{-\lambda}$.

5 Augmented Secure Channels and Verifiable Redundancy

Looking at the specification of $\mathbf{RIC}_\lambda$, we observe that for large values λ, the probability of successful injections becomes exponentially small, and so are the repetition events at interface E. We are particularly interested in the resource that specifies this abstraction: a channel that allows to securely transmit messages consisting of an associated data and a payload part such that an attacker is limited to seeing the associated data and the length of the payload, to deliver the next message, or to abort the whole communication. This channel abstraction corresponds to an augmented secure channel. Such channels were introduced in [1] and shown to be achievable by the AEAD notion of [12]. Not surprisingly, this confirms that RAE and AEAD achieve the same security goals for large ciphertext expansion.

Additionally, we formally show how redundancy in messages can be exploited to improve authenticity, where redundancy restricts the set of valid messages to a subset of $\mathcal{M} = \Sigma^*$.

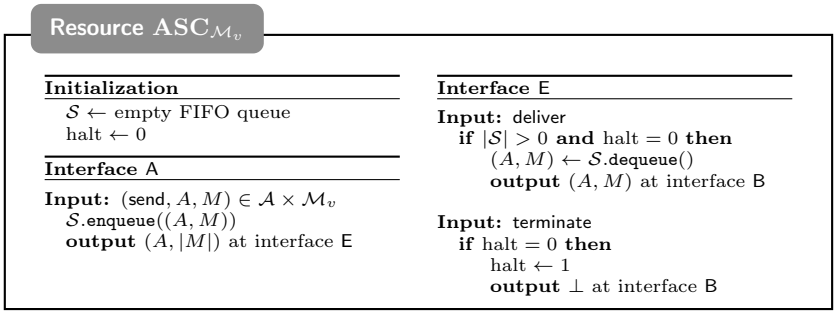

Fig. 7. Description of **ASC**, an augmented secure channel.

The following theorem provides the exact security bound in terms of redundancy in the message space and ciphertext expansion λ. We thereby confirm a conjecture of [6]. Let $v : \mathcal{M} \mapsto \{\mathsf{true}, \mathsf{false}\}$ be a predicate on the message space. We define the subset $\mathcal{M}_v \coloneqq \{M \mid M \in \mathcal{M} \ \wedge \ v(M)\}$ which we call the set of valid messages. Following [6], the density of $\mathcal{M}_v$ is defined as

$$d_v \coloneqq \max_{\ell \in \mathbb{N}} \frac{|\Sigma^\ell \cap \mathcal{M}_v|}{|\Sigma^\ell|}.$$

A similar statement, namely for a specific encryption mode that allows the attacker to either forward messages or to inject purely random ones into the channel, and for a specific redundant encoding, has been shown by Rüedlinger [15].

The constructed channel. The augmented secure channel $\mathbf{ASC}_{\mathcal{M}_v}$ is described in Fig. 7. The channel is derived from $\mathbf{RIC}_\lambda$ by requiring that $M \in \mathcal{M}_v$ and by removing undesired capabilities that vanish due to the exponentially small success probability for large λ.

Protocol. The protocol for the sender, sndChk_v, accepts inputs of the form (send, A, M) at its outer interface and forwards the pair to the channel $\mathbf{RIC}_\lambda$ if and only if $v(M)$ (and otherwise ignores the request). The receiver converter $\mathsf{rcvChk_v}$, on receiving the pair (A, M) from $\mathbf{RIC}_\lambda$, outputs (A, M) at its outer interface if and only if $v(M)$. If $\mathsf{rcvChk_v}$ receives $\perp$ from $\mathbf{RIC}_\lambda$ or if $\neg v(M)$, it outputs $\perp$ at its outer interface and halts.

Theorem 2. *Let $\lambda \in \mathbb{N}$. The protocol $(\mathsf{sndChk}_v, \mathsf{rcvChk_v})$ constructs $\mathbf{ASC}_{\mathcal{M}_v}$ from $\mathbf{RIC}_\lambda$ with respect to $(\mathsf{dlv}, \mathsf{dlv})$ and simulator $\mathsf{sim}_{\mathrm{ASC}}$ defined in Fig. 8. More specifically, for all distinguishers* $\mathbf{D}$

$$\Delta^{\mathbf{D}}\left(\mathsf{sndChk}_v^{\mathsf{A}}\mathsf{rcvChk_v}^{\mathsf{B}}\mathsf{dlv}^{\mathsf{E}}\mathbf{RIC}_\lambda, \mathsf{dlv}^{\mathsf{E}}\mathbf{ASC}_{\mathcal{M}_v}\right) = 0 \tag{3}$$

and

$$\Delta^{\mathbf{D}}\left(\mathsf{sndChk}_v^{\mathsf{A}}\mathsf{rcvChk_v}^{\mathsf{B}}\mathbf{RIC}_\lambda, \mathsf{sim}_{\mathrm{ASC}}^{\mathsf{E}}\mathbf{ASC}_{\mathcal{M}_v}\right) \leq d_v \cdot |\Sigma|^{-\lambda}. \tag{4}$$

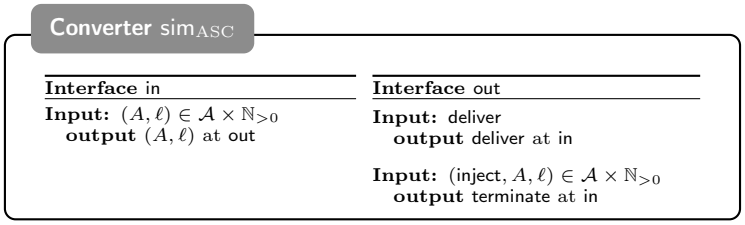

Fig. 8. Simulator for the security condition of the construction of $\mathbf{ASC}_{\mathcal{M}_v}$.

Proof. The availability condition (3) is straightforward to verify. We only prove the security condition (4). It is easy to see that the two systems behave identically as long as no injection attempt is successful. This is because successful injections are necessary for observing repeat: as long as no injection is successful, for any send-query to $\mathbf{RIC}_\lambda$, the condition $i \leq |\mathcal{R}|$ is not satisfied after incrementing i. We thus only have to bound this probability. We hence consider the event that in an interaction of a distinguisher with the real system $\mathsf{sndChk}_v^{\mathsf{A}}\mathsf{rcvChk}_{\mathsf{v}}{}^{\mathsf{B}}\mathbf{RIC}_\lambda$ the first attempt to inject a random message is successful (since in case of an unsuccessful attempt, both channels stop delivering messages). In any interaction of $\mathbf{D}$ with the resource, the probability of the event is determined by $\mathbf{RIC}_\lambda$ as one out of two possibilities, see Fig. 4. For any $i \in \mathbb{N}$, if the ith query at interface E is the first attempt to inject a message, then the probability depends on whether the specified associated data and the length coincides with the length of the message and the associated data of the ith input at interface A. Both probabilities are upper bounded by

$$\max\left\{\frac{\left|\Sigma^{|C_i|-\lambda}\cap\mathcal{M}_v\right|-1}{\left|\Sigma^{|C_i|}\right|-1}, \frac{\left|\Sigma^{|C_i|-\lambda}\cap\mathcal{M}_v\right|}{\left|\Sigma^{|C_i|}\right|}\right\} \leq \frac{\left|\Sigma^{|C_i|-\lambda}\cap\mathcal{M}_v\right|}{\left|\Sigma^{|C_i|}\right|}$$
$$= \frac{\left|\Sigma^{|C_i|-\lambda}\cap\mathcal{M}_v\right|}{\left|\Sigma^{|C_i|-\lambda}\right|}\cdot|\Sigma|^{-\lambda} \leq d_v\cdot|\Sigma|^{-\lambda},$$

where we used $\frac{x-1}{y-1} \leq \frac{x}{y}$ for $x \leq y$ in the first step, and the definition of d_v in the last step. □

6 Guarantees for Nonce-Reuse

One goal of robust authenticated encryption is to provide resilience to the misuse when nonces are repeated. While the expected security loss was only informally stated in [6], we rigorously derive the exact guarantees that can still be expected in such a scenario. To this end, we consider the extreme case where the nonce is a constant value.

Repetition ASC. The channel that is achieved if the nonce is repeating is denoted $\mathbf{RASC}_{\mathcal{M}_v}$ and its description is given in Fig. 9. There are two differences

to $\mathbf{ASC}_{\mathcal{M}_v}$: First, not only the length of the message is leaked at interface E but also the number i of the first transmitted message that equals the current message. This leaks the repetition pattern of transmitted values. Second, the adversary can replay messages and induce arbitrary out-of-order delivery.

Protocol. The protocol, which we denote by $(\mathsf{rsnd}, \mathsf{rrcv})$, invokes **URI** using the constant nonce 0. Furthermore, the protocol verifies that all messages are from the set $\mathcal{M}_v$. The protocol is specified in Fig. 10.

Theorem 3. *Let $\lambda \in \mathbb{N}$. The protocol $(\mathsf{rsnd}_\lambda, \mathsf{rrcv}_\lambda)$ constructs $\mathbf{RASC}_{\mathcal{M}_v}$ from $[\mathbf{URI}, \mathbf{IC}]$ with respect to $(\mathsf{dlv}, \mathsf{dlv})$ and simulator $\mathsf{sim}_{\mathrm{RASC}}$ defined in Fig. 11. More specifically, we have for all distinguishers $\mathbf{D}$*

$$\Delta^{\mathbf{D}}\left(\mathsf{rsnd}_\lambda^{\mathsf{A}}\mathsf{rrcv}_\lambda^{\mathsf{B}}\mathsf{dlv}^{\mathsf{E}}[\mathbf{URI}, \mathbf{IC}], \mathsf{dlv}^{\mathsf{E}}\mathbf{RASC}_{\mathcal{M}_v}\right) \leq \frac{2d_v + q \cdot (q-1)}{2} \cdot |\Sigma|^{-\lambda} \quad (5)$$

and

$$\Delta^{\mathbf{D}}\left(\mathsf{rsnd}_\lambda^{\mathsf{A}}\mathsf{rrcv}_\lambda^{\mathsf{B}}[\mathbf{URI}, \mathbf{IC}], \mathsf{sim}_{\mathrm{RASC}}^{\mathsf{E}}\mathbf{RASC}_{\mathcal{M}_v}\right) \leq \frac{2d_v + q \cdot (q-1)}{2} \cdot |\Sigma|^{-\lambda}, \quad (6)$$

where q is the total number of inputs made by $\mathbf{D}$.

Proof. We prove the security condition (6) in a similar way as in the previous section.

On the ith input $(\mathsf{send}, A_i, M_i)$ at interface A: In $\mathsf{rsnd}_\lambda^{\mathsf{A}}\mathsf{rrcv}_\lambda^{\mathsf{B}}[\mathbf{URI}, \mathbf{IC}]$, system **URI** is queried with $(\mathsf{fun}, 0, A_i, \lambda, M_i)$ to produce ciphertext C_i. Next, the pair (A_i, C_i) is sent over **IC** and thus output at interface E. Clearly, if there exists some $j < i$ s.t. $A_j = A_i$ and $M_j = M_i$ then $C_j = C_i$ (with probability 1). On the other hand, if $A_j = A_i$ but $M_j \neq M_i$, then $C_i \neq C_j$ with probability 1. Otherwise, the distribution of ciphertext C_i is independent of the distribution of any C_j that is encrypted with different parameters $A_j \neq A_i$ or $|M_j| \neq |M_i|$ (by definition of **URI**).

In system $\mathsf{sim}_{\mathrm{RASC}}^{\mathsf{E}}\mathbf{RASC}_{\mathcal{M}_v}$, the simulator gets the pair (ℓ, k), where $\ell = |M_i|$ and k is the number of the first input that equals (A_i, M_i). This allows $\mathsf{sim}_{\mathrm{RASC}}$ to consistently reflect repetitions of ciphertexts. If $k = i$, i.e., if the message is new, a uniformly random ciphertext C_i of length $\ell + \lambda$ is output and stored for future reference. By comparing the behavior to $\mathsf{rsnd}_\lambda^{\mathsf{A}}\mathsf{rrcv}_\lambda^{\mathsf{B}}[\mathbf{URI}, \mathbf{IC}]$, we observe that the two systems have the same output distribution, as long as there is no collision to a previous ciphertext C_j that has been generated on input (A_j, M_j) with $A_j = A_i$ and $|M_i| = |M_j|$ but $M_i \neq M_j$. The probability of such a collision is bounded by

$$\frac{q \cdot (q-1)}{2} \cdot |\Sigma|^{-\lambda}. \quad (7)$$

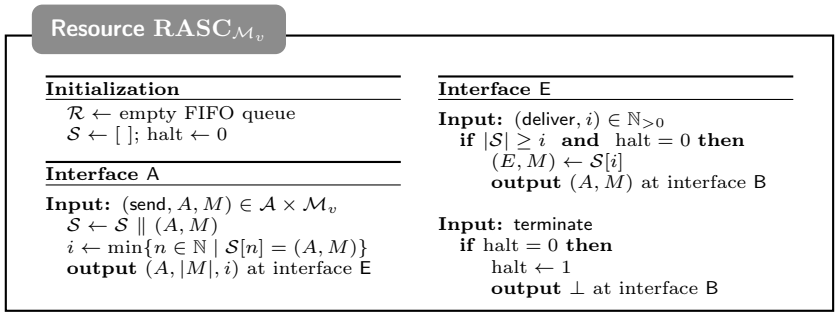

Fig. 9. Description of **RASC**.

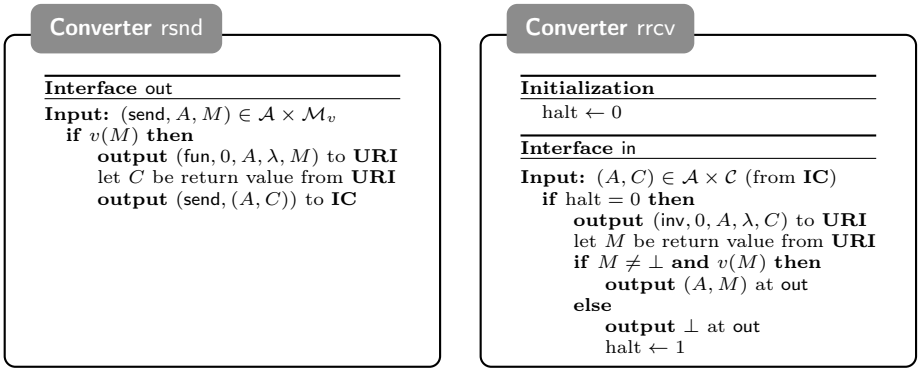

Fig. 10. The converters for the sender (left) and the receiver (right).

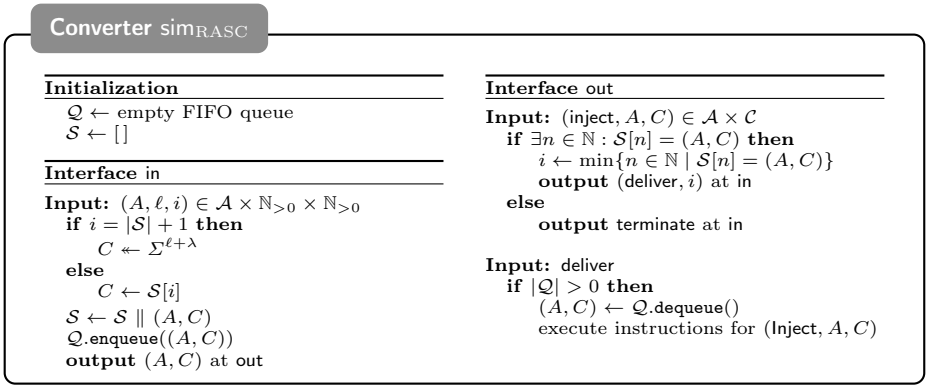

Fig. 11. Simulator for the security condition of the construction of $\mathbf{RASC}_{\mathcal{M}_v}$.

On the ith input $(\mathsf{inject}, A_i, C_i)$ **at interface** E**:** In the following, we assume that interface B has not output $\perp$ already, as otherwise the two systems behave identically anyway.

In $\mathsf{rsnd}^{\mathsf{A}}_{\lambda}\mathsf{rrcv}^{\mathsf{B}}_{\lambda}[\mathbf{URI}, \mathbf{IC}]$, rsnd queries **URI** with input $(\mathsf{inv}, 0, A_i, \lambda, C_i)$. In the following analysis, Let $\ell \coloneqq |C_i| - \lambda$ and let denote $q_{\mathsf{A}}^{A_i,\ell}$ the total number of distinct queries (send, A_i, M) with $\ell = |M|$ at interface A.

When rrcv decrypts ciphertext C_i of length $\ell + \lambda$ using associated data A_i, then there are two cases to consider: either **URI** has already generated a function mapping messages of length ℓ to ciphertexts of length $\ell + \lambda$ or **URI** has not yet generated such a function yet.

1. If **URI** has generated an injective function already, then there have been $q_{\mathsf{A}}^{A_i,\ell}$ fixed message-ciphertext pairs (M'_j, C'_j) generated by **URI**. Clearly, if C_i equals one of the C'_j, then the output at interface B is M'_j. Otherwise, there is only an output at interface B if C_i has a preimage under the random injection and since there are $|\Sigma^\ell \cap \mathcal{M}_v| - q_{\mathsf{A}}^{A_i,\ell}$ possible valid preimages left and each one is assigned to one of $|\Sigma|^{\ell+\lambda} - q_{\mathsf{A}}^{A_i,\ell}$ ciphertexts, the probability that C_i has a preimage (and consequently that B produces an output) is at most

$$\frac{|\Sigma^\ell \cap \mathcal{M}_v| - q_{\mathsf{A}}^{A_i,\ell}}{|\Sigma|^{\ell+\lambda} - q_{\mathsf{A}}^{A_i,\ell}} \leq \frac{|\Sigma^\ell \cap \mathcal{M}_v|}{|\Sigma|^{\ell+\lambda}} \leq d_v \cdot |\Sigma|^{-\lambda}, \tag{8}$$

   where we used $\frac{x-d}{y-d} \leq \frac{x}{y}$ for $x \leq y$ and $0 \leq d < y$ in the first step and the definition of d_v in the second step.
2. The second case follows trivially by observing that if **URI** has not generated a function yet, then $q_{\mathsf{A}}^{A_i,\ell} = 0$.

In $\mathsf{sim}^{\mathsf{E}}_{\mathrm{RASC}}\mathbf{RASC}_{\mathcal{M}_v}$, if the injected pair (A_i, C_i) is equal to a previously generated pair (E_j, C_j), the simulator outputs $(\mathsf{deliver}, j)$ at the inner interface to instruct the channel to replay the jth message, which generates the output (A_j, M_j) at interface B.

On the other hand, if the pair (A_i, C_i) is new, $\mathsf{sim}_{\mathrm{RASC}}$ terminates the channel. This element is guaranteed to decrypt to $\perp$ on any decryption attempt at interface B.

System $\mathsf{sim}^{\mathsf{E}}_{\mathrm{RASC}}\mathbf{RASC}_{\mathcal{M}_v}$ reflects the behavior of $\mathsf{snd}^{\mathsf{A}}_{v,\lambda}\mathsf{rcv}^{\mathsf{B}}_{v,\lambda}[\mathbf{URI}, \mathbf{IC}]$ as long as the converter rrcv_λ does not generate a valid output upon its first attempt to decrypt an injected associated data-ciphertext pair (A, C) that has not been generated in reaction to a send-query at interface A. In such a case, $\mathsf{sim}_{\mathrm{RASC}}$ would always insert an empty element and provoke the output $\perp$. However, the probability of rrcv_λ producing a valid output upon decryption of a new associated data-ciphertext pair is bounded in (8).

On the ith input $(\mathsf{deliver})$ **at interface** E**:** In $\mathsf{rsnd}^{\mathsf{A}}_{\lambda}\mathsf{rrcv}^{\mathsf{B}}_{\lambda}[\mathbf{URI}, \mathbf{IC}]$, the front element, say (A, C) of the sender queue (within **IC**) is output to rrcv that tries to decrypt the associated data-ciphertext pair. Furthermore, since the pair is a valid encryption, the message is output at interface B.

In $\mathsf{sim}^{\mathsf{E}}_{\mathrm{RASC}}\mathbf{RASC}_{\mathcal{M}_v}$, the simulator retrieves the front element (A, C) of queue $\mathcal{Q}$ which simulates the sender queue of the real execution of the protocol.

The simulator executes the same instructions as for (inject, A, C) which yields the same behavior for both systems, since also in this case, the pair (A, C) is in the simulator's list of valid associated data-ciphertext pairs.

This concludes the analysis and we can bound the distinguishing advantage by using inequalities (7) and (8). The analysis of the availability condition (5) is straightforward and omitted. □

Acknowledgments. Ueli Maurer was supported by the Swiss National Science Foundation (SNF), project no. 200020-132794. Björn Tackmann was supported by the Swiss National Science Foundation (SNF) via Fellowship no. P2EZP2_155566 and the NSF grants CNS-1228890 and CNS-1116800. Much of the work on this paper was done while Phil Rogaway was visiting Ueli Maurer's group at ETH Zurich. Many thanks to Ueli for hosting that sabbatical. Rogaway was also supported by NSF grants CNS-1228828 and CNS-1314885.

A Equivalence of Security Definitions for RAE

The security of an RAE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined in [6] via the reference games $\mathbf{REAL}_\Pi$ and $\mathbf{RAE}_\Pi$ depicted in Fig. 12. The game $\mathbf{REAL}_\Pi$ provides oracle access to $\mathcal{E}$ and $\mathcal{D}$, and $\mathbf{RAE}_\Pi$ provides oracle access to ideal uniform random injections and their inverses. The advantage of an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mathrm{rae,\,game}}_{\Pi}(\mathcal{A}) := \mathsf{Pr}\left[\mathcal{A}^{\mathbf{REAL}_\Pi} = 1\right] - \mathsf{Pr}\left[\mathcal{A}^{\mathbf{RAE}_\Pi} = 1\right].$$

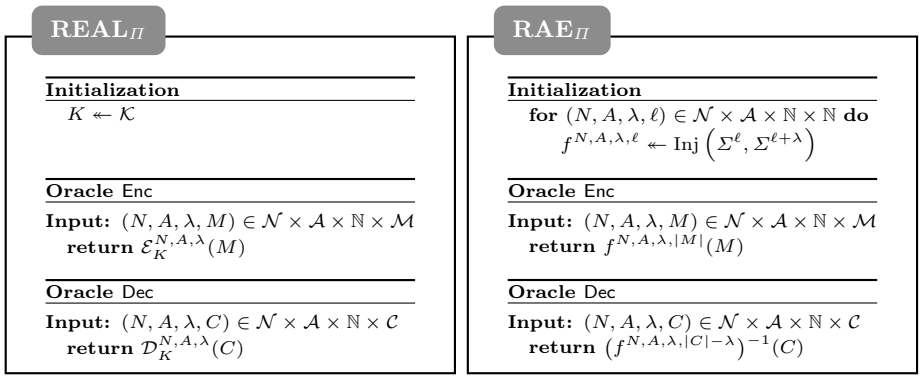

Fig. 12. Security games for the RAE scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ as defined in [6].

The following lemma implies that this definition is equivalent to Definition 4.

Lemma 3. *For every distinguisher* $\mathbf{D}$ *there is an adversary* $\mathcal{A}$ *(with essentially the same efficiency) such that* $\mathbf{Adv}^{\mathrm{rae,\,game}}_{\Pi}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{rae}}_{\Pi}(\mathbf{D})$*. Conversely, for every adversary* $\mathcal{A}$ *there is a distinguisher* $\mathbf{D}$ *(with essentially the same efficiency) such that* $\mathbf{Adv}^{\mathrm{rae}}_{\Pi}(\mathbf{D}) = \mathbf{Adv}^{\mathrm{rae,\,game}}_{\Pi}(\mathcal{A})$*.*

Proof. Observe that inputs $(\mathsf{fun}, \cdot, \cdot, \cdot, \cdot)$ and $(\mathsf{inv}, \cdot, \cdot, \cdot, \cdot)$ to $\mathsf{rae}_{\Pi}{}^{\mathsf{A}}\,\mathsf{rae}_{\Pi}{}^{\mathsf{B}}\,\mathbf{SK}_{\mathcal{K}}$ correspond to queries in $\mathbf{REAL}_{\Pi}$ to the oracles Enc and Dec, respectively: in both cases, the resulting outputs are generated by the algorithms $\mathcal{E}$ and $\mathcal{D}$ for a uniformly random key. Similarly, the same inputs to the resource **URI** correspond to the oracles Enc and Dec in the game $\mathbf{RAE}_{\Pi}$ since the outputs are in both cases computed by a uniformly random injection or its inverse. Hence, a distinguisher can be turned into an adversary with the same advantage and vice versa by exchanging inputs to the resources with the corresponding oracle queries. □

References

1. Badertscher, C., Matt, C., Maurer, U., Rogaway, P., Tackmann, B.: Augmented Secure Channels as the Goal of the TLS 1.3 Record Layer. Cryptology ePrint Archive, Report 2015/394, 2015
2. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Advances in Cryptology – ASIACRYPT 2000, pp. 531–545. Springer Berlin Heidelberg (2000)
3. Bellare, M., Rogaway, P.: Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In: Advances in Cryptology – ASIACRYPT 2000, pp. 317–330. Springer (2000)
4. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: Fast Software Encryption, pp. 389–407. Springer (2004)
5. Gligor, V., Donescu, P.: Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In: Fast Software Encryption, pp. 92–108. Springer (2002)
6. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption: AEZ and the problem that it solves. In: Advances in Cryptology – EUROCRYPT 2015, pp. 15–44. Springer (2015)
7. Jutla, C.: Encryption modes with almost free message integrity. In: Advances in Cryptology – EUROCRYPT 2001, pp. 529–544. Springer (2001)
8. Katz, J., Yung, M.: Unforgeable encryption and chosen ciphertext secure modes of operation. In: Fast Software Encryption, pp. 284–299. Springer (2001)
9. Maurer, U.: Constructive cryptography – a new paradigm for security definitions and proofs. In: Mödersheim, S., Palamidessi, C. (eds.) Theory of Security and Applications, Lecture Notes in Computer Science, vol. 6993, pp. 33–56. Springer (2012)
10. Maurer, U., Renner, R.: Abstract cryptography. In: Chazelle, B. (ed.) The Second Symposium on Innovations in Computer Science, ICS 2011, pp. 1–21. Tsinghua University Press (2011)
11. Maurer, U., Rüedlinger, A., Tackmann, B.: Confidentiality and integrity: a constructive perspective. In: TCC 2012, pp. 209–229. Springer (2012)
12. Rogaway, P.: Authenticated-encryption with associated-data. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 98–107. ACM (2002)
13. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Transactions on Information and System Security (TISSEC) 6(3), 365–403 (2003)
14. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Advances in Cryptology – EUROCRYPT 2006, pp. 373–390. Springer (2006)

15. Rüedlinger, A.: Restricted types of malleability in encryption schemes. Masters thesis, ETH Zürich (2011)