An extended abstract of this paper is published in the proceedings of the 13th Privacy Enhancing Technologies Symposium — PETS 2013. This is the full version.

Anonymity-preserving Public-Key Encryption: A Constructive Approach

Markulf Kohlweiss[1], Ueli Maurer[2], Cristina Onete[3], Björn Tackmann[2], and Daniele Venturi[4]

[1] *Microsoft Cambridge*
[2] *ETH Zürich*
[3] *Technische Universität Darmstadt*
[4] *Aarhus University*

Abstract.

A receiver-anonymous channel allows a sender to send a message to a receiver without an adversary learning for whom the message is intended. Wireless broadcast channels naturally provide receiver anonymity, as does multi-casting one message to a receiver population containing the intended receiver. While anonymity and confidentiality appear to be orthogonal properties, making anonymous communication confidential is more involved than one might expect, since the ciphertext might reveal which public key has been used to encrypt. To address this problem, public-key cryptosystems with enhanced security properties have been proposed.

This paper investigates constructions as well as limitations for preserving receiver anonymity when using public-key encryption (PKE). We use the constructive cryptography approach by Maurer and Renner and interpret cryptographic schemes as constructions of a certain ideal resource (e.g. a confidential anonymous channel) from given real resources (e.g. a broadcast channel). We define appropriate anonymous communication resources and show that a very natural resource can be constructed by using a PKE scheme which fulfills three properties that appear in cryptographic literature (IND-CCA, key-privacy, weak robustness). We also show that a desirable stronger variant, preventing the adversary from selective "trial-deliveries" of messages, is unfortunately unachievable by any PKE scheme, no matter how strong. The constructive approach makes the guarantees achieved by applying a cryptographic scheme explicit in the constructed (ideal) resource; this specifies the exact requirements for the applicability of a cryptographic scheme in a given context. It also allows to decide which of the existing security properties of such a cryptographic scheme are adequate for the considered scenario, and which are too weak or too strong. Here, we show that weak robustness is necessary but that so-called strong robustness is unnecessarily strong in that it does not construct a (natural) stronger resource.

Keywords: public-key encryption, key privacy, robust encryption, anonymity, constructive cryptography

Contents

1 Introduction

Protocols and other mechanisms for protecting privacy often use cryptographic schemes in non-standard ways, sometimes requiring such schemes to have cryptographic properties that go beyond data authenticity and confidentiality. It is important that new cryptographic schemes take these requirements into account and that designers of privacy protocols are aware which cryptographic properties are needed in which situation. Several types of cryptographic schemes have been investigated with a focus on anonymity. In "key-private" public-key encryption, the ciphertext does not reveal information about the intended receiver [BBDP01, ABN10], in private key exchange [Aba02, CK02, AF04] two parties can exchange a key without revealing their identities, and "anonymous" signatures protect the signer's identity at least as long as parts of the signed plaintext remain hidden [YWDW06, Fis07]. In this paper, we focus on public-key encryption and receiver anonymity.

The cryptographic community traditionally defines security notions for cryptographic schemes such as encryption from a game-based perspective, i.e., one defines properties of schemes by means of theoretical experiments, usually referred to as games. Though game-based definitions are frequently used in cryptography and have been studied and improved over the years, they have two major shortcomings. First, they model the use of a scheme abstractly and simplify it to the interaction between an adversary and a so-called challenger (both in a sense artificial). Consequently, the system-level security guarantees that one obtains by applying a provably secure scheme in a specific context are usually not evident. Second, if an (encryption) scheme proven secure in this way is used in a larger protocol, then the security of the full protocol can only be proven by showing a reduction of breaking a certain security property of the underlying (encryption) scheme to breaking the security of the protocol. Each such protocol requires in principle its own tailor-made security reduction.

A fundamentally different approach to defining the security of cryptographic schemes has been proposed by Maurer and Renner [MR11]. Following their *constructive cryptography* paradigm, one models both the resources assumed by a protocol or scheme and the desired functionality explicitly, and the goal of the protocol is to *construct* (in a well-defined sense) the desired resource from the assumed resources. For a public-key encryption scheme, for instance, this means that one assumes an authenticated communication channel from the receiver to the sender to transmit the public key, as well as an insecure channel from the sender to the receiver to transmit the ciphertext. The goal of the scheme is to construct, from the assumed channels, a confidential communication channel (from sender to receiver, cf. [MS96]). The assumed channels can either be physically realized or can themselves be constructed cryptographically, and the resulting channel can directly be used in any application that requires such a channel. Furthermore, as the constructive approach makes the guarantees of both the assumed and the constructed resources explicit, it allows to capture the *exact* cryptographic assumptions required for security.

Anonymity in constructive cryptography. In constructive cryptography, a network is modeled as a resource that can be accessed by multiple (honest or dishonest) parties. The parties access the resource through interfaces provided by the resource and specific to each party; the interfaces specify exactly in which way each party can access the resource. In this context, anonymity is an explicit guarantee of such a resource (e.g., a network). In particular, as adversarial interaction with the network is also modeled by means of an interface (the attacker is a dishonest party), the security (or privacy) guarantees of the underlying network are described by the (absence of) capabilities of such an adversary.

For the particular case of receiver-anonymous communication, we model a network resource

with multiple receiver interfaces. Whenever a sender inputs a message at its interface and chooses a receiver, the network resource might leak certain information (such as the length of the message or even the complete plaintext) at the adversary's interface; however, the resource will *not* reveal the receiver. The goal of a public-key encryption scheme in this context is to construct a resource that will hide the receiver, while leaking no information on the message contents apart from (potentially) the length.

We consider the case where encryption schemes are used for end-to-end encryption between senders and receivers. For such protocols, anonymity cannot be created by employing a cryptographic primitive. In fact, a constructive approach makes it apparent that schemes can only *preserve* anonymity that is guaranteed by the underlying network, but never *produce* it.[1] If Alice sends a message to Bob over the Internet using Bob's publicly known IP address, then there is no hope for the encryption scheme (or key exchange protocol) to hide the fact that Bob is the intended receiver of Alice's message. In fact, encrypting messages potentially makes the problem worse: Even if the transmission of the ciphertext is itself anonymous, the ciphertext might still reveal under which public key it was encrypted.

Hence, in a constructive analysis of the end-to-end use of cryptographic schemes, we always consider the *preservation* of anonymity. If the underlying network (one of the initial resources) is insecure, but guarantees a certain level of anonymity, then an "anonymity-preserving" scheme will improve the security while maintaining as much anonymity as possible. The obtained guarantees are strong in that they hold *regardless of the context*, that is, of any prior knowledge the adversary might have and of any protocols that are executed in parallel.

Our contributions. We provide a treatment of receiver anonymity in the context of public-key encryption schemes from the perspective of constructive cryptography. In particular, we show how anonymity is described as a feature of a communication resource, and we prove which security properties of the underlying encryption scheme are necessary and/or sufficient to achieve a confidential receiver-anonymous communication resource from a non-confidential, but also receiver-anonymous one. (Schemes with these properties are known in the literature.) More specifically we consider the following network resources (specified in more detail in Section 3.1):

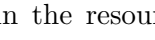
- The insecure broadcast network $\prec$,[2] allowing a single sender to broadcast messages to multiple receivers, and allowing the adversary to learn the entire message and to remove, change, or inject messages;
- The confidential receiver-anonymous channel $\multimap\!\!\diamond\!\!\prec\bullet$, which preserves both the confidentiality of the message and the anonymity of the intended receiver, leaking only the length of the message and allowing the adversary only to delete and honestly deliver messages, and to inject different messages to chosen recipients.

We show that $\multimap\!\!\diamond\!\!\prec\bullet$ can be constructed from $\prec$ and authenticated channels $\longleftarrow\!\!\bullet$ (in an initial step), by employing a secure (IND-CCA), key-private (IK-CCA), and weakly robust (WROB-CCA) public-key encryption scheme. In fact, we show that constructing $\multimap\!\!\diamond\!\!\prec\bullet$ does *not* require strong robustness (SROB-CCA)—a stricter property for anonymous secure encryption proposed in [ABN10]. Naturally, using SROB-CCA public-key encryption also constructs $\multimap\!\!\diamond\!\!\prec\bullet$; however, this property is not *required*. In other words, the treatment in [ABN10] relies on slightly too strong assumptions. Employing SROB-CCA security, however, *does* yield a tighter security reduction.

[1] Note that this observation does not hold for active networks or overlay networks that can implement their own multi-hop anonymous routing strategy for which encryption is in fact crucial. Buses [BD03] is a cryptographic design exemplifying this, while TOR [DMS04] is the most widely used anonymity system based on this principle.

[2] In naming our resources we extend the $\bullet$-notation of [MS96].

Furthermore, we show that one (the only natural) channel providing stronger anonymity than what we achieve with IND-CCA, IK-CCA, and WROB-CCA encryption *cannot* be achieved by *any* encryption scheme at all (see Section 3.3). In other words, using e.g. the stronger property of SROB-CCA encryption does *not* construct the stronger channel. Note that this does not exclude that SROB-CCA may be a useful property in other scenarios; however, our results indicate that simply improving the properties guaranteed by  in a natural way cannot be done by using SROB-CCA, or any other type of encryption.

Related work. The first definition of key-private public-key encryption appears in [BBDP01]; the goal of the primitive was to attain receiver anonymity. Abdalla et al. [ABN10] noted that also robustness is needed for the PKE scheme to achieve this property, since otherwise an honest receiver is unable to detect whether he is the intended recipient of a given ciphertext and could obtain a bogus decryption. We explicitly describe the guarantees achieved without robustness in the resource  in Section 4.1. Mohassel [Moh10] analyzed game-based security and anonymity notions for KEM-DEM encryption schemes, showing that, for this particular type of composition, weak robustness together with the key privacy of the KEM (key-encapsulation mechanism) and DEM (data-encapsulation mechanism) components is sufficient to obtain a key-private hybrid public-key encryption scheme. Our result implies that weak robustness is sufficient even for universal composition; a *constructive* formulation of KEM-DEM schemes is currently being developed. However, as shown recently by Farshim et al. [FLPQ13] (even strong) robustness is insufficient in certain contexts, such as Sako's auction protocol. The same concept (i.e., that only the intended recipient must be able to decrypt a ciphertext to a meaningful plaintext) lies at the core of *incomparable* public keys in [WFS03].

More general (game-based) frameworks that mix the analysis of cryptographic schemes and traffic-analysis resistance have been proposed in [HM08] and [OV11]. Independently, different cryptographic [CL05, BGKM12] and traffic-analysis models [FJS07, FJS12] have been developed for variants of onion routing. Whereas our work here does not consider traffic analysis explicitly, our in-depth results can be composed with meaningful models of traffic analysis. We discuss implications of our results for traffic analysis in Section 5.

In particular, we note that our confidential receiver-anonymous channel is a simple resource that captures (a form of) receiver anonymity. We note that Pfitzmann and Waidner [PW85] gave an early treatment of several flavors of anonymity across networks, including receiver anonymity. They explicitly considered the idea of using public-key encryption schemes in the realization of receiver-anonymous networks. However, we note that our treatment in this paper gives a more thorough assessment of the notion of anonymity and additionally investigates the properties that are sufficient and necessary to achieve different levels of receiver anonymity. Nagao et al. [NMO08] describe a similar resource for two sender-anonymous channels and show that such channels can be related by reductions to other types of channels, such as secure channels and direction hiding channels. Ishai et al. [IKOS06] provide a broader investigation on how to bootstrap cryptographic functionalities using anonymity. However, it should be noted that the kind of anonymous channels we construct here cannot be used for their purposes. In particular, Ishai et al. assume the existence of a particular type of sender-anonymous channel which is somewhat compatible with the notion of sender-anonymity in the framework of Hevia and Micciancio [HM08] (however, whereas the distinguishing adversary in [HM08] *chooses* the multiset of messages delivered by the senders in the game instantiation, the adversary in [IKOS06] is simply given this set, but has no control over it). By contrast, the resource we construct here provides receiver anonymity, rather than sender anonymity, and we also include confidentiality as a crucial property of our final resource (Ishai et al., however, assume that the adversary is aware of the messages sent in clear by various anonymous senders).

2 Preliminaries

Notation. We use the symbol $\lozenge$ to denote an "error" output of an algorithm. Moreover, for an integer $n \in \mathbb{N}$, we let $[n] \coloneqq \{1, \ldots, n\}$. We generally use typewriter fonts such as `enc` or `dec` to denote algorithms.

2.1 Systems: Resources and Converters, Distinguishers, and Games

We model objects like resources and protocols in terms of *systems*. At the highest level of abstraction—following the hierarchy in [MR11]—systems are objects with *interfaces* by which they connect to (interfaces of) other systems. Each interface is labeled with an element of a given label set. Multiple interfaces can be merged into a single interface; we then call the original interfaces the *sub-interfaces* of the combined interface.

This concept of *abstract systems* captures the topological structures that result when several systems are connected via their interfaces. In the following, we describe the basic types of systems that appear in this work at this level (of abstraction), and we introduce a notation for describing the structure in which multiple such systems are composed.

The abstract systems concept however does not model the behavior of systems, i.e., *how* the systems interact via their interfaces. As statements about cryptographic protocols are statements about behavior, they are formalized at the next (lower) abstraction level. In this respect, all systems in this work are (probabilistic) discrete systems, similar to [Mau02].

Resources and converters. A *resource* for a multi-party setting is a system that provides one interface for each party. In our setting, resources have one interface labeled A for the sender, n interfaces labeled $B_1, \ldots, B_n$ for the n receivers, and one interface labeled E associated with the attacker. Resources are usually denoted either by special symbols such as $\preccurlyeq$ or by bold-face upper-case letters like $\mathbf{R}$ or $\mathbf{S}$. Protocols are formalized as tuples of so-called *converters*, one for each honest party; converters are systems that have two interfaces: one *inside* and one *outside* interface. Standard notations for converters are small Greek letters or special identifiers such as enc or dec; the set of all converters is denoted as Σ. A complete protocol (i.e., a tuple of converters) is denoted by a bold-face Greek letter, such as $\boldsymbol{\pi}$.

Converters can be attached to resources by connecting the inside interface of the converter to one interface of the resource. Notationally, if we attach the inside interface of the converter $\phi \in \Sigma$ to interface I of the resource $\mathbf{R}$, we write $\phi^I\mathbf{R}$. The resulting system $\phi^I\mathbf{R}$ is again a resource which provides all the interfaces of $\mathbf{R}$ (apart from I) as the respective interfaces, and the outside interface of the converter as the I-interface. This operation extends to the case where the interface I is obtained by merging several sub-interfaces. If all (honest) parties use a protocol $\boldsymbol{\pi}$, then all converters that together form $\boldsymbol{\pi}$, one for each (honest) party, are attached to the respective interfaces of the resource. This is then denoted as $\boldsymbol{\pi}\mathbf{R}$.

Multiple resources $\mathbf{R}_1, \ldots, \mathbf{R}_m$ with the same label set $\mathcal{I}$ can be composed in parallel. This is denoted $[\mathbf{R}_1, \ldots, \mathbf{R}_m]$ and is again a resource, such that each interface $I \in \mathcal{I}$ of $[\mathbf{R}_1, \ldots, \mathbf{R}_m]$ allows to access the corresponding interfaces of $\mathbf{R}_1, \ldots, \mathbf{R}_m$.

Distinguishers. A *distinguisher* $\mathbf{D}$ is a special type of system that connects to all interfaces of a resource $\mathbf{U}$ and outputs a single bit at the end of its interaction with $\mathbf{U}$. We write this as the expression $\mathbf{DU}$, which defines a binary random variable. The *distinguishing advantage of a distinguisher* $\mathbf{D}$ *on two systems* $\mathbf{U}$ *and* $\mathbf{V}$ is defined as

$$\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) \coloneqq |\mathsf{P}(\mathbf{DU} = 1) - \mathsf{P}(\mathbf{DV} = 1)|,$$

and we define $\Delta^{\mathcal{D}}(\mathbf{U}, \mathbf{V}) \coloneqq \sup_{\mathbf{D}\in\mathcal{D}} \Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V})$ as the advantage of a class $\mathcal{D}$ of distinguishers. The distinguishing advantage measures how much the output distribution of $\mathbf{D}$ differs when it is connected to either $\mathbf{U}$ or $\mathbf{V}$. Intuitively, if no distinguisher differentiates between $\mathbf{U}$ and $\mathbf{V}$, they can be used interchangeably in any environment (otherwise the environment can serve as a distinguisher).

The distinguishing advantage is a pseudo-metric. In particular, it satisfies the triangle inequality, i.e., $\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{W}) \leq \Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) + \Delta^{\mathbf{D}}(\mathbf{V}, \mathbf{W})$ for all resources $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$, and for all distinguishers $\mathbf{D}$. Two systems are *equivalent*, denoted by $\mathbf{U} \equiv \mathbf{V}$, if they have the same behavior, which is the same as requiring that $\Delta^{\mathbf{D}}(\mathbf{U}, \mathbf{V}) = 0$ for *all* distinguishers $\mathbf{D}$.

The notion of construction. The formalization of constructive security definitions follows the ideal-world/real-world paradigm. The "real world" corresponds to an execution of the protocol $\boldsymbol{\pi}$ in which all honest parties have their converter attached to the assumed resource $\mathbf{R}$; more formally, we consider the *real-world system* $\boldsymbol{\pi}\mathbf{R}$. The "ideal world" corresponds to the constructed resource $\mathbf{S}$ with a simulator σ connected to the E-interface of $\mathbf{S}$, written $\sigma^E\mathbf{S}$ and referred to as *ideal-world system*. The purpose of σ is to adapt the E-interface of $\mathbf{S}$ such that it resembles the corresponding interface of $\boldsymbol{\pi}\mathbf{R}$.[3] If the two systems $\boldsymbol{\pi}\mathbf{R}$ and $\sigma^E\mathbf{S}$ are indistinguishable, then this roughly means that "whatever an attacker can do in the real world, he can also do in the ideal world".

Apart from the *security* condition described above, we also require an *availability* condition,[4] which excludes trivial protocols: If no attacker is present, the protocol must implement the specified functionality. In the definition, we use the special converter "$\bot$" that, when attached to a certain interface of a system, blocks this interface for the distinguisher.[5]

Definition 2.1 (Construction)**.** *The protocol* π constructs $\mathbf{S}$ from the resource $\mathbf{R}$ within ε and with respect to the class $\mathcal{D}$ of distinguishers *if*

$$\exists\, \sigma \in \Sigma :\ \Delta^{\mathcal{D}}(\boldsymbol{\pi}\mathbf{R}, \sigma^E\mathbf{S}) \leq \varepsilon \qquad \text{and} \qquad \Delta^{\mathcal{D}}(\bot^E\boldsymbol{\pi}\mathbf{R}, \bot^E\mathbf{S}) \leq \varepsilon.$$

An important property of Definition 2.1 is its composability. Intuitively, if a resource $\mathbf{S}$ is used in the construction of a larger system, then the composability implies that $\mathbf{S}$ can be replaced by a construction $\boldsymbol{\pi}\mathbf{R}$ without requiring an explicit security reduction. For completeness, we include the composition theorem (which is adapted from [MT10]) in Section A of the appendix.

(Static) corruption of parties. In our setting with multiple receivers, we are interested in statements about settings in which one or more of the receivers are *corrupted* by the adversary. Here, corruption refers to a setting in which the adversary has full control over the (computer of the) receiver, e.g. by infecting the computer with malware. A resource $\mathbf{R}$ in this setting is (formally) a family of systems $\mathbf{R}_{\mathcal{C}}$ parametrized by a subset $\mathcal{C} \subseteq [n]$, where $i \in \mathcal{C}$ means that B_i is considered corrupted, and the system $\mathbf{R}_{\mathcal{C}}$ will (usually) provide the capabilities corresponding to the interfaces B_i with $i \in \mathcal{C}$ at the E-interface.[6] Constructing a resource $\mathbf{S}$ from a resource $\mathbf{R}$ with respect to static corruption then means that the condition must hold with respect to each set $\mathcal{C} \subseteq [n]$, more formally, for each $\mathcal{C} \subseteq [n]$,

$$\exists\, \sigma \in \Sigma :\ \Delta^{\mathcal{D}}(\boldsymbol{\pi}\mathbf{R}_{\mathcal{C}}, \sigma^E\mathbf{S}_{\mathcal{C}}) \leq \varepsilon$$

[3] Indeed, the adversary can emulate the behavior of any efficient simulator σ; thus, using $\sigma^E\mathbf{S}$ instead of $\mathbf{S}$ can only restrict the adversary's power, so using $\sigma^E\mathbf{S}$ and hence $\boldsymbol{\pi}\mathbf{R}$ instead of $\mathbf{S}$ is safe.

[4] This corresponds to the completeness or correctness properties in some contexts.

[5] The $\bot$-converter also signals to the resource that no attacker is present.

[6] The exact behavior upon corruption depends on the particular resource. In particular, all resources in this work forward all inputs and outputs at interfaces B_i with $i \in \mathcal{C}$ to/from the E-interface, further modifications of their behavior are described explicitly.

must hold in addition to the availability condition in Definition 2.1. The protocol $\boldsymbol{\pi}$ is then said to *construct* $\mathbf{S}$ *from* $\mathbf{R}$ *with respect to static corruptions* if the above conditions are fulfilled for all $\mathcal{C} \subseteq [n]$.

Public-key encryption schemes. A public-key encryption (PKE) scheme with message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, and public-key space $\mathcal{PK}$ is typically described as three algorithms $\mathsf{PKE} = (\mathtt{kgen}, \mathtt{enc}, \mathtt{dec})$. The key-generation algorithm $\mathtt{kgen}$ outputs a key pair (pk, sk), the (probabilistic) encryption algorithm $\mathtt{enc}$ takes a message $m \in \mathcal{M}$ and a public key pk and outputs a ciphertext $c = \mathtt{enc}(pk; m)$, and the decryption algorithm takes a ciphertext $c \in \mathcal{C}$ and a secret key sk and outputs a plaintext $m = \mathtt{dec}(sk; c)$. It is possible that the output of the decryption algorithm is the special symbol $\Diamond$; this indicates that the ciphertext c is invalid.

In constructive cryptography, using PKE in a setting with only one sender and one receiver can be described as deploying converters enc_1 (associated with the sender) and dec_1 (associated with the receiver) as follows. The receiver (within dec_1) initially runs the key-generation algorithm $\mathtt{kgen}$ to obtain a key pair (sk, pk), stores the private key sk locally, and sends the public key pk via an authenticated channel (denoted $\longleftarrow\!\!\bullet$, the first assumed resource). Upon receiving a ciphertext $\tilde{c}$ at the inside interface (via an a priori insecure communication channel $\longrightarrow$, the second assumed resource), dec_1 computes $\tilde{m} = \mathtt{dec}(sk; \tilde{c})$ and outputs $\tilde{m}$. The encryption converter enc_1 initially obtains the public key pk (via $\longleftarrow\!\!\bullet$) and, for each message m obtained at the outside interface, enc_1 computes $c = \mathtt{enc}(pk; m)$ and sends c over the insecure channel $\longrightarrow$. As pointed out already in [MS96], this constructs a confidential channel $\longrightarrow\!\!\!\bullet$.

In this paper, we consider PKE schemes deployed in a setting with one sender A and n receivers $B_1, \ldots, B_n$, corresponding to a tuple $(\mathsf{enc}, \mathsf{dec}, \ldots, \mathsf{dec})$ of $n+1$ converters. Each converter dec is defined similarly to dec_1 above, but if the decryption algorithm $\mathtt{dec}$ outputs an error $\Diamond$, then the converter dec outputs nothing. The encryption converter enc connects at its inside interface to $n+1$ resources. By using the first n resources (which will be instantiated by $\longleftarrow\!\!\bullet^n$, denoting that for each receiver B_i there is one authenticated channel from B_i to A), enc expects to obtain public keys $pk_1, \ldots, pk_n$. Upon receiving $(m, i) \in \mathcal{M} \times [n]$ at the outside interface, enc computes $c = \mathtt{enc}(pk_i; m)$ and sends (c, i) via the $(n+1)$st resource (instantiated by an insecure broadcast network $\lhd$) at the inside interface.

Games and security properties. Game-based definitions specify a property of a cryptographic scheme based on an interaction between two (hypothetical) entities: the game (or challenger) and the adversary. During the interaction, the adversary may issue "oracle queries" to the challenger, the responses of which model what information may be leaked to the adversary. The adversary's goal is specified by the game, and could be, e.g., forging a message or distinguishing encryptions of different messages. If this game cannot be won by any (efficient) adversary, then the scheme is secure against the considered type of attack.

We formalize the adversary and the game as systems that are connected by their interfaces. The game, often denoted as $\mathbf{G}$ with additional super- and subscripts, allows the adversary $\mathbf{A}$ to issue "oracle queries" via that interface. Whether or not the game is won is signaled by a special (monotone) output bit of $\mathbf{G}$ (this can be considered as an additional interface) that is initially 0 but switches to 1 as soon as the winning condition is fulfilled. This bit is denoted Output. For a game $\mathbf{G}$ and an adversary $\mathbf{A}$, we define the *game-winning probability* after q steps (queries) as

$$\Gamma_q^{\mathbf{A}}(\mathbf{G}) \coloneqq \mathsf{P}^{\mathbf{AG}}(\mathsf{Output}_q = 1).$$

For an adversary $\mathbf{A}$ that halts after (at most) q steps, we write $\Gamma^{\mathbf{A}}(\mathbf{G}) \coloneqq \Gamma_q^{\mathbf{A}}(\mathbf{G})$.

Many games considered in the context of encryption schemes, including most games considered here, are *bit-guessing games.* These games can often be described by a pair of systems $\mathbf{G}_0$

and $\mathbf{G}_1$, with the interpretation that in the beginning of the game, a bit $B \in \{0,1\}$ is chosen uniformly at random. The adversary will then be given access to $\mathbf{G}_B$, and the goal is to guess the bit B. The adversary can win such a game with probability $\frac{1}{2}$ trivially by simply guessing the hidden bit. Hence, we measure the adversary's success in terms of his *advantage*, that is, the (absolute) difference between $\mathbf{A}$'s probability of winning $\mathbf{G}$ and the success probability for these "trivial" strategies, formally $\Phi^{\mathbf{A}}(\mathbf{G}) = 2 \cdot \left|\Gamma^{\mathbf{A}}(\mathbf{G}) - \frac{1}{2}\right|$. Note also that $\Phi^{\mathbf{A}}(\mathbf{G}) = \Delta^{\mathbf{A}}(\mathbf{G}_0, \mathbf{G}_1)$.

For a security property that is defined by means of $\mathbf{G}$, we say that the scheme is secure within ε and with respect to a class $\mathcal{A}$ of adversaries if the advantage $\mathbf{A} \in \mathcal{A}$ has in winning $\mathbf{G}$ is bounded by ε.

Asymptotics. To allow for asymptotic security definitions, cryptographic protocols are often equipped with a so-called *security parameter*. We formulate all statements in this paper in a non-asymptotic fashion, but asymptotic statements can be obtained by treating systems $\mathbf{S}$ as asymptotic families $\{\mathbf{S}_k\}_{k\in\mathbb{N}}$ and letting the distinguishing advantage be a real-valued function of k. Then, for a given notion of efficiency, one can consider security with respect to classes of efficient distinguishers and a suitable negligibility notion. All reductions in this work are efficient with respect to the standard polynomial-time notions.

2.2 Games for Confidentiality, Key Privacy, and Robustness

We describe the queries that an adversary can ask in a game formally as *procedures* that he can *call*; the specific game structure is enforced by the order in which they are called. This is not a technically new approach (see for instance [BR06]); however, it integrates smoothly with the security statements we aim for in this work. The most important properties for our work are IND-CCA-security, key privacy, and robustness.

Confidentiality. The most common security property required of PKE schemes is indistinguishability under chosen ciphertext attacks (IND-CCA). Usually, IND-CCA is defined as a left-or-right (LoR) indistinguishability game, where an adversary must guess which of two messages of his choice are encrypted under a known public key. Here, we generally use the real-or-random (RoR) version of IND-CCA; the difference is that the RoR challenger encrypts either an adversarially-chosen message m_0 or a *randomly chosen* m_1 (of the same length as m_0), depending on a hidden bit B. The RoR game formalizes a slightly relaxed condition; there is a (simple) reduction that loses a factor of 2. We describe this game in Figure 1.

Init()	**Decrypt**(c)	**GenChallenge**(m)	**GameOutput**(d)
$(sk, pk) \leftarrow \mathtt{kgen}()$	if $c = \mathsf{Chal}$	if $\mathsf{Chal} \neq \emptyset$	if $\mathsf{Chal} = \emptyset$
$\mathsf{Chal} \leftarrow \emptyset$ //*chal. ctext*	return $\Diamond$	return $\Diamond$	return $\Diamond$
$\mathsf{Output} \leftarrow 0$ //*Output bit*	end if.	$\mathsf{Chal} \leftarrow \mathtt{enc}(pk_B, m)$	else
return pk	$m \leftarrow \mathtt{dec}(sk, c)$	return Chal	$\mathsf{Output} \leftarrow (d = B)$
end.	return m	**end.**	**end.**
	end.		

Figure 1: The IND-CCA game for input B, $\mathbf{G}^{\mathsf{ind\text{-}cca}}$.

Definition 2.2 (IND-CCA security)**.** *A public-key encryption scheme* $\mathsf{PKE} = (\mathtt{kgen}, \mathtt{enc}, \mathtt{dec})$ *is* ε-IND-CCA*-secure with respect to a class* $\mathcal{D}$ *of adversaries if for every* $\mathbf{A} \in \mathcal{D}$ *it holds that*

$$\Phi^{\mathbf{A}}(\mathbf{G}^{\mathsf{ind\text{-}cca}}) = 2 \cdot \left|\Gamma^{\mathbf{A}}(\mathbf{G}^{\mathsf{ind\text{-}cca}}) - \frac{1}{2}\right| \leq \varepsilon.$$

We can also use replayability in IND-CCA security, as described in [CKN03]. In the IND-RCCA game, replays of the challenge ciphertext won't be decrypted (an error message will be output by the decryption algorithm). In particular, c^* is a replay of the challenge ciphertext c if: (1) c^* is input to the decryption oracle after c is generated; and (2) $\texttt{equiv}(c^*, c) = 1$ for an equivalence relation $\texttt{equiv}$. In IND-RCCA security, we set $\texttt{equiv}(c^*, c) = 1$ iff. c^* decrypts to either of the challenge plaintexts m_0, m_1. The game is described in detail in Figure 2, and the security definition is analogous to Definition 2.2.

Init()	**Decrypt**(c)	**GenChallenge**(m)	**GameOutput**(d)
$(sk, pk) \leftarrow \texttt{kgen}()$	if $(\texttt{dec}(sk, c) = \texttt{dec}(sk, \mathsf{Chal}))$	if $\mathsf{Chal} \neq \emptyset \vee\ (c = \mathsf{Chal})$	if $\mathsf{Chal} = \emptyset$
$\mathsf{Chal} \leftarrow \emptyset$	return $\Diamond$	return $\Diamond$	return $\Diamond$
$\mathsf{Output} \leftarrow 0$	end if.	else	else
return pk	$m \leftarrow \texttt{dec}(sk, c)$	$c_B \leftarrow \texttt{enc}(pk_B, m)$	$\mathsf{Output} \leftarrow (d = B)$
end.	return m	$\mathsf{Chal} \leftarrow c_B$	**end.**
	end.	return c_B	
		end.	

Figure 2: The IND-RCCA game for input B, $\mathbf{G}^{\mathsf{ind\text{-}rcca}}$.

Key privacy. In a key-private PKE scheme, the adversary, given two public keys pk_0 and pk_1, must be unable to tell which key was used to generate a given ciphertext [BBDP01]. This definition is similar in spirit to the standard "left-or-right" IND-CCA definition, where the adversary is given the public key, but does not know which of two messages is encrypted under it. In the key-privacy game the message is known, but not the public key. The notion can be formalized as a bit-guessing game $\mathbf{G}_B$ (for a hidden bit B), which we show in Figure 3.

In particular, we denote by Chal the challenge ciphertext and by Output, the output bit. The decryption procedure $\mathbf{Decrypt}(\cdot, \cdot)$ takes as input a ciphertext c and a bit bit, the latter indicating under which secret key $\mathbf{A}$ wishes to decrypt c. If the adversary asks to decrypt the challenge ciphertext *after* the challenge has been generated, the algorithm returns $\Diamond$; else, it returns the decryption of the ciphertext c under sk_{bit}. The challenge generation algorithm **GenChallenge** can be run only exactly once during the game; if the adversary runs this procedure for a second time, the algorithm outputs $\Diamond$. The challenge ciphertext is then output to the adversary.

Init()	**Decrypt**(c, bit)	**GenChallenge**(m)	**GameOutput**(d)
$(sk_0, pk_0) \leftarrow \texttt{kgen}()$	if $c = \mathsf{Chal}$	if $\mathsf{Chal} \neq \emptyset$	if $\mathsf{Chal} = \emptyset$
$(sk_1, pk_1) \leftarrow \texttt{kgen}()$	return $\Diamond$	return $\Diamond$	return $\Diamond$
$\mathsf{Chal} \leftarrow \emptyset$	end if.	else	else
$\mathsf{Output} \leftarrow 0$	$m \leftarrow \texttt{dec}(sk_{\mathsf{bit}}, c)$	$c_B \leftarrow \texttt{enc}(pk_B, m)$	$\mathsf{Output} \leftarrow (d = B)$
return pk_0, pk_1	return m	$\mathsf{Chal} \leftarrow c_B$	**end.**
end.	**end.**	return c_B	
		end.	

Figure 3: The IK-CCA game for input B, $\mathbf{G}_B^{\mathsf{ik\text{-}cca}}$.

Definition 2.3 (IK-CCA security)**.** *A public-key encryption scheme* $\mathsf{PKE} = (\texttt{kgen}, \texttt{enc}, \texttt{dec})$ *is* ε-IK-CCA*-secure with respect to a class* $\mathcal{D}$ *of adversaries if for every* $\mathbf{A} \in \mathcal{D}$ *it holds that*

$$\Phi^{\mathbf{A}}(\mathbf{G}^{\mathsf{ik\text{-}cca}}) = 2 \cdot \left| \Gamma^{\mathbf{A}}(\mathbf{G}^{\mathsf{ik\text{-}cca}}) - \frac{1}{2} \right| \leq \varepsilon.$$

We use two further variants of key privacy. The first is replayable IK-CCA (or IK-RCCA) security, which is a weakened version of IK-CCA security in the same spirit in which IND-RCCA

is a weakened version of IND-CCA. The complete description of the IK-RCCA game is depicted in Figure 4.

Init()	**Decrypt**(c, bit)	**GenChallenge**(m)	**GameOutput**(d)
$(sk_0, pk_0) \leftarrow \mathtt{kgen}()$ $(sk_1, pk_1) \leftarrow \mathtt{kgen}()$ $\mathsf{Chal} \leftarrow \emptyset$ *//chal. ctext* $\mathsf{Output} \leftarrow 0$ *//Output bit* return pk_0, pk_1 **end.**	if $(c = \mathsf{Chal}) \vee (\mathtt{dec}(sk_B, \mathsf{Chal})$ $\in \{\mathtt{dec}(sk_0, c), \mathtt{dec}(sk_1, c)\})$ return $\Diamond$ end if. $m \leftarrow \mathtt{dec}(sk_{\mathsf{bit}}, c)$ return m **end.**	if $\mathsf{Chal} \neq \emptyset$ return $\Diamond$ else $c_B \leftarrow \mathtt{enc}(pk_B, m)$ $\mathsf{Chal} \leftarrow c_B$ return c_B **end.**	if $\mathsf{Chal} = \emptyset$ return $\Diamond$ else $\mathsf{Output} \leftarrow (d = B)$ **end.**

Figure 4: The IK-RCCA game for input B, $\mathbf{G}_B^{\mathsf{ik\text{-}rcca}}$.

The second notion we consider is 1-sided-replayable-CCA security, where the decryption oracle is modified so that it only decrypts under the first of the generated secret keys. The game is depicted in Figure 5. The security definitions in both cases is analogous to Definition 2.3, but with respect to the modified games. We relate the notions of 1-sided and standard IK-CCA

Init()	**Decrypt**(c)	**GenChallenge**(m)	**GameOutput**(d)
$(sk_0, pk_0) \leftarrow \mathtt{kgen}()$ $(sk_1, pk_1) \leftarrow \mathtt{kgen}()$ $\mathsf{Chal} \leftarrow \emptyset$ $\mathsf{Output} \leftarrow 0$ return pk_0, pk_1 **end.**	if $c = \mathsf{Chal}$ return $\Diamond$ end if. $m \leftarrow \mathtt{dec}(sk_0, c)$ return m **end.**	if $\mathsf{Chal} \neq \emptyset$ return $\Diamond$ else $c_B \leftarrow \mathtt{enc}(pk_B, m)$ $\mathsf{Chal} \leftarrow c_B$ return c_B **end.**	if $\mathsf{Chal} = \emptyset$ return $\perp$ else $\mathsf{Output} \leftarrow (d = B)$ **end.**

Figure 5: The IK-1-sided-CCA game for input B, $\mathbf{G}_B^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}$.

security in the following Lemma.

Lemma 2.4. *Let* $\mathsf{PKE} = (\mathtt{kgen}, \mathtt{enc}, \mathtt{dec})$ *be a public-key encryption scheme, and assume that the two invocations of* $\mathtt{kgen}$ *in* $\mathbf{G}^{\mathsf{ik\text{-}cca}}$ *use independent randomness. The following two statements hold:*

1. *There is a reduction* $\mathbf{R}(\cdot)$ *such that for every 1-sided-IK-CCA adversary* $\mathbf{A}$*,*

$$\Phi^{\mathbf{A}}(\mathbf{G}^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}) \leq \Phi^{\mathbf{R}(\mathbf{A})}(\mathbf{G}^{\mathsf{ik\text{-}cca}}).$$

2. *There is a reduction* $\mathbf{R}'(\cdot)$ *such that for every IK-CCA adversary* $\mathbf{A}$*,*

$$\Phi^{\mathbf{A}}(\mathbf{G}^{\mathsf{ik\text{-}cca}}) \leq 2 \cdot \Phi^{\mathbf{R}'(\mathbf{A})}(\mathbf{G}^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}).$$

Proof sketch. The first statement follows immediately, as the reduction can emulate $\mathbf{G}^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}$ perfectly. For the second statement, the reduction must simulate $\mathbf{A}$'s queries to the decryption oracle. There are two crucial points here. First, the reduction will hand $\mathbf{A}$ the public key pk_0 received from $\mathbf{G}^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}$ (denoted pk), and one generated by $\mathbf{R}'(\mathbf{A})$ itself, denoted pk^*, for which it knows the private key. However, note that the *order* in which the adversary receives the keys is important, as $\mathbf{G}^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}$ always decrypts ciphertexts using the "first" private key: thus, the reduction first flips an independently chosen bit b; the order in which it hands the public keys to $\mathbf{A}$ depends on this bit. The reduction answers decryption queries either by using sk^* or by using $\mathbf{G}^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}$ (depending on b and the bit in the decryption query). Finally, when $\mathbf{A}$ outputs some guess d, $\mathbf{R}'(\cdot)$ guesses $d \oplus b$. For the analysis, note that if $\mathbf{R}'(\mathbf{A})$

interacts with $\mathbf{G}_0^{\text{1-sided-ik-cca}}$, then the simulation of $\mathbf{G}^{\text{ik-cca}}$ is perfect. Otherwise, the adversary's guess is statistically independent of (and hence completely randomized by) b. The total success probability is $\mathsf{P}^{\mathbf{R}'(\mathbf{A})\mathbf{G}^{\text{1-sided-ik-cca}}}(\mathsf{Output}=1) = \frac{1}{2}p_{\mathbf{A}} + \frac{1}{4}$. The overall advantage of the reduction is $2\mathsf{P}^{\mathbf{R}'(\mathbf{A})\mathbf{G}^{\text{1-sided-ik-cca}}}(\mathsf{Output}=1) - 1 = \mathsf{P}^{\mathbf{A}\mathbf{G}^{\text{ik-cca}}}(\mathsf{Output}=1) - \frac{1}{2} = \frac{\Phi^{\mathbf{A}}(\mathbf{G}^{\text{ik-cca}})}{2}$. □

Robustness. The notion of *robustness* in encryption was formalized by Abdalla et al. [ABN10] in two flavors: *weak* and *strong* robustness. They consider both versions under both chosen plaintext and chosen ciphertext attacks. We focus here on weak, resp. strong robustness under chosen ciphertext attacks (WROB-CCA, resp. SROB-CCA), associated with the experiments in Figures 6, resp. 7, where the adversary may call the following oracles.

- On input an identifier ID, the oracle $\mathbf{GenUser}(\cdot)$ generates a public and a private key for the user ID and returns the public key. A set U keeps track of the users generated by the $\mathbf{GenUser}(\cdot)$ oracle, i.e. the honestly generated key pairs.
- On input a valid identifier $\mathsf{ID} \in U$, the oracle $\mathbf{Corrupt}(\cdot)$ returns the private key corresponding to user ID and adds the identifier to a set V.
- On input a valid identifier $\mathsf{ID} \in U$ and a ciphertext c, the decryption oracle $\mathbf{Decrypt}(\cdot,\cdot)$ outputs the corresponding plaintext m.

Init()	**GenUser**(ID)	**Corrupt**(ID)	**Decrypt**(ID, c)	**GameOutput**(m, ID_0, ID_1)
$U \leftarrow \emptyset$	$(sk_{\mathsf{ID}}, pk_{\mathsf{ID}}) \leftarrow \mathtt{kgen}()$	if $(\mathsf{ID};\cdot;\cdot) \notin U$	if $(\mathsf{ID};\cdot;\cdot) \notin U$	if $(\mathsf{ID}_0 = \mathsf{ID}_1) \vee \{\mathsf{ID}_0, \mathsf{ID}_1\} \cap V \neq \emptyset$
$V \leftarrow \emptyset$	$U \leftarrow U \cup \{(\mathsf{ID}; sk_{\mathsf{ID}}; pk_{\mathsf{ID}})\}$	return $\Diamond$	return $\Diamond$	return $\Diamond$
$\mathsf{Output} \leftarrow 0$	return pk_{ID}	end if.	end if.	end if. $c \leftarrow \mathtt{enc}(pk_{\mathsf{ID}_0}, m)$
end.	**end.**	$V \leftarrow V \cup \{\mathsf{ID}\}$	return $\mathtt{dec}(sk_{\mathsf{ID}}, c)$	$m_1 \leftarrow \mathtt{dec}(sk_{\mathsf{ID}_1}, c)$
		return sk_{ID} from U	**end.**	$\mathsf{Output} \leftarrow (m \neq \Diamond) \wedge (m_1 \neq \Diamond)$
		end.		**end.**

Figure 6: The weak robustness game, $\mathbf{G}^{\text{w-rob}}$.

In the WROB-CCA game, the adversary chooses a plaintext and two identities. The plaintext is encrypted by the challenger (without tampering) for the first identity. The adversary wins if this ciphertext decrypts to a valid plaintext for the second identity. By contrast, for strong robustness (SROB-CCA), the adversary can manipulate ciphertexts and wins if a chosen ciphertext decrypts to valid plaintexts for two different public keys.

Init()	**GenUser**(ID)	**Corrupt**(ID)	**Decrypt**(ID, c)	**GameOutput**(c, ID_0, ID_1)
$U \leftarrow \emptyset$	$(sk_{\mathsf{ID}}, pk_{\mathsf{ID}}) \leftarrow \mathtt{kgen}()$	if $(\mathsf{ID};\cdot;\cdot) \notin U$	if $(\mathsf{ID};\cdot;\cdot) \notin U$	if $(\mathsf{ID}_0 = \mathsf{ID}_1) \vee \{\mathsf{ID}_0, \mathsf{ID}_1\} \cap V \neq \emptyset$
$V \leftarrow \emptyset$	$U \leftarrow U \cup \{(\mathsf{ID}; sk_{\mathsf{ID}}; pk_{\mathsf{ID}})\}$	return $\Diamond$	return $\Diamond$	return $\Diamond$
$\mathsf{Output} \leftarrow 0$	return pk_{ID}	end if.	end if.	end if.
end.	**end.**	$V \leftarrow V \cup \{\mathsf{ID}\}$	return $\mathtt{dec}(sk_{\mathsf{ID}}, c)$	$\mathsf{Output} \leftarrow (\mathtt{dec}(sk_{\mathsf{ID}_0}, c) \neq \Diamond) \wedge$
		return sk_{ID} from U	**end.**	$(\mathtt{dec}(sk_{\mathsf{ID}_1}, c) \neq \Diamond)$
		end.		**end.**

Figure 7: The strong robustness game, $\mathbf{G}^{\text{s-rob}}$.

3 Receiver-Anonymous Communication

The main goal of this work is to model and achieve confidential and receiver-anonymous communication. We first formalize a useful anonymity guarantee by describing in Section 3.1 the resource  , which *can* actually be constructed from a "broadcast" channel and several authenticated channels (to transmit the public keys). We then discuss in Section 3.2 in which

(inefficient) way this construction can be achieved by vanilla public-key encryption, and, in Section 3.3, we argue that "much more" anonymity is impossible to achieve. Finally, in Section 3.4 we show how to achieve this construction more efficiently, by means of a public-key encryption scheme with the properties IND-CCA, IK-CCA [BBDP01], and WROB-CCA [ABN10].

3.1 Resources for Receiver-Anonymous Communication

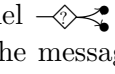
An n-receiver channel is a resource with an interface labeled A for the sender, interfaces labeled $B_1, \ldots, B_n$ for the receivers, and a third type of interface labeled E that captures potential adversarial access. The security properties of different n-receiver channels are described in the following; the symbolic notation for the channels extends that from [MS96].

The security statements in this work are parametrized by the number of messages that are transmitted over the channels. More precisely, for each of the following channels and each $q \in \mathbb{N}$, we define the *q-bounded channel* as the one that processes (only) the first q queries at the A-interface and the first q queries at the E-interface as described, and ignores all further queries at these interfaces. We then require from a protocol that it constructs, for all $q \in \mathbb{N}$, the q-bounded "ideal" channel from the q-bounded assumed channel.[7] Wherever the number q is significant, such as in the theorem statements, we denote the q-bounded versions of channels by writing the q on top of the channel symbol (e.g., $\overset{q}{\preceq}$); we omit it in places that are of less formal nature.

Insecure broadcast communication. We base our constructions on a resource $\preceq$, which allows the sender to broadcast a given message to all receivers $B_1, \ldots, B_n$. Such a channel can be implemented, for example, by multi-sending the same message individually to each receiver over an insecure network; the channel models also what is achieved by wireless broadcast. The resource $\preceq$ leaks the complete message at the E-interface, and allows to delete, change, or inject messages destined for particular receivers via the E-interface. In more detail:

- If at the E-interface the $\perp$-converter is connected,[8] then on input the k-th message m_k at the A-interface, output m_k at B_j for all $j \in [n]$.
- Otherwise, on input the k-th message m_k at the A-interface, output m_k at the E-interface. Upon the query $(\texttt{inject}, j, \tilde{m})$ at the E-interface for $j \in [n]$ and $\tilde{m} \in \mathcal{M}$, deliver $\tilde{m}$ at interface B_j.

The behavior of $\preceq$ is depicted in Figure 8. On an input a message m at the A-interface, this message is leaked at the E-interface. In contrast, the E-interface allows a potential attacker to target messages to the individual recipients, such as message m to receiver i or message $\tilde{m}$ to receiver i' in the figure. In case no adversary is present (this is not depicted), messages input at the A-interface are immediately delivered at all receivers' interfaces.

Confidential receiver-anonymous communication. The confidential receiver-anonymous channel  leaks neither the message contents nor the intended recipient to the adversary, just the message length. It allows, however, to "conditionally" deliver a message to a chosen user if and only if this chosen user was the originally intended recipient.

- If at the E-interface the $\perp$-converter is connected, then on the k-th input (m_k, i_k) at the A-interface, output m_k at B_{i_k}.

[7] This condition is equivalent to considering an "unbounded" channel; the important feature is that *the protocol* is independent of the number q of messages.

[8] Formally, there is a special input that provokes this behavior, and the converter $\perp$ provides this input.

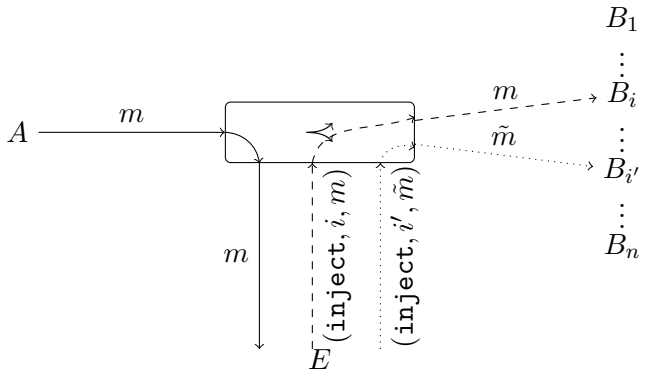


Figure 8: The insecure broadcast channel.

- Otherwise, on the k-th input (m_k, i_k) at the A-interface, output the message length $|m_k|$ at the E-interface. (If $i_k \in \mathcal{C}$, then output (m_k, i_k) at the E-interface.) Furthermore, the E-interface allows the following queries:
  - $(\texttt{inject}, j, \tilde{m})$ for $j \in [n]$ and $\tilde{m} \in \mathcal{M}$: delivers $\tilde{m}$ at interface B_j;
  - $(\texttt{deliver}, j, \bar{k})$ for $j \in [n]$, $\bar{k} \in \mathbb{N}$: If at least $\bar{k}$ messages have been sent via A and $i_{\bar{k}} = j$, then it delivers the message $m_{\bar{k}}$ at B_j.

This is also depicted in Figure 9. In the application of a public-key cryptosystem to a broadcast network such as —◇—, the capabilities at the E-interface correspond to trial deliveries of intercepted messages and to adversarial encryptions.

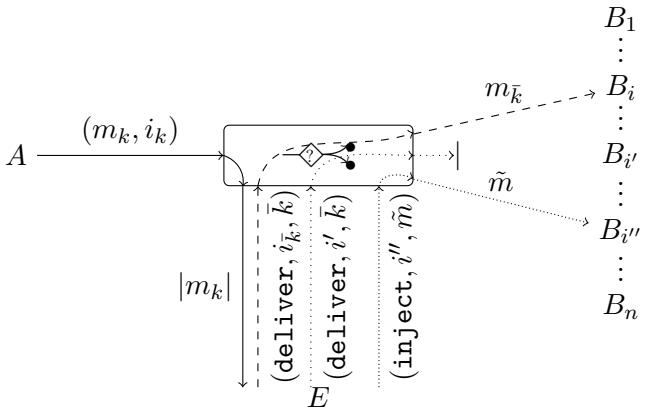


Figure 9: The confidential receiver-anonymous channel.

Authenticated channel. Each receiver uses one authenticated channel $\longleftarrow\!\bullet$ to send its public key to the sender; we use n parallel authenticated channels, denoted $\longleftarrow\!\bullet^n$ (one for each receiver), as assumed resources in our constructions. Formally, a single authenticated channel $\longleftarrow\!\bullet$ with message space $\mathcal{M}$ is a three-party resource with interfaces A, B_i (for some i), and E. On input a message $m \in \mathcal{M}$ at interface B_i, the channel outputs m at the E-interface. The channel outputs m at the A-interface only upon receiving an acknowledgement from the E-interface (the adversary controls message delivery). If the sender B_i is corrupted, the message can be injected via the E-interface.

3.2 Generic Construction using Public-Key Encryption

The channel $\multimap\!\!\diamond\!\!\prec$ can be constructed from $\prec$ and $\longleftarrow\!\bullet^n$ using any secure public-key scheme: Each receiver generates a key pair and sends the public key through its authenticated channel $\longleftarrow\!\bullet$ to the sender; the sender transmits a message to a specific receiver by concatenating (in a fixed predetermined order): an encryption of this message under the intended receiver's public key and a "garbage" message encrypted with the appropriate key for each additional potential receiver; this composite message is then sent via the broadcast channel. Each receiver decrypts only "its" part of the composite ciphertext and checks whether or not the message was "garbage." (A simple way to implement the above "garbage" message is to set it to a public constant message $\bar{m} \in \mathcal{M}$ which is used only for that purpose.) If the broadcast channel is achieved by multi-sending the same message to each receiver, then one can also send only the corresponding part to each receiver.

Yet, this approach has two main disadvantages. First, the computation and communication complexity is linear in the (potentially large) number of possible receivers. Second, the sender must *know* the public keys of all potential receivers, not just of the one intended receiver.

3.3 "Upper Bounds" on Anonymity

Informally speaking, anonymity beyond the guarantees formalized by $\multimap\!\!\diamond\!\!\prec$ is unlikely to be achieved from the resources $\prec$ and $\longleftarrow\!\bullet^n$ which we assumed. Indeed, we show that a (minor and natural) extension of $\multimap\!\!\diamond\!\!\prec$ cannot be achieved from our assumed resources. The extension, denoted by ANON, removes the "conditional delivery" capability provided at the E-interface in resource $\multimap\!\!\diamond\!\!\prec$, and enables deliveries of the type $(\texttt{deliver}, \bar{k})$ for $\bar{k} \in \mathbb{N}$, where, if at least $\bar{k}$ messages have been sent via A, then the message $m_{\bar{k}}$ is delivered to $B_{i_{\bar{k}}}$. In particular, the distinguisher can use the E-interface of system $\prec$ to deliver the messages to, e.g., only one chosen receiver, which will output the message if and only if it is the intended recipient. We call this process a "trial delivery" and show that it allows the distinguisher to tell the real-world system apart from the ideal-world system with ANON, where trial deliveries are impossible by definition.

This result is formalized in Theorem 3.1, and we give a proof sketch below. Note that the channel ANON is just one type of ideal resource providing stronger anonymity guarantees than $\multimap\!\!\diamond\!\!\prec$; however, our impossibility result extends easily to any resource without conditional deliveries.

In the proof, we construct a distinguisher $\mathbf{D}_2$ that exploits this difference in that inputs at the A-interface a message m for some receiver B_j with $j \in [n]$ uniformly at random. Once the ciphertext sent via $\prec$ is received at the E-interface, $\mathbf{D}_2$ forwards this message (each with probability $\frac{1}{2}$) *either* to the intended receiver B_j, *or* to some other receiver $B_{j'}$, with $j' \neq j$. While in the protocol, m is delivered iff the message was delivered to B_j (known to $\mathbf{D}_2$), the simulator is oblivious of j and will choose the wrong action—whether or not to use `deliver`—with probability at least $\frac{1}{2}$.

Theorem 3.1. *Let $(\pi_A, \pi_B, \ldots, \pi_B)$ be any protocol supposed to construct ANON from $\prec$ and $\longleftarrow\!\bullet^n$. There are distinguishers $\mathbf{D}_1$ and $\mathbf{D}_2$ such that, for any simulator σ,*

$$\varepsilon_{\mathsf{av}} \coloneqq \Delta^{\mathbf{D}_1}(\pi_A^A \pi_B^{B_1} \ldots \pi_B^{B_n} \bot^E[\longleftarrow\!\bullet^n, \prec], \bot^E(\mathsf{ANON}))$$

and

$$\varepsilon_{\mathsf{sec}} \coloneqq \Delta^{\mathbf{D}_2}(\pi_A^A \pi_B^{B_1} \ldots \pi_B^{B_n}[\longleftarrow\!\bullet^n, \prec], \sigma^E(\mathsf{ANON}))$$

such that $\varepsilon_{\mathsf{av}} + \varepsilon_{\mathsf{sec}} \geq \frac{1}{2}$.

Proof sketch. Assume that there exist a protocol $(\pi_A, \pi_B, \ldots, \pi_B)$ and a simulator σ such that the transformation holds. The distinguishers $\mathbf{D}_1$ and $\mathbf{D}_2$ described in the following start by taking some fixed message $m \in \mathcal{M}$ and sending it to the receiver B_j with $j \in [n]$ chosen uniformly at random.

If the output at the B_j-interfaces is as expected—i.e., B_j outputs m, all B_i with $i \neq j$ remain silent—the distinguisher $\mathbf{D}_1$ outputs 0, otherwise it outputs 1. The distinguisher $\mathbf{D}_2$ begins similarly to $\mathbf{D}_1$ but obtains the message c that is broadcast by the sender at the adversarial interface of $\prec\!\!\!\triangleleft$. Then, $\mathbf{D}_2$ chooses some $j' \neq j$ and uses the E-interface of $\prec\!\!\!\triangleleft$ to forward the transmitted message *either* only to B_j *or* only to $B_{j'}$, each with probability $\frac{1}{2}$. Then,

- $\mathbf{D}_2$ delivers to B_j: if $\mathbf{D}_2$ obtains the output m at B_j (and no output elsewhere), output 0, otherwise output 1;
- $\mathbf{D}_2$ delivers to $B_{j'}$: if $\mathbf{D}_2$ obtains any output at any receiver's interface, output 1, otherwise output 0.

By definition of ANON and $\mathbf{D}_1$, $\mathsf{P}\left(\mathbf{D}_1 \bot^E(\mathsf{ANON}) = 0\right) = 1$. Note that, when connected to the real-world system, the distinguisher $\mathbf{D}_2$ outputs 0 with at least the same probability as $\mathbf{D}_1$: If $\mathbf{D}_2$ outputs 1, then either $\mathbf{D}_2$ delivers to B_j and B_j does not output m although it received the message, so $\mathbf{D}_1$ would also output 1; or $\mathbf{D}_2$ delivers to $B_{j'}$ and $B_{j'}$ provides output, in which case $\mathbf{D}_1$ also outputs 1. Formally,

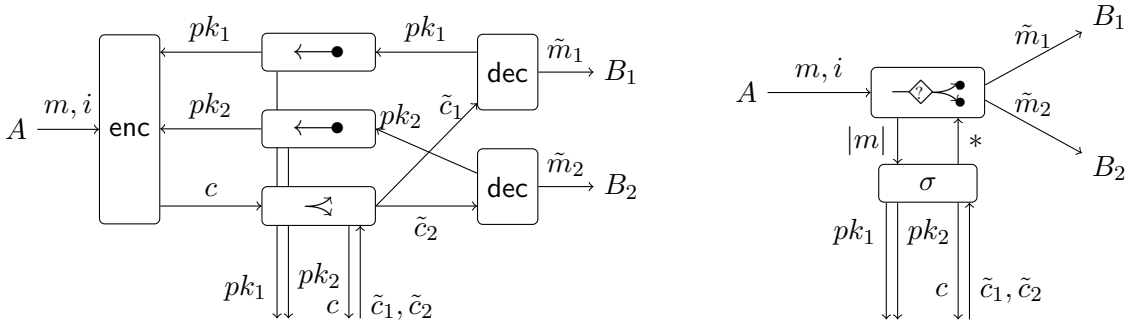
$$\mathsf{P}\left(\mathbf{D}_2 \pi_A^A \pi_B^{B_1} \ldots \pi_B^{B_n}[\prec\!\!\!\triangleleft, \longleftarrow\!\!\bullet^n] = 0\right) \geq \mathsf{P}\left(\mathbf{D}_1 \pi_A^A \pi_B^{B_1} \ldots \pi_B^{B_n} \bot^E[\prec\!\!\!\triangleleft, \longleftarrow\!\!\bullet^n] = 0\right).$$

Furthermore, $\mathsf{P}\left(\mathbf{D}_2 \sigma^E(\mathsf{ANON}) = 0\right) \leq \frac{1}{2}$. To see this, note that, when the distinguisher forwards the message, the simulator has a choice of whether to deliver the message (by using the `deliver`-command) or not. This choice is independent of whether or not $\mathbf{D}_2$ forwards the message to either B_j or to $B_{j'}$ (the simulator is oblivious to the distinguisher's choice of j), so the decision of the simulator is correct with probability at most $\frac{1}{2}$. This concludes the proof. □

3.4 Achieving Confidential Receiver-Anonymous Communication

A public-key encryption scheme constructs the resource $\multimap\!\!\diamond\!\!\prec\!\!\bullet$ from a broadcast channel if it has the properties IND-CCA, IK-CCA, and WROB-CCA. The property WROB-CCA (weak robustness) captures the guarantee that ciphertexts honestly generated for one user will not be successfully decrypted by another user. We show that weak robustness is sufficient for our construction; in view of the discussion in [ABN10], this may be surprising, since the adversary can inject arbitrary ciphertexts into the channel $\prec\!\!\!\triangleleft$. The intuitive reason why WROB-CCA is sufficient is two-fold: First, preventing the adversary from generating a single "fresh" ciphertext that is accepted by two receivers is only helpful if, for some reason, injecting two different ciphertexts is impossible, or harder for the adversary than injecting a single one (cf. Section 4.3). Second, the non-malleability guarantees of IND-CCA exclude that the adversary can "maul" honestly generated ciphertexts such that unintended receivers decrypt "related" plaintexts (this is used in the reduction to IND-CCA in the proof of Theorem 3.2).

The security statement we prove below is depicted in Figure 10, where we show how the scheme is used together with the assumed resources: Each sender transmits its public key authentically to the sender, who then uses the broadcast channel to transmit the ciphertext to both receivers. Figure 10b shows the idealized setting, where the message is transmitted via the resource $\multimap\!\!\diamond\!\!\prec$ (which guarantees confidentiality). The value "$*$" is determined by the simulator and depends on the values $\tilde{c}_1$ and $\tilde{c}_2$ given by the adversary; the symbol may stand for a query to deliver the message m or to inject unrelated messages.



(a) Using public-key encryption over an (insecure) broadcast.

(b) The idealized setting, the value "$*$" depends on $\tilde{c}_1$ and $\tilde{c}_2$.

Figure 10: The security statement in a setting with two receivers.

Theorem 3.2 shows that if the public-key encryption scheme has the three assumed properties, then the two settings in Figure 10 are indistinguishable. The intuitive interpretation of this statement is that whenever such a scheme is used to protect messages transmitted via a broadcast channel such as $\prec$, one obtains the guarantees explicitly described by the "idealized" network resource $-\diamond\!\!\prec\bullet$. The proof of the theorem shows that every distinguisher for the two settings can be transformed into an adversary against (at least) one of the three properties IND-CCA, IK-CCA, and WROB-CCA with loss qn for q messages and n receivers.

Theorem 3.2. *Let* $(\mathtt{kgen}, \mathtt{enc}, \mathtt{dec})$ *be a public-key encryption scheme that has the three properties* IND-CCA, IK-CCA, *and* WROB-CCA. *Then, the protocol* $\mathsf{pke} = (\mathsf{enc}, \mathsf{dec}, \ldots, \mathsf{dec})$ *obtained from* $(\mathtt{kgen}, \mathtt{enc}, \mathtt{dec})$ *as in Section 2.1 constructs* $-\diamond\!\!\prec\bullet$ *from* $\prec$ *and* $(\longleftarrow\!\bullet)^n$ *with respect to static corruptions. More formally, for each* $\mathcal{C} \subseteq [n]$*, there are a simulator* σ *and for each* $q \in \mathbb{N}$ *four reductions* $\mathbf{A}_q(\cdot)$, $\mathbf{A}'_q(\cdot)$, $\mathbf{A}''_q(\cdot), \mathbf{A}'''_q(\cdot)$ *such that*

$$\Delta^{\mathbf{D}}\left(\mathsf{pke}\perp^E\left[\overset{q}{\prec}, \longleftarrow\!\bullet^n\right], \perp^E -\overset{q}{\diamond}\!\!\prec\bullet\right) \leq qn \cdot \Gamma^{\mathbf{A}_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{w\text{-}rob}}\right), \tag{1}$$

and

$$\begin{aligned}\Delta^{\mathbf{D}}&\left(\mathsf{pke}\left[\overset{q}{\prec}_{\mathcal{C}}, \longleftarrow\!\bullet^n_{\mathcal{C}}\right], \sigma^E -\overset{q}{\diamond}\!\!\prec\bullet_{\mathcal{C}}\right)\\ &\leq qn \cdot \Phi^{\mathbf{A}'_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{ind\text{-}cca}}\right) + qn \cdot \Phi^{\mathbf{A}''_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{ik\text{-}cca}}\right) + qn \cdot \Gamma^{\mathbf{A}'''_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{w\text{-}rob}}\right).\end{aligned} \tag{2}$$

Proof sketch. We sketch the proofs for conditions (1) and (2) independently.

Availability. We describe a reduction $\mathbf{A}_q(\cdot)$ that turns a distinguisher $\mathbf{D}$ between the real-world system $\mathsf{pke}\perp^E[\prec, \longleftarrow\!\bullet^n]$ (which we denote $\mathbf{R}_\perp$) and the ideal-world system $\perp^E(-\diamond\!\!\prec\bullet)$ (denoted $\mathbf{S}_\perp$) into an adversary for the the WROB-CCA game. The idea of the proof is to construct a monotone event sequence (MES, see [Mau02]), which becomes true once the distinguisher inputs a pair (m, i) at the A-interface such that a receiver B_j for some index $\mathcal{C} \not\ni j \neq i$ outputs some plaintext $m_j \neq \Diamond$. If the encryption scheme has perfect correctness, the systems $\mathbf{R}_\perp$ and $\mathbf{S}_\perp$ are equivalent, conditioned on the MES remaining false (if the scheme is *not* perfectly correct, we alter the MES to take this into account). Yet, note that even isolating a query (m, i) that invokes the MES does not immediately imply that a new encryption of the same m and pk_i will yield another ciphertext (in the query of the WROB-CCA game) that decrypts to $m'_j \neq \Diamond$ by sk_j for the index $j \neq i$. Instead, for the reduction to be successful, the reduction

$\mathbf{A}_q$ guesses the query and the receiver where this erroneous decryption will occur. Thus, the reduction loses a factor qn, as claimed. (If $\mathcal{C} \neq \emptyset$, the loss is smaller.)

Security. We first describe the simulator σ attached to the E-interface of the ideal resource. The role of σ is to simulate the interaction at the E-interface to a distinguisher. We then prove that σ is indeed a good simulator: in other words, we provide reductions that transform a given successful distinguisher into a successful adversary against one of the following games: IND-CCA, IK-1-sided-CCA, or WROB-CCA. The simulator σ runs as follows:

- Generate $n-|\mathcal{C}|$ private-/public-key pairs (pk_i, sk_i) with $i \in [n]\backslash\mathcal{C}$ to simulate each pk_i that it is transmitted via the corresponding channel $\longleftarrow\!\bullet$. Furthermore, generate one auxiliary key pair $(\tilde{pk}, \tilde{sk})$. For all $j \in \mathcal{C}$, obtain a public key $\bar{pk}_i$ at the simulated E-interface of $\longleftarrow\!\bullet$.
- Upon the k-th message length ℓ_k from $-\!\diamond\!\!\prec\!\bullet$, generate a new ciphertext $c_k = \mathtt{enc}(\tilde{pk}; 0^{\ell_k})$ and simulate c_k as a message on $\prec$. If the channel leaked a pair (m_k, i_k) (which means $i_k \in \mathcal{C}$), encrypt m_i using $\bar{pk}_{i_k}$ and simulate that ciphertext instead.
- When $\mathbf{D}$ delivers a message $\tilde{c}$ to some user $j \in [n] \setminus \mathcal{C}$:
  - In case $\tilde{c} = c_{\bar{k}}$ for some $\bar{k} \in \mathbb{N}$, issue $(\mathtt{deliver}, j, \bar{k})$ to $-\!\diamond\!\!\prec\!\bullet$.
  - In case $\tilde{c}$ is "fresh," compute $\tilde{m}_j \coloneqq \mathtt{dec}(sk_j; \tilde{c})$, and, if $\tilde{m}_j \neq \Diamond$, issue $(\mathtt{inject}, j, \tilde{m}_j)$ to $-\!\diamond\!\!\prec\!\bullet$.

Assume that there exists a distinguisher $\mathbf{D}$ that successfully distinguishes the real-world system $\mathsf{pke}\,[\prec_{\mathcal{C}}, \longleftarrow\!\bullet^n_{\mathcal{C}}]$ from the ideal-world system $\sigma^E - \!\diamond\!\!\prec\!\bullet_{\mathcal{C}}$. Before we sketch the security reductions to the underlying games, we remark that the simulation for corrupted receivers is always perfect (the ciphertext is computed exactly as in the honest encryption, and the injected ciphertexts are simply forwarded) and can hence be ignored in the following arguments.

WROB-CCA**.** As a first intermediate step, we introduce a hybrid resource $\mathbf{H}_1$. This resource behaves like $-\!\diamond\!\!\prec\!\bullet$, except that it allows for the delivery of an arbitrary message to a party other than the intended recipient: namely, instead of the query $(\mathtt{deliver}, j, \bar{k})$, we allow to deliver a message $\tilde{m}$ to a user B_j for $j \neq i_{\bar{k}}$ (still $m_{\bar{k}}$ for $j = i_{\bar{k}}$) by means of $(\mathtt{deliver}, j, \bar{k}, \tilde{m})$. We use a modified simulator σ_1 that sends the decryption of the ciphertext simulated for message $\bar{k}$ under the key of user j. The systems $\sigma_1^E\mathbf{H}_1$ and $\sigma^E - \!\diamond\!\!\prec\!\bullet$ are equivalent unless, for some query, there is a user B_j, not the intended recipient of some ciphertext, that outputs a message upon receiving the ciphertext. A distinguisher that provokes this situation (i.e., it causes some unintended recipient to output a message from a ciphertext) can be used to win the WROB-CCA game. The reduction $\mathbf{A}'''_q(\cdot)$ obtains n generated keys from the WROB-CCA game, which correspond to the users, and an additional key, used to simulate ciphertexts. As in the availability proof, $\mathbf{A}'''_q$ has to guess on which query $(\tilde{m}_l, i_l)$ and with respect to which other index j the erroneous decryption will occur, for sending the appropriate $\tilde{m}_l$, i_l, and j as its challenge in the weak robustness game. In order to properly simulate the eavesdropper to the environment, we use a slightly tweaked version of weak robustness (equivalent to the original one) where we also obtain the generated ciphertext when running the **GameOutput** oracle.

IND-CCA**.** We introduce a second hybrid $\mathbf{H}_2$ that behaves as $\mathbf{H}_1$ but additionally leaks the receiver's identity (no anonymity). The suitable simulator σ_2 always encrypts the all-zero string of appropriate length for the respective user, and decrypts as needed. Two things must be shown:

first, that $\sigma_2^E\mathbf{H}_2$ is indistinguishable from the real-world system; and second, that $\sigma_1^E\mathbf{H}_1$ and $\sigma_2^E\mathbf{H}_2$ are indistinguishable. We start with the former one, where the reduction $\mathbf{A}'_q(\cdot)$ uses a hybrid argument to employ a distinguisher for $\sigma_2^E\mathbf{H}_2$ and $\mathsf{pke}\,[\prec, \longleftarrow\!\bullet^n]$ to win the IND-CCA game. Technically, one defines a sequence of hybrid systems, where the i-th hybrid simulates "ideal" encryptions for the first $i-1$ receivers, uses the game to simulate for the i-th receiver, and "real" encryptions for the remaining receivers. The reduction $\mathbf{A}'_q(\cdot)$ then chooses $i \in [n]$ uniformly at random. Overall, the first hybrid with no simulated encryptions is equivalent to $\mathsf{pke}\,[\prec, \longleftarrow\!\bullet^n]$, while the hybrid with only simulated encryptions is equivalent to the hybrid $\sigma_2^E\mathbf{H}_2$. As the IND-CCA game offers only a single challenge query, another hybrid argument must be employed to account for the number of encryptions; this adds a factor of q.

IK-CCA. The last step is to show a reduction $\mathbf{A}''_q(\cdot)$ that turns a distinguisher between $\sigma_1^E\mathbf{H}_1$ and $\sigma_2^E\mathbf{H}_2$ corresponding, respectively, to the first and second hybrid introduced in the proof, into an IK-CCA-adversary. Recall that $\mathbf{H}_2$ behaves just like $\mathbf{H}_1$ except that it does not grant anonymity. The first step is to show a reduction to 1-sided-IK-CCA, then we use Lemma 2.4 to complete the reduction to IK-CCA. We again use a hybrid argument with qn "intermediate" systems between $\sigma_1^E\mathbf{H}_1$ and $\sigma_2^E\mathbf{H}_2$, similarly to the IND-CCA case, such that each intermediate system embeds the challenge at a different position (as above). All other keys, encryptions, or decryptions are either simulated as "real" or as "ideal," depending on their position. The system where only the queries are "real" is equivalent to $\sigma_2^E\mathbf{H}_2$, and the system where only $\tilde{pk}$ was used (all queries are "ideal") is equivalent to $\sigma_1^E\mathbf{H}_1$. □

4 Relation to Notions of Robustness

While the confidential receiver-anonymous channel can be achieved using an encryption scheme that fulfills IND-CCA, IK-CCA, and WROB-CCA, anonymity without robustness is not sufficient. This was already noted by Abdalla et al. [ABN10], who point out that if one receiver obtains a ciphertext that was intended for a different receiver, the decryption should yield this information—by producing an error symbol—instead of an arbitrary, but well-formed plaintext, because this undetected, but unintended plaintext message might "upset" higher level protocols. This "robustness," however, is not guaranteed by IND-CCA or IK-CCA.

This section formalizes and proves statements related to robustness. In Section 4.1 we describe the type of channel one obtains if the public-key scheme is only IND-CCA- and IK-CCA-secure; this confirms the intuition given in [ABN10]. We then show in Section 4.2 that WROB-CCA is indeed formally *necessary* to construct the channel $\multimap\!\!\diamond\!\!\prec\!\!\bullet$: Every scheme that achieves the constructive notion will also be weakly robust (aside from being at least IND-RCCA and IK-RCCA-secure). Finally, in Section 4.3 we show that a strongly robust scheme will also *only* construct the resource $\multimap\!\!\diamond\!\!\prec\!\!\bullet$, albeit with a tighter reduction with respect to the security properties. We also explain why a strongly robust scheme does not help to construct a "qualitatively better" resource from the given ones.

4.1 Anonymity with Erroneous Transmission

The channel one obtains from applying an IND-CCA and IK-CCA-secure scheme to $\prec$ and $\longleftarrow\!\bullet^n$ is the resource $\multimap\!\!\$\!\!\prec\!\!\bullet$ which is parametrized by a family of distributions $\left(\mathsf{P}^{\ell}_{Y_1\ldots Y_n}\right)_{\ell\in\mathbb{N}}$ and differs from $\multimap\!\!\diamond\!\!\prec\!\!\bullet$ only in the cases where honestly generated messages are transmitted to receivers other than the intended one (either during an honest transmission or because the adversary forwards an honestly sent message to such a receiver). Without weak robustness, the

unintended receiver will output a message according to the (scheme-specific) distribution. A formal description of $\multimap\!\!\diamond\!\!\$\!\!\prec\!\!\bullet$ follows.

- If at the E-interface the $\perp$-converter is connected, then for the k-th input (m_k, i_k) at the A-interface, choose $m'_{k,1}, \ldots, m'_{k,n}$ according to $\mathsf{P}^{|m_k|}_{Y_1 \ldots Y_n}$, output m_k at B_{i_k} and $m'_{k,j}$ at B_j for $j \neq i_k$ (if $m'_{k,j} \neq \Diamond$, else nothing).
- Otherwise, on the k-th input (m_k, i_k) at the A-interface, output only the message length $|m_k|$ at the E-interface. (Again, if $i_k \in \mathcal{C}$ then the channel leaks (m_k, i_k) at the E-interface.) Furthermore, the E-interface allows the following queries:
  - $(\texttt{inject}, j, \tilde{m})$ for $j \in \{1, \ldots, n\}$ and $\tilde{m} \in \mathcal{M}$: Delivers $\tilde{m}$ at B_j,
  - $(\texttt{deliver}, j, \bar{k}, \tilde{m})$ for $j \in \{1, \ldots, n\}$, $\bar{k} \in \mathbb{N}$, and $m \in \mathcal{M}$: If at least $\bar{k}$ messages have been sent via A, then delivers message $m_{\bar{k}}$ at B_j if $i_{\bar{k}} = j$, and delivers $\tilde{m}$ at B_j otherwise.

The behavior of the channel is also depicted in Figure 11. On input a message m and an intended receiver i at the A-interface, the channel leaks only the message length $|m|$ at the E-interface (not the identity of the intended receiver). The E-interface allows the attacker to "deliver" such a message to any receiver. If the correct receiver is chosen (here i), the output is the correct message (here m), if a wrong receiver is chosen (here i'), the output is a different message (here $m'_{i'}$). Alternatively, the attacker can also inject messages to any receiver, in the figure this corresponds to the message $\tilde{m}$ sent to receiver i''.

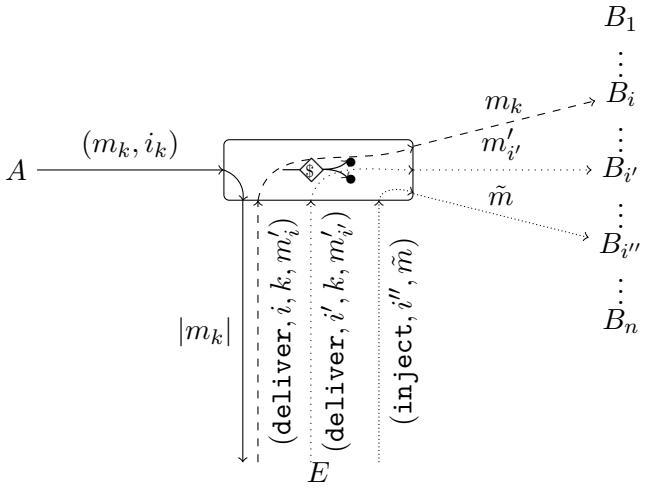


Figure 11: The receiver-anonymous channel with erroneous transmission, the value $m'_{i'}$ is chosen according to $\mathsf{P}^{|m_k|}_{Y_{i'}}$.

Theorem 4.1 below shows that the channel $\multimap\!\!\diamond\!\!\$\!\!\prec\!\!\bullet$ is constructed from $\prec\!\!\triangleleft$ and n authenticated channels $\longleftarrow\!\!\bullet$ if the encryption scheme is IND-CCA- and IK-CCA-secure. In the proof, we instantiate the channel $\multimap\!\!\diamond\!\!\$\!\!\prec\!\!\bullet$ with a distribution $\mathsf{P}^{\ell}_{Y_1 \ldots Y_n}$ that we define by honestly choosing keys for the receivers and, whenever a message is sent to a party B_i, decrypting a "random ciphertext" of the correct length with respect to the keys of all parties B_j with $j \neq i$.

Theorem 4.1. *An encryption scheme that is both* IND-CCA- *and* 1-sided-IK-CCA-*secure constructs* $\multimap\!\!\diamond\!\!\$\!\!\prec\!\!\bullet$ *from* $\prec\!\!\triangleleft$ *and* $(\longleftarrow\!\!\bullet)^n$ *with respect to static corruptions. More formally, for each*

$\mathcal{C} \subseteq [n]$, *there are a simulator* σ *and three reductions* $\mathbf{A}_q(\cdot)$, $\mathbf{A}'_q(\cdot)$, *and* $\mathbf{A}''_q(\cdot)$ *such that*

$$\Delta^{\mathbf{D}}\left(\mathsf{pke}\perp^{E}\left[\overset{q}{\prec\!\!\!\cdot},\longleftarrow\!\!\bullet^{n}\right],\perp^{E}(-\overset{q}{\diamond\!\!\!\cdot})\right) \leq qn\cdot\Phi^{\mathbf{A}_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{ind\text{-}cca}}\right), \tag{3}$$

and

$$\Delta^{\mathbf{D}}\left(\mathsf{pke}\left[\overset{q}{\prec\!\!\!\cdot}_{\mathcal{C}},\longleftarrow\!\!\bullet^{n}_{\mathcal{C}}\right],\sigma^{E}(-\overset{q}{\diamond\!\!\!\cdot}_{\mathcal{C}})\right) \leq qn\cdot\Phi^{\mathbf{A}'_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{ind\text{-}cca}}\right)+qn\cdot\Phi^{\mathbf{A}''_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}\right). \tag{4}$$

Proof sketch. We first sketch the proof of the availability condition (3). This proof has a similar structure as the one of Theorem 3.2. The first step is to define a distribution $\mathsf{P}^{\ell}_{Y_1\ldots Y_n}$ for $-\diamond\!\!\!\cdot$ as follows: initially choose (using the key generation algorithm) n public-/private-key pairs for the users. For the intended recipient, the message is delivered unchanged; each other recipient B_j obtains $m'_j \coloneqq \mathtt{dec}(sk_j;\tilde{c})$ for $\tilde{c} \coloneqq \mathtt{enc}(pk_i, 0^{|m|})$, or no output if $m'_j = \Diamond$. The reduction $\mathbf{A}_q(\cdot)$ then again uses a hybrid argument similar to the one in the proof of Theorem 3.2; intuitively, each intermediate system embeds the challenge query of the IND-CCA-game at one particular recipient and message (number).

Security. To show condition (4), we describe a simulator σ that emulates the adversary's interface of $\longleftarrow\!\!\bullet$ and $\prec\!\!\!\cdot$. This simulator behaves as follows:

- Generate $n-|\mathcal{C}|$ private-/public-key pairs (pk_i, sk_i) with $i \in [n]\backslash\mathcal{C}$ to simulate each pk_i that it is transmitted via the corresponding channel $\longleftarrow\!\!\bullet$. Furthermore, generate one auxiliary key pair $(\tilde{pk}, \tilde{sk})$. For all $j \in \mathcal{C}$, obtain a public key $\bar{pk}_i$ at the simulated E-interface of $\longleftarrow\!\!\bullet$.
- Upon receiving the k-th message length ℓ_k at the E-interface of $-\diamond\!\!\!\cdot$, generate a ciphertext $c_k = \mathtt{enc}(\tilde{pk}; 0^{\ell_k})$ and simulate c_k as a message on $\prec\!\!\!\cdot$. If the channel leaked a pair (m_k, i_k) (which means $i_k \in \mathcal{C}$), encrypt m_i using $\bar{pk}_{i_k}$ and simulate that ciphertext instead.
- When delivering a message $\tilde{c}$ to some user $j \in \{1,\ldots,n\}$:
  - In case $\tilde{c} = c_{\bar{k}}$ for some $\bar{k} \in \mathbb{N}$, compute $\tilde{m}_j \coloneqq \mathtt{dec}(sk_j;\tilde{c})$ and, if $\tilde{m}_j \neq \Diamond$, issue $(\mathtt{deliver}, j, \bar{k}, \tilde{m}_j)$ to $-\diamond\!\!\!\cdot$.
  - In case $\tilde{c}$ is "fresh," compute $\tilde{m}_j \coloneqq \mathtt{dec}(sk_j;\tilde{c})$, and, if $\tilde{m}_j \neq \Diamond$, issue $(\mathtt{inject}, j, \tilde{m}_j)$ to $-\diamond\!\!\!\cdot$.

Now, we have to show that if $(\mathtt{kgen}, \mathtt{enc}, \mathtt{dec})$ is a "good" encryption scheme, then applying it to $[\prec\!\!\!\cdot_{\mathcal{C}}, \longleftarrow\!\!\bullet^{n}_{\mathcal{C}}]$ results in a system that is indistinguishable from $\sigma^{E}(-\diamond\!\!\!\cdot_{\mathcal{C}})$. This is proven similar to Theorem 3.2, i.e., via reductions that win the IND-CCA or 1-sided-IK-CCA games, respectively. The proof follows exactly the same structure as the corresponding parts of the one for Theorem 3.2. □

4.2 "Equivalence" with Weak Robustness

In Section 3.4 we showed that IND-CCA, IK-CCA, and WROB-CCA security are *sufficient* to construct $-\diamond\!\!\!\cdot$. Indeed, (slightly weaker variants of) these properties are also *necessary*: If a PKE scheme is sufficient for the construction, then it must also be weakly robust, IND-RCCA,

and IK-RCCA. Note that "CCA"-notions are sufficient, but not necessary, as they also prohibit that a scheme allows for "trivial" modifications of the ciphertext, which do not have an impact on the actual security [CKN03, MRT12].

The formal statement and the proof of Theorem 4.2 are given below. The basic idea is that for each of the three games for weak robustness, IND-RCCA, and IK-RCCA, we show that a successful adversary will also serve as a good distinguisher in the constructive security statement.

Theorem 4.2. *Let* $(\mathsf{enc}, \mathsf{dec})$ *be a public-key encryption scheme that transforms* $[\longleftarrow\!\bullet^n, \prec]$ *into* $-\diamond\!\!\prec\!\!\bullet$*, such that* $\mathcal{M} = \bigcup_{l\in\mathcal{L}}\{0,1\}^l$ *with* $\mathcal{L} \subseteq \{l \in \mathbb{N} \mid l \geq \ell\}$ *for some* $\ell \in \mathbb{N}$*. Then there are a reduction* $\mathbf{R}_1(\cdot)$ *such that for any* q*-query* IND-RCCA*-adversary* $\mathbf{A}_1$ *we have*

$$\Phi^{\mathbf{A}_1}(\mathbf{G}^{\mathsf{ind\text{-}rcca}}) \leq 4\Delta^{\mathbf{R}_1(\mathbf{A}_1)}\left(\mathsf{enc}^A\mathsf{dec}^{B_1}\ldots\mathsf{dec}^{B_n}\left[\overset{2q}{\prec}, \longleftarrow\!\bullet^n\right], \sigma^E(-\overset{2q}{\diamond\!\!\prec\!\!\bullet})\right) + q^2/2^{\ell-2}, \quad (5)$$

a reduction $\mathbf{R}_2(\cdot)$ *such that for any* q*-query* IK-RCCA*-adversary* $\mathbf{A}_2$ *we have*

$$\Phi^{\mathbf{A}_2}(\mathbf{G}^{\mathsf{ik\text{-}rcca}}) \leq 4\Delta^{\mathbf{R}_2(\mathbf{A}_2)}\left(\mathsf{enc}^A\mathsf{dec}^{B_1}\ldots\mathsf{dec}^{B_n}\left[\overset{2q}{\prec}, \longleftarrow\!\bullet^n\right], \sigma^E(-\overset{2q}{\diamond\!\!\prec\!\!\bullet})\right) + q^2/2^{\ell-2}, \quad (6)$$

and a reduction $\mathbf{R}_3(\cdot)$ *such that for any* q*-query* WROB-CCA*-adversary* $\mathbf{A}_3$ *we have*

$$\Gamma^{\mathbf{A}_3}(\mathbf{G}^{\mathsf{w\text{-}rob}}) \leq \bar{N}^2 \cdot \Delta^{\mathbf{R}_3(\mathbf{A}_3)}\left(\mathsf{enc}^A\mathsf{dec}^{B_1}\ldots\mathsf{dec}^{B_n}\left[\overset{2q}{\prec} \longleftarrow\!\bullet^n\right], \sigma^E(-\overset{2q}{\diamond\!\!\prec\!\!\bullet})\right) + \bar{N}^2 \cdot q^2/2^{\ell}, \quad (7)$$

where $\mathbf{A}_3$ *invokes at most* $\bar{N}$ *users.*

Proof sketch. The main idea of the proof is to show that any of the adversaries $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ can be turned into a distinguisher for the constructive security statement, i.e, distinguishing the real-world system from the ideal-world system. To prove this, we first construct a particular simulator, which is "universal" in the sense that if there is *any* simulator such that the real-world system and ideal-world system are indistinguishable, then *this particular* simulator will be (nearly) as good. After describing the simulator and proving that it is universal, we describe reductions $\mathbf{R}_i(\cdot)$ such that for any adversary $\mathbf{A}_i$ (here $i = 1, \ldots, 3$), it holds that $\mathbf{R}_i(\mathbf{A}_i)$ is a good distinguisher between the real resource and the universal simulator.

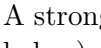
The simulator we describe is denoted σ_{pk_1}, since it always uses the key pk_1 to encrypt. This simulator behaves exactly as the one in Theorem 3.2, except: (1) upon receiving message length l', it encrypts $0^{l'-l}||x$ with $x \in \{0,1\}^l$ chosen uniformly at random; (2) on receiving a ciphertext, the simulator decrypts with sk_1, considers it a replay iff decryption succeeds and the obtained plaintext $0^{l_1}||x'$ was encrypted by the simulator before, (in this case, σ_{pk_1} conditionally forwards the respective message to the chosen user), and, if the message is not a replay, the ciphertext is decrypted with the key of the indicated recipient and injected into the channel.

To prove that σ_{pk_1} is universal, we describe a reduction $\mathbf{R}$ that has a "forward" and a "random" mode. $\mathbf{R}$ connects to either the real-world system or the ideal-world system (like a distinguisher). In "forward" mode, $\mathbf{R}$ only relays all inputs and outputs. In "random" mode, however, it will replace plaintexts in exactly the same manner as σ_{pk_1}. The reduction $\mathbf{R}$ chooses either "forward" or "random" mode with probability $\frac{1}{2}$ each, and in "random" mode inverts the distinguisher's guess. $\mathbf{R}$ turns any distinguisher that achieves with q queries an advantage ε between the real-world system and the ideal-world system with simulator σ_{pk_1} into a distinguisher between the real-world system and the ideal-world system with an arbitrary simulator. The latter distinguisher makes $2q$ queries and has an advantage of $\frac{\varepsilon}{2} - \frac{q^2}{2^l}$.

Now we describe reductions $\mathbf{R}_1, \mathbf{R}_2$, and $\mathbf{R}_3$ as follows. Reductions $\mathbf{R}_1$ and $\mathbf{R}_2$ use adversaries $\mathbf{A}_1, \mathbf{A}_2$ against IND-RCCA resp. IK-RCCA-security. The reduction $\mathbf{R}_1$ outputs pk_1 as a key to $\mathbf{A}_1$, while $\mathbf{R}_2$ outputs pk_1, pk_2 to $\mathbf{A}_2$. The reduction $\mathbf{R}_1$ handles $\mathbf{A}_1$'s challenge query (m_0, m_1) by transmitting m_b for a random bit b to B_1 and returning the ciphertext to $\mathbf{A}_1$; $\mathbf{R}_2$ sends the given message to either B_1 or B_2, depending on a random bit u, and also returns the result. The output of each $\mathbf{A}_i$ for $i = 1, 2$ is denoted d_i, and the reductions output $b \oplus d_1$ or $u \oplus d_2$, respectively. If $\mathbf{R}_i$ interacts with the real-world system in either case, it emulates the underlying game, IND-RCCA and resp. IK-RCCA; else, if it interacts with the ideal-world system with simulator σ_{pk_1} the view of $\mathbf{A}_i$ is statistically independent of the bit b resp. u, which shows equations (5) and (6).

The last reduction $\mathbf{R}_3$ runs a WROB-CCA-adversary $\mathbf{A}_3$; it has to choose two users from the set of $\bar{N}$ receivers to associate keys pk_1 and pk_2 with (the other users get "fresh" simulated key pairs). Thus, $\mathbf{R}_3$ must guess for which users robustness will be violated (losing a factor $\bar{N}^2$)—if $\mathbf{A}_3$ corrupts either of these users or runs the **GameOutput** oracle for users other than the selected ones, then $\mathbf{R}_3$ will output a random bit. Ciphertexts are decrypted with the simulated keys for all but the guessed indices, and else forwarded to the selected users. If the guessed users are correct (i.e. neither are corrupted and they are used as input for the **GameOutput** oracle) then the message input to **GameOutput** is forwarded to the first input user, and the ciphertext is forwarded to both users. The output is either 0, if both users output messages, and 1 otherwise. This results in equation (7). □

4.3 Anonymity with Strong Robustness

Strong robustness (SROB-CCA, [ABN10]) is strictly stronger than weak robustness. Intuitively, whereas weak robustness states that honestly generated ciphertexts are not decryptable by two distinct receivers, strong robustness requires this even for adversarially generated ciphertexts. A strongly robust scheme will of course also be sufficient to achieve  , in Theorem 4.3 (see below) we even achieve better bounds in the reduction. Intuitively, due to the exact definition of the oracles in the games, the reduction to SROB-CCA can exploit *every* inconsistency in an emulated interaction with the distinguisher, whereas the reduction to WROB-CCA has to guess *when* the inconsistency will occur.

Somewhat surprisingly, strong robustness does not provide a "qualitatively" better security guarantee than weak robustness. ("Qualitative" refers to the properties of the resources, in contrast to the "quantitative" reduction tightness.) This is particularly relevant since obtaining a WROB-CCA secure scheme from a non-robust one is easier than obtaining an SROB-CCA one [ABN10].

To some extent, the fact that the "qualitative" guarantees of weak and strong robustness coincide stems from the assumed resource $\prec$. Since $\prec$ allows the adversary to inject arbitrary ciphertexts to arbitrary receivers, there is no incentive to send *the same* (faked) ciphertext to two or more different users; the adversary could also send different ciphertexts. Technically, from a network that allows the adversary to inject one message to multiple receivers more "easily" than it allows him to inject different messages, a strongly robust scheme indeed achieves a "better" resource than a weakly robust one; in the weakly robust case the adversary can inject messages to *several* receivers "easily," in the strongly robust case *only to one*. We think, however, that such a network guarantee (injecting several different messages is more difficult) would have to be justified by a particular application and should not be the focus of a general-purpose discussion as ours.

Theorem 4.3. *An encryption scheme that is* IND-CCA-, 1-sided-IK-CCA-, *and* SROB-CCA-*secure constructs* *from* $\prec$ *and* $(\longleftarrow\!\bullet)^n$ *with respect to static corruption. More formally,*

for each $\mathcal{C} \subseteq [n]$*, there exist a simulator* σ' *and reductions* $\mathbf{A}_{nq}(\cdot)$*,* $\mathbf{A}'_q(\cdot)$*,* $\mathbf{A}''_q(\cdot), \mathbf{A}'''_{nq}(\cdot)$ *such that*

$$\Delta^{\mathbf{D}}\left(\mathsf{pke}\perp^E\left[\overset{q}{\prec}, \longleftarrow\!\bullet^n\right], \perp^E(-\overset{q}{\diamond}\!\!\prec\bullet)\right) \leq \Gamma^{\mathbf{A}_{nq}(\mathbf{D})}\left(\mathbf{G}^{\mathsf{s\text{-}rob}}\right), \tag{8}$$

and

$$\Delta^{\mathbf{D}}\left(\mathsf{pke}\left[\overset{q}{\prec}\mathcal{C}, \longleftarrow\!\bullet^n_{\mathcal{C}}\right], \sigma'^E(-\overset{q}{\diamond}\!\!\prec\bullet_{\mathcal{C}})\right)$$
$$\leq qn \cdot \Phi^{\mathbf{A}'_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{ind\text{-}cca}}\right) + qn \cdot \Phi^{\mathbf{A}''_q(\mathbf{D})}\left(\mathbf{G}^{\mathsf{1\text{-}sided\text{-}ik\text{-}cca}}\right) + \Gamma^{\mathbf{A}'''_{nq}(\mathbf{D})}\left(\mathbf{G}^{\mathsf{s\text{-}rob}}\right). \tag{9}$$

Proof sketch. The reductions we describe for this proof are less efficient in terms of queries than our other reductions; however, they are tighter than the proofs using WROB-CCA. The proof of equation (8) follows that of Theorem 3.2; we use the same MES (which becomes true once a given pair (m, i) provokes that one user other than i outputs a message), and the real-world and ideal-world systems are equivalent as long as the MES is false. Yet, using WROB-CCA, we lost a factor qn as we had to guess the correct query and user for breaking robustness in advance; this was because the **GameOutput** oracle takes as input the plaintext m which has to be honestly encrypted. Using SROB-CCA, we can simply encrypt each message m sent by the distinguisher via the A-interface, and decrypt it using the keys of all the receivers other than the intended one. If any of these decryptions result in $m' \neq \Diamond$, then the ciphertext can be used in the SROB-CCA **GameOutput**-oracle.

To show equation (9), we use the simulator σ in Theorem 3.2, the simulator (call it σ' here) and reductions $\mathbf{A}'_q$ and $\mathbf{A}''_q$ of Eq. (4) in Theorem 4.1, and a reduction $\mathbf{A}'''_{nq}$ which we sketch below, turning a successful distinguisher between σ'^E–◇\$⊸• and σ^E–◇?⊸• into an adversary against SROB-CCA. This last part is done by means of the MES used in Theorem 3.2, which becomes true if a simulated message c_k sent to a user that was not its intended recipient, yields output at this user. We argue that σ^E–◇?⊸• and σ'^E–◇\$⊸• are equivalent unless if the MES is true, and show that MES becomes true only if SROB-CCA is broken. The reduction $\mathbf{A}'''_{nq}$ creates $n+1$ key pairs in the SROB-CCA-game (n for users and 1 auxiliary key $\tilde{pk}$ to emulate the outputs of σ^E–◇?⊸• and σ'^E–◇\$⊸•). "Fresh" ciphertexts are decrypted using the corresponding decryption oracles; if the MES becomes true and one non-fresh ciphertext results $m_j \neq \Diamond$ for some receiver other than the intended one, the SROB-CCA-game can easily be won. □

5 Conclusion and Possible Extensions

We analyzed the problem of achieving confidentiality for a receiver-anonymous channel; our results are the constructive counterpart of the notions discussed in [BBDP01, ABN10]. In particular, we showed that confidentiality, key privacy, and weak robustness are indeed sufficient for such a scheme to be useful, and that (slightly relaxed versions of) these are indeed necessary. We have also discussed why strong robustness is not necessary in this context. Our results do not only support the trust in existing schemes and constructions; they also show that the simpler and more efficient weakly robust schemes (see [ABN10]) can be used safely.

Our constructive statements help explore the boundary between cryptography and traffic analysis. For example, an (active) instance of the latter, conditional delivery, cannot be prevented by end-to-end encryption (even if it has all the properties we suggest); indeed countermeasures against such attacks at the application level are critical to provide any meaningful traffic analysis resistance. Our ideal resource, thus, does not yet correspond directly to the black-box system models used by traffic analysis research, but is a component upon which such a model could be based. In contrast to our model here, traffic analysis frameworks usually

consider restricted attackers that observe only parts of the system and a probabilistic model for sender and receiver behavior.[9]

Protocols in which encrypted messages are processed by multiple parties can, to some extent, prevent conditional deliveries. In a MIX-network, for instance, the attacker cannot direct a multi-layered ciphertext at a particular recipient, as he will be unable to remove the outer ciphertext layers. Thus receiver-anonymous communication using onions, threshold decryption, or verifiable re-randomization bypasses our impossibility result, instead requiring additional (distributed) trust in third parties.

Our study of anonymity properties of end-to-end encryption is only a first step in the constructive modeling of resources with useful anonymity properties and constructions thereof. The general paradigm of examining the security of anonymity-preserving cryptographic schemes in a constructive manner can (and should) be applied to other schemes as well, including topics such as anonymous signature schemes and key-exchange protocols.

Acknowledgments

Ueli Maurer and Björn Tackmann were supported by the Swiss National Science Foundation (SNF), project no. 200020-132794. Daniele Venturi acknowledges support from the Danish National Research Foundation, the National Science Foundation of China (under the grant 61061130540), the Danish Council for Independent Research (under the DFF Starting Grant 10-081612) and also from the CFEM research center within which part of this work was performed.

We'd like to thank Martin Hirt for pointing out that the proceedings version of our work did not specify static corruption of receivers. We'd also like to thank the anonymous reviewers for their helpful comments.

References

[Aba02] Martín Abadi. Private authentication. In *Privacy Enhancing Technologies*, volume 2482 of *LNCS*, pages 27–40. Springer, 2002.

[ABN10] Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In *TCC*, volume 5978 of *LNCS*, pages 480–497. Springer, 2010.

[AF04] Martín Abadi and Cédric Fournet. Private authentication. *Theor. Comput. Sci.*, 322(3):427–476, 2004.

[BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT*, volume 2248 of *LNCS*, pages 566–582. Springer, 2001.

[BD03] Amos Beimel and Shlomi Dolev. Buses for anonymous message delivery. *Journal of Cryptology*, 16(1):25–39, 2003.

[BGKM12] Michael Backes, Ian Goldberg, Aniket Kate, and Esfandiar Mohammadi. Provably secure and practical onion routing. In Stephen Chong, editor, *CSF*, pages 369–385. IEEE, 2012.

[9] In this aspect, our analysis plays a role similar to the cryptographic analysis of an onion routing protocol in [BGKM12], which provides a formal foundation for the traffic analysis of onion routing in [FJS12]. Our analysis targets a cryptographic primitive and not a protocol, and can thus be more detailed.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *EUROCRYPT*, volume 4004 of *LNCS*, pages 409–426. Springer, 2006.

[CK02] Ran Canetti and Hugo Krawczyk. Security analysis of IKE's signature-based key-exchange protocol. In *CRYPTO*, volume 2442 of *LNCS*, pages 27–52. Springer, 2002.

[CKN03] Ran Canetti, Hugo Krawczyk, and Jesper Nielsen. Relaxing chosen-ciphertext security. In *CRYPTO*, volume 2729 of *LNCS*, pages 262–582. Springer, 2003.

[CL05] Jan Camenisch and Anna Lysyanskaya. A formal treatment of onion routing. In *CRYPTO*, volume 3621 of *LNCS*, pages 169–187. Springer, 2005.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[Fis07] Marc Fischlin. Anonymous signatures made easy. In *PKC*, volume 4450 of *LNCS*, pages 31–42. Springer, 2007.

[FJS07] Joan Feigenbaum, Aaron Johnson, and Paul F. Syverson. A model of onion routing with provable anonymity. In *Financial Crypto*, volume 4886 of *LNCS*, pages 57–71. Springer, 2007.

[FJS12] Joan Feigenbaum, Aaron Johnson, and Paul F. Syverson. Probabilistic analysis of onion routing in a black-box model. *ACM Trans. Inf. Syst. Secur.*, 15(3):14, 2012.

[FLPQ13] Pooya Farshim, Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Robust encryption, revisited. In *PKC*, pages 352–368. Springer, 2013.

[HM08] Alejandro Hevia and Daniele Micciancio. An indistinguishability-based characterization of anonymous channels. In *PETS*, volume 5134 of *LNCS*, pages 24–43. Springer, 2008.

[IKOS06] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *FOCS*, pages 239–248. IEEE Computer Society, 2006.

[Mau02] Ueli Maurer. Indistinguishability of random systems. In *EUROCRYPT*, volume 2332 of *LNCS*, pages 110–132. Springer, 2002.

[Moh10] Payman Mohassel. A closer look at anonymity and robustness in encryption schemes. In *ASIACRYPT*, volume 6477 of *LNCS*, pages 501–518. Springer, 2010.

[MR11] Ueli Maurer and Renato Renner. Abstract cryptography. In *Innovations in Computer Science*. Tsinghua University Press, 2011.

[MRT12] Ueli Maurer, Andreas Rüedlinger, and Björn Tackmann. Confidentiality and integrity: A constructive perspective. In *TCC*, LNCS. Springer, 2012.

[MS96] Ueli Maurer and Pierre Schmid. A calculus for security bootstrapping in distributed systems. *Journal of Computer Security*, 4(1):55–80, 1996.

[MT10] Ueli Maurer and Björn Tackmann. On the soundness of Authenticate-then-Encrypt: Formalizing the malleability of symmetric encryption. In *ACM CCS*. ACM, 2010.

[NMO08] Waka Nagao, Yoshifumi Manabe, and Tatsuaki Okamoto. Relationship of three cryptographic channels in the UC framework. In *ProvSec*, volume 5324 of *LNCS*, pages 268–282. Springer, 2008.

[OV11] Cristina Onete and Daniele Venturi. Security & indistinguishability in the presence of traffic analysis. Cryptology ePrint Archive, Report 2011/260, 2011.

[PW85] Andreas Pfitzmann and Michael Waidner. Networks without user observability. In *EUROCRYPT*, pages 245–253, 1985.

[WFS03] Brent R. Waters, Edward W. Felten, and Amit Sahai. Receiver anonymity via incomparable public keys. In *ACM CCS*, pages 112–121, 2003.

[YWDW06] Guimin Yang, Duncan S. Wong, Xiaotie Deng, and Hauxiong Wang. Anonymous signature schemes. In *PKC*, volume 3958 of *LNCS*, pages 347–363. Springer, 2006.

A The Composition Theorem

We describe the composition of protocols and formulate the composition theorem in constructive cryptography. Formally, for multiple converters $\psi_{(1)}, \ldots, \psi_{(m)}$ we write $[\psi_{(1)}, \ldots, \psi_{(m)}]$ for the parallel composition, and it generally holds that

$$[\psi_{(1)}, \ldots, \psi_{(m)}]^I[\mathbf{R}_1, \ldots, \mathbf{R}_m] = [\psi_{(1)}^I \mathbf{R}_1, \ldots, \psi_{(m)}^I \mathbf{R}_m]$$

for all $\mathbf{R}_1, \ldots, \mathbf{R}_m$. Moreover, any two converters $\psi, \phi \in \Sigma$ can be composed sequentially by connecting the inside interface of the one converter ψ to the outside interface of ϕ. This is denoted as $\psi \circ \phi$ and it holds that $(\psi \circ \phi)^I \mathbf{R} = \psi^I(\phi^I \mathbf{R})$ for all resources $\mathbf{R}$. We extend both notations to protocols, i.e., we write $\boldsymbol{\psi} \circ \boldsymbol{\pi}$ or $[\boldsymbol{\pi}_{(1)}, \ldots, \boldsymbol{\pi}_{(m)}]$ and mean that the respective operations apply to all converters individually. We also make use of a special converter id that behaves transparently (i.e., allows access to the underlying interface of the resource). The protocol where all parties have to converter id is denoted $\mathbf{id}$.

This composition theorem here resembles the one in [MT10], but is phrased such that it applies to settings where one does not assume that the distinguisher class is closed under absorption of converters or resources, such as concrete security notions. The proof follows the same steps as the one in [MT10]. For the statement of the theorem we assume the operation $[\cdot, \ldots, \cdot]$ to be left-associative; in this way we can simply express multiple resources using the single variable $\mathbf{U}$.

Theorem A.1 (Composition for the n-party setting with explicit adversary)**.** *Let $\mathbf{R}$, $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{U}$ be resources. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ and $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_n)$ be protocols (such that $\boldsymbol{\pi}$ is intended to construct $\mathbf{S}$ from the resource $\mathbf{R}$ and $\boldsymbol{\psi}$ is intended to construct $\mathbf{T}$ from $\mathbf{S}$).*

For each distinguisher $\mathbf{D}$, denote by $\mathbf{D}'$ the distinguisher that runs $\mathbf{D}$ but emulates ψ_i at interface I_i for all $i \in [n]$, and by $\mathbf{D}''$ the distinguisher that runs $\mathbf{D}$ and emulates σ_π at interface E. Then, for all $\mathbf{D}$,

$$\begin{aligned}\Delta^{\mathbf{D}}\left((\boldsymbol{\psi} \circ \boldsymbol{\pi})\mathbf{R}, (\sigma_\pi \circ \sigma_\psi)^E \mathbf{T}\right) &\leq \Delta^{\mathbf{D}'}\left(\boldsymbol{\pi}\mathbf{R}, \sigma_\pi^E \mathbf{S}\right) + \Delta^{\mathbf{D}''}\left(\boldsymbol{\psi}\mathbf{S}, \sigma_\psi^E \mathbf{T}\right), \text{ and} \\ \Delta^{\mathbf{D}}\left(\perp^E (\boldsymbol{\psi} \circ \boldsymbol{\pi})\mathbf{R}, \perp^E \mathbf{T}\right) &\leq \Delta^{\mathbf{D}'}\left(\perp^E \boldsymbol{\pi}\mathbf{R}, \perp^E \mathbf{S}\right) + \Delta^{\mathbf{D}}\left(\perp^E \boldsymbol{\psi}\mathbf{S}, \perp^E \mathbf{T}\right).\end{aligned}$$

For each distinguisher $\mathbf{D}$, let $\mathbf{D}'''$ be the distinguisher that runs $\mathbf{D}$ and additionally emulates a concurrent execution of $\mathbf{U}$. Then, for all $\mathbf{D}$,

$$\begin{aligned}\Delta^{\mathbf{D}}\Big([\boldsymbol{\pi}, \mathbf{id}][\mathbf{R}, \mathbf{U}], [\sigma_\pi, \mathsf{id}]^E[\mathbf{S}, \mathbf{U}]\Big) &\leq \Delta^{\mathbf{D}'''}\left(\boldsymbol{\pi}\mathbf{R}, \sigma_\pi^E \mathbf{S}\right), \text{ and} \\ \Delta^{\mathbf{D}}\Big(\perp^E[\boldsymbol{\pi}, \mathbf{id}][\mathbf{R}, \mathbf{U}], \perp^E[\mathbf{R}, \mathbf{U}]\Big) &\leq \Delta^{\mathbf{D}'''}\left(\perp^E \boldsymbol{\pi}\mathbf{R}, \perp^E \mathbf{S}\right).\end{aligned}$$

The similar argument holds with respect to $[\mathbf{id}, \boldsymbol{\pi}]$, $[\mathbf{U}, \mathbf{S}]$, and $[\mathbf{U}, \mathbf{R}]$.

If one considers classes of distinguishers that are closed under composition with converters, that is $\mathcal{D} \circ \Sigma \subseteq \mathcal{D}$, and π constructs $\mathbf{S}$ from the resource $\mathbf{R}$ within ε_1 and ψ constructs $\mathbf{T}$ from $\mathbf{S}$ within ε_2, then $\boldsymbol{\psi} \circ \boldsymbol{\pi}$ constructs $\mathbf{T}$ from $\mathbf{R}$ within $\varepsilon_1 + \varepsilon_2$, $[\boldsymbol{\pi}, \mathbf{id}]$ constructs $[\mathbf{S}, \mathbf{U}]$ from $[\mathbf{R}, \mathbf{U}]$ within ε_1, and $[\mathbf{id}, \boldsymbol{\pi}]$ constructs $[\mathbf{U}, \mathbf{S}]$ from $[\mathbf{U}, \mathbf{R}]$ within ε_1.