Under consideration for publication in Math. Struct. in Comp. Science

Quantifying Information Flow in Cryptographic Systems

MICHAEL BACKES[1] and BORIS KÖPF[2]

[1] *Saarland University and MPI-SWS, Saarbrücken, Germany.*
[2] *IMDEA Software Institute, Madrid, Spain.*

We provide a novel definition of quantitative information flow, called transmissible information, that is suitable for reasoning about informational-theoretically secure (or non-cryptographic) systems, as well as about cryptographic systems with their polynomially bounded adversaries, error probabilities, etc. Transmissible information captures deliberate communication between two processes, and it safely over-approximates the quantity of information that a process unintentionally leaks to another process.

We show that transmissible information is preserved under universal composability, which constitutes the prevalent cryptographic notion of a secure implementation. This result enables us to lift quantitative bounds of transmissible information from simple ideal functionalities of cryptographic tasks to actual cryptographic systems.

We furthermore prove a connection between transmissible information in the unconditional setting and channel capacity, based on the weak converse of Shannon's coding theorem. This connection enables us to compute an upper bound on the transmissible information for a restricted class of protocols, using existing techniques from quantitative information flow.

1. Introduction

Preserving the confidentiality of sensitive data is one of the fundamental goals of many security protocols. However, in some scenarios, the partial release of sensitive data is acceptable, or even unavoidable: a password checker necessarily reveals some information about the secret password, the length of a ciphertext reveals partial information about the plaintext, etc. Intuitively, these protocols are still secure as long as only a small quantity of secret information is revealed. For example, entering a wrong password in the password checker allows an attacker to exclude a single password candidate, which reduces the attacker's initial uncertainty about the secret password by a small quantity. For more sophisticated programs, it is a challenge to establish quantitative bounds on the secret information they reveal.

There has been remarkable progress on defining and computing bounds for the amount of leaked information for different models of computation (see the section on related work). However, the existing approaches are not directly applicable for quantifying information-flow in cryptographic systems because they use information-theoretic notions of entropy as a measure for quantifying leakage. The security guarantees associated with those measures hold for attackers with unbounded computational resources; they are of limited use for analyzing protocols with cryptographic primitives and their complexity-theoretic security guarantees: On the one hand, information-theoretic approaches would vastly over-approximate the amount of information leaked by modern cryptographic primitives, e.g., although a public-key encryption intuitively does not reveal any information about its plaintext, from an information-theoretic perspective it already contains *all* information about the plaintext. On the other hand, cryptographic messages might intuitively carry more information than expected from the perspective of information theory; e.g., if a computationally bounded adversary knows a certain number of public-key encryptions, but not the plaintexts and the secret key, then no information about these plaintexts is revealed. However, if the adversary receives the decryption key during a protocol run, he can decrypt all ciphertexts. In this way, releasing a small amount of information in the information-theoretic sense (the key) may trigger the release of a large amount of information in the computational sense (the plaintexts). Any meaningful attempt to define quantitative information flow in cryptographic settings must account for such scenarios, and any such definition should be general enough to reason about common cryptographic settings.

Furthermore, the definition of quantitative information flow for cryptographic settings should be complemented with a general methodology for establishing quantitative bounds on the information flow. This methodology should link this definition to information-theoretic foundations and their tools for quantifying information flow, and provide suitable techniques for concisely determining this quantification for cryptographic systems with all their idiosyncrasies, such as error probabilities, computational restrictions, their reactive execution with surrounding protocols, etc.

1.1. *Our contribution*

We make the following contributions to this problem space: (1) We define transmissible information, a novel notion of quantitative information flow for general reactive, cryptographic settings; (2) we show that transmissible information is preserved under universal composability, which constitutes the prevalent cryptographic notion of a secure implementation; (3) we prove a connection between transmissible information and the information-theoretic notion of channel capacity. We moreover illustrate by means of a simple, public-key encryption-based example how these results can be combined for deriving quantitative security guarantees.

Novel definitions of quantitative information flow. We present a definition of information flow, which we call *transmissible information.* The definition has an unconditional case that is suitable for reasoning about information-theoretically secure or non-cryptographic systems, as well as a computational case that allows for reasoning about cryptographic settings with their polynomially bounded adversaries, error probabilities, etc. Our definition captures the maximal probability that one process (called the high user) correctly transmits a given number of bits to another process (called the low user), using the considered cryptographic protocol as a means of communication. Roughly, we maximize over all possible different behaviors of the high user and the low user, and we determine the probability that the low user correctly determines the bits that were initially only given to the high user. Imposing no computational constraints on the behaviors of these users yields the unconditional case of the definition; the complexity-theoretic case of the definition is obtained by only maximizing over those behaviors that belong to the considered complexity class; e.g., we require both users to run in probabilistic polynomial time. Transmissible information captures deliberate communication between the two processes, and it safely over-approximates the quantity of information that a process unintentionally leaks to another process. In this way, transmissible information can be used for expressing security against covert channels and against information leaks.

Preservation under universal composability. We show that transmissible information in both unconditional and computational settings is preserved under universal composability, which constitutes the prevalent cryptographic notion of a secure implementation: since the invention of the UC framework (Can01) and the related Reactive Simulatability (RSIM) framework, a vast amount of subsequent works have built upon these frameworks to establish the security of novel cryptographic constructions by means of comparing them to corresponding ideal functionalities. Preservation means that securely implementing an ideal functionality by a cryptographic primitive in the sense of universal composability does not increase the probability of correctly transmitting a given number of bits in the unconditional case, and only by a negligible quantity in the computational case. This preservation theorem makes it substantially easier to place a bound on the quantity of transmissible information in a cryptographic system, since universal composability helps to eliminate cryptography-related details such as error probabilities and complexity-theoretic restrictions. Moreover, the preservation theorem allows us to directly benefit from the increasing number of ideal functionalities that have been shown to have a secure cryptographic implementation, and enables the seamless integration of our definition with state-of-the-art compositional security proofs of cryptographic protocols.

Relationship to channel capacity. Third, we use the weak converse of Shannon's coding theorem to establish a formal connection between transmissible information in the unconditional setting and channel capacity. This connection enables us to compute an upper bound on the transmissible information for a restricted class of protocols, using existing automated techniques from quantitative information flow. Since a bound on transmissible information for unbounded settings is by definition also a bound for computational settings, this result enables existing techniques from information theory to derive meaningful computational bounds as well.

1.2. *Outline*

Section 2 gives an overview on how the results established in the subsequent sections can be combined to obtain bounds on the transmissible information for cryptographic settings. Section 3 briefly reviews the Sequential Probabilistic Process Calculus (SPPC), which we use to state our definitions and results. Section 4 presents the definition of transmissible information and shows that it is preserved under universal composability.

```
input $msg \in \{0,1\}^{\leq m}$
begin
    $b \leftarrow L$
        if $b = 0$
            output $loss$
        else
            output $E_{pk}(msg)$
    return
end
```

Fig. 1. The example protocol

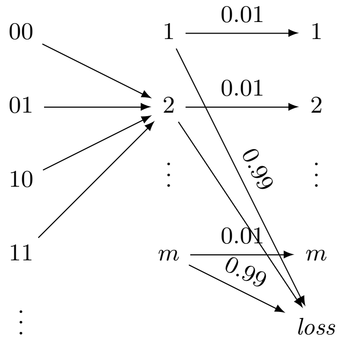

Fig. 2. The corresponding information-theoretic channel

Section 5 presents the connection between transmissible information in the unconditional setting and channel capacity. Section 6 shows in detail how the example protocol can be analyzed using our results. Section 7 discusses related work. Section 8 concludes.

2. Illustrative Example

In this section, we illustrate how the notion of transmissible information proposed in this paper can be used to leverage existing techniques from quantitative information flow for the quantitative analysis of cryptographic protocols. For this, we analyze the following simple example protocol. Consider an agent A whose only means of communication is a secure (i.e. encrypted) channel to an agent B. Consider further an agent C who can observe a certain fraction of the encrypted packets sent from A to B. Suppose that – although there is no legitimate communication channel between A and C – A wants to transmit a message to C. A can encode the message to C by varying the lengths of the messages sent to B. As C can observe the ciphertexts, he can estimate these lengths and hence decode the message. Intuitively, this is the only possible way for A to transmit information to C, since the encryption will hide everything except for the lengths of the plaintexts.

We model the channel from A to C as the protocol depicted in Figure 1. The protocol expects as input a message *msg* of length at most m and outputs either *loss* or the encryption of *msg* under a public key *pk*; we assume that the corresponding secret key is only known to the protocol. E denotes a semantically secure public key encryption, i.e., an encryption scheme that is secure under passive attacks (GM84). The command $b \leftarrow L$ denotes that b is chosen according to a distribution L; the probability of choosing $b = 0$ corresponds to the probability of a packet not being observed by C, which we denote by p_{loss}. Our goal is to derive a limit on the information that can be transmitted from A to C. We provide such a limit in terms of the probability p of C correctly recovering l bits from n runs of the program. For illustration purposes, we choose $l = 1000$, $n = 500$, $m = 1024$, and $p_{loss} = 0.99$. The analysis of the protocol proceeds in two main steps, which we illustrate below; these two steps exploit results that we formalize and prove in the remainder of this paper.

In the first step, we determine the channel capacity of the *core protocol* – a stripped-down version of the initial protocol – where all encryption operations are replaced by constants corresponding to different message lengths; i.e., the protocol outputs the length $|msg| \in \{1, \ldots, m\}$ instead of $E_{pk}(msg)$. We use this core protocol to determine the information-theoretic characteristics of the original protocol. The core protocol corresponds to the information-theoretic channel of Figure 2, where a message msg is first mapped to $|msg|$ and then, with probability p_{loss}, to $loss$. The left-hand side of the channel is deterministic, and the right-hand side of the channel constitutes an m-ary erasure channel, which is known to have a capacity of $c := (1 - p_{loss}) \log m$, see (DG06; CT06). For $p_{loss} = 0.99$ and $m = 1024$ in our example, the channel's capacity thus amounts to 0.1. The data processing inequality then shows that this capacity bound remains valid even if we perform arbitrary computations with the channel's output, as long as they are independent of the channel's input. In particular, this holds true if we replace i by $E_{pk}(0^i)$, i.e., by the encryption of a string of zeros of length i. In this work, we establish a theorem that uses capacity to bound the probability p that l bits can be transmitted in n runs:

$$p \leq \frac{c \cdot n + 1}{l}$$

For our parameters, we obtain $p \leq 0.051$ for transmitting $l = 1000$ bits in $n = 500$ protocol runs. Channel-capacity based bounds on the transmissible information are safe (but not necessarily tight) over-approximations of the transmissible information, see also Section 5. Moreover, they can be derived using existing tools from information theory and quantitative information flow.

In the second step, we show how this information-theoretic bound translates to a setting with real payload data and computationally bounded adversaries. This requires us to formalize and relate the notions of transmissible information in information-theoretic and computationally bounded settings. After that, we exploit the semantic security of the encryption scheme to conclude that $E_{pk}(0^{|msg|})$ and $E_{pk}(msg)$ are indistinguishable for probabilistic polynomial-time attackers. As a consequence, using the compositionality of our underlying framework, a protocol that outputs $E_{pk}(0^{|msg|})$ is indistinguishable from a protocol that outputs $E_{pk}(msg)$. In this work, we establish a novel preservation theorem that shows that transmissible information is preserved under universal composability. This means that the probability of correctly transmitting l bits in n protocol runs is upper bounded by p plus a negligible quantity in the security parameter.

3. A primer of SPPC

In this section, we briefly review the Sequential Probabilistic Process Calculus (SPPC) (DKMR08). SPPC is a calculus for reactive systems with a probabilistic execution model, providing universal composability properties while including computational aspects as needed for cryptography. SPPC provides a compact syntax for describing communicating machines and their interaction, and it is flexible enough to concisely capture different security notions, without the notational clutter of more concrete models. Its simplicity and its holistic approach to defining security properties makes SPPC an ideal foundation for our definition of quantitative information flow in cryptographic settings, which we present in Section 4. While our results will be formulated over SPPC, they can be easily recast in other reactive security models such as the Reactive Simulatability (RSIM) framework (BPW07) or the Universally Composable (UC) framework (Can01). All details and extended features of SPPC that are not relevant for our results are omitted here; they can be found in (DKMR08).

3.1. *Review of the Syntax*

Intuitively, SPPC describes interacting probabilistic polynomial-time machines. Every such machine is a black-box that receives inputs, performs polynomially-bounded computation and then produces outputs. Inputs are received on *input channels* and outputs are sent through *output channels*. A system of interacting machines is then simply a set of individual machines where any input channel of one machine can be connected directly to an identically named output channel of another machine. The manner in which these machines are wired together is uniquely determined by the channel names, yielding a *valid* set of machines. Two machines are *compatible* if they have the same channel names. We permit output channels that are not connected to any input channels; these correspond to overall (external) outputs of the system. The execution of a system of machines is defined sequentially: the currently active machine makes its transition, the output is transmitted to the connected machine, which now is considered active, etc. One machine is additionally marked as the designated *master process* that is triggered if the execution does not proceed, e.g., because the currently active machine does not produce any output.

More formally, SPPC models every machine (interacting entity) as a *process* $\mathcal{P}$. Such a process corresponds to, e.g., a description of an interactive Turing machine or I/O automaton. A process $\mathcal{P}$ can be parameterized by a security parameter and free variables $\vec{x} = x_1, \dots, x_i$ that represent initial inputs (auxiliary information) to the process. We write $\mathcal{P}_k(\vec{a})$ to denote the process obtained from the process $\mathcal{P}(\vec{x})$ by replacing the variables $\vec{x}$ by values $\vec{a}$ and using k as the security parameter. We write $\mathsf{Pr}[\mathcal{P}_k(\vec{a}) \leadsto \vec{b}]$ to denote the probability that process $\mathcal{P}$ outputs $\vec{b}$ when run on security parameter k and input $\vec{a}$. In the following, we often omit the security parameter for readability.

For a set of valid processes $\mathcal{P}_1, \dots, \mathcal{P}_n$, we write $\mathcal{P}_1 \rceil \dots \rceil \mathcal{P}_n$ to denote the combined *system* of these machines. Instead of interpreting $\mathcal{P}_1 \rceil \dots \rceil \mathcal{P}_n$ as a system of n machines, one can consider this system to be an identically behaving single machine, which internally consists of n submachines. We write $\mathcal{P}_1 \rceil \dots \rceil \mathcal{P}_n \overset{\mathcal{P}_i}{\leadsto} \vec{b}$ to denote the event that process $\mathcal{P}_i$ outputs $\vec{b}$ (as part of the overall output) in the execution of the system.

3.2. *Defining Universal Composability*

Universal composability constitutes the cryptographic notion that a process $\mathcal{P}$ securely realizes another process $\mathcal{Q}$. Intuitively, it formalizes the idea that whatever might happen to a surrounding protocol interacting with process $\mathcal{P}$ can also happen to the protocol interacting with process $\mathcal{Q}$, under the assumption that both $\mathcal{P}$ and $\mathcal{Q}$ offer the same set of channels to which the protocol can connect. More precisely, we assume that the channels of $\mathcal{P}$ and $\mathcal{Q}$ are partitioned into IO channels and network channels. Then $\mathcal{P}$ and $\mathcal{Q}$ are required to have identical IO channels; we speak of *IO-compatible* processes $\mathcal{P}$ and Q in this case. To reflect that $\mathcal{Q}$ is an idealized version of $\mathcal{P}$, one often says that $\mathcal{Q}$ is an ideal functionality, and writes $\mathcal{F}$ instead of $\mathcal{Q}$. More precisely, we will require that for every process $\mathcal{A}$, called the real adversary, there exists a process $\mathcal{I}$, called the ideal adversary, such that $\mathcal{E} \upharpoonright \mathcal{A} \upharpoonright \mathcal{P}$ and $\mathcal{E} \upharpoonright \mathcal{I} \upharpoonright \mathcal{F}$ are computationally indistinguishable for every process $\mathcal{E}$, called the environment. Intuitively, $\mathcal{E}$ should not be able to distinguish if it is interacting with the process and the real adversary, or with the ideal functionality and the ideal adversary. By definition of processes, the adversary $\mathcal{A}$ and the environment $\mathcal{E}$ can interact, similarly for $\mathcal{I}$ and $\mathcal{E}$.

To formally define universal composability, we need to define the set of valid adversaries and valid environments of a given process $\mathcal{Q}$; these are processes that adhere to certain syntactic constraints such as only connecting to certain classes of channels. We write $\mathsf{Adv}(\mathcal{Q})$ to denote the set of processes that constitute *valid adversaries* of $\mathcal{Q}$: $\mathsf{Adv}(\mathcal{Q})$ consists of all non-master processes that are valid for $\mathcal{Q}$, and that do not connect to the IO channels of $\mathcal{Q}$. We write $\mathsf{Adv}_{\mathcal{P}}(\mathcal{Q})$ to denote the set of all processes $\mathcal{P}' \in \mathsf{Adv}(\mathcal{Q})$ such that $\mathcal{P}' \upharpoonright Q$ and $\mathcal{P}$ are compatible. Similarly, we write $\mathsf{Env}(\mathcal{Q})$ to denote the set of processes that constitute *valid environments* of $\mathcal{Q}$: $\mathsf{Env}(\mathcal{Q})$ consists of all master processes that are valid for $\mathcal{Q}$, that do not connect to the network channels of $\mathcal{Q}$, and that contain a distinguished output channel of range $\{0, 1\}$ for outputting the final result of the overall computation. Intuitively, this final result denotes the environment's decision about whether is interacting with the real process or the ideal functionality.

Definition 1 (Indistinguishability). Two systems of processes $\mathcal{P}(\vec{x})$ and $\mathcal{Q}(\vec{x})$ are called indistinguishable, written $\mathcal{P}(\vec{x}) \equiv \mathcal{Q}(\vec{x})$, iff for every polynomial $p(n)$ there exists k_0 such that

$$|\mathsf{Pr}[\mathcal{P}_k(\vec{a}) \rightsquigarrow 1] - \mathsf{Pr}[\mathcal{Q}_k(\vec{a}) \rightsquigarrow 1]| \leq \frac{1}{p(k)}$$

for every $k \geq k_0$ and for every tuple $\vec{a}$ of bitstrings.

We will now define universal composability. Recall that the final output of each of these systems is the value that $\mathcal{E}$ writes on its distinguished output channel.

Definition 2 (Universal Composability). Let $\mathcal{P}$ be a process and $\mathcal{F}$ an ideal functionality.

— We say that $\mathcal{P}$ *securely realizes* $\mathcal{F}$ *computationally*, written $\mathsf{UC}^{\mathsf{comp}}(\mathcal{P}, \mathcal{F})$ iff $\mathcal{P}$ and $\mathcal{F}$ are IO-compatible, and for every probabilistic polynomial-time $\mathcal{A} \in \mathsf{Adv}(\mathcal{P})$ (called the *real adversary*) there exists a probabilistic polynomial-time $\mathcal{I} \in \mathsf{Adv}_{\mathcal{A} \upharpoonright \mathcal{P}}(\mathcal{F})$ (called the *ideal adversary*) such that $\mathcal{E} \upharpoonright \mathcal{A} \upharpoonright \mathcal{P} \equiv \mathcal{E} \upharpoonright \mathcal{I} \upharpoonright \mathcal{F}$ for every probabilistic polynomial-time $\mathcal{E} \in \mathsf{Env}(\mathcal{A} \upharpoonright \mathcal{P})$ (called the environment).

— We say that $\mathcal{P}$ *securely realizes* $\mathcal{F}$ *perfectly*, written $\mathsf{UC}^{\mathsf{perf}}(\mathcal{P}, \mathcal{F})$ if the above statement holds as well if $\mathcal{A}$, $\mathcal{I}$, and $\mathcal{E}$ are not required to run in probabilistic polynomial-time, and if indistinguishability is replaced by equality of probability distributions.

One trivially obtains that $\mathsf{UC}^{\mathsf{perf}}(\mathcal{P}, \mathcal{F})$ implies $\mathsf{UC}^{\mathsf{comp}}(\mathcal{P}, \mathcal{F})$ for arbitrary processes $\mathcal{P}$ and $\mathcal{F}$.

4. Quantifying Transmissible Information in Reactive Systems

In this section, we provide a notion of confidentiality that captures the amount of information that can be transmitted via an arbitrary reactive system (expressed as a process in SPPC). The definition is suited to reasoning about both computationally unbounded and computationally bounded settings. We show that our definition has universal composability features: If an ideal functionality allows the transmission of l bits with at most a certain probability, then any process that securely realizes this functionality allows the transmission of l bits with at most the same probability, up to a negligible error. This compositionality property is key for modular analysis: it allows for analyzing transmissible information of simple ideal functionalities, and for automatically inferring results about the transmissible information for real, cryptographic systems, which are typically much more difficult to analyze.

4.1. *Defining Transmissible Information*

To define the quantity of transmissible information of an arbitrary process $\mathcal{P}$, we assume two users, called *high* and *low*. We provide the high user with a number l of secret bits that he tries to transmit to the low user via $\mathcal{P}$. Intuitively, we say that $\mathcal{P}$ allows for transmitting l bits from high to low, if there exist two processes (representing the high and low users, respectively) that connect to $\mathcal{P}$ and behave as follows: First, the high user chooses l random bits. Then high and low users start interacting with $\mathcal{P}$ (but not with each other). Intuitively, we say that $\mathcal{P}$ allows for transmitting l bits from high to low, if after interacting with $\mathcal{P}$, the low user can successfully output the l bits chosen by the high user (with a certain probability). The advantage of this definition is that it can be used to reason about transmissible information in various computational settings; e.g., we can consider arbitrary unbounded users, or require them to run in probabilistic polynomial time.

To cast this definition in SPPC, let Γ denote a partition of $\mathcal{P}$'s IO channels into high and low; we say that Γ is an *IO partition of* $\mathcal{P}$. We model the high user and the low user as an environment process that is split in two parts $\mathcal{E}_H$ and $\mathcal{E}_L$, which each connect to a given process $\mathcal{P}$ only through the respective channels in Γ. In order to ensure that $\mathcal{E}_H$ and $\mathcal{E}_L$ can only communicate via $\mathcal{P}$, we additionally require that they do not share a common channel. More formally, we write $\mathsf{Env}_\Gamma(\mathcal{P})$ to denote the set of pairs of processes that constitute *valid split-environments* of $\mathcal{P}$: $\mathsf{Env}_\Gamma(\mathcal{P})$ consists of all pairs $(\mathcal{E}_H, \mathcal{E}_L)$ such that $\mathcal{E}_H \mid \mathcal{E}_L \in \mathsf{Env}(\mathcal{P})$, $\mathcal{E}_H$ connects to the high channels in Γ, $\mathcal{E}_L$ connects to the low channels in Γ, and the only unconnected channel of $\mathcal{E}_H$ and $\mathcal{E}_L$ is the distinguished overall output channel, which belongs to $\mathcal{E}_L$.

We are now ready to define the quantity of transmissible information in arbitrary reactive systems. The definition considers unbounded (perfect) as well as probabilistic polynomial-time (computational) adversaries and environments.

Definition 3 (Transmissible Information). Let $\mathcal{P}$ be a process, Γ an IO partition of $\mathcal{P}$, and $p\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ a function that maps into the nonnegative reals.

- We say that $\mathcal{P}$ *perfectly allows the transmission of l bits with probability at most p*, written $\mathcal{P} \preceq_l^{\Gamma,\mathsf{perf}} p$, if for every $\mathcal{A} \in \mathsf{Adv}(\mathcal{P})$ and for every $(\mathcal{E}_H, \mathcal{E}_L) \in \mathsf{Env}_\Gamma(\mathcal{A} \upharpoonleft \mathcal{P})$ we have
$$\mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \xleftarrow{r} \{0,1\}^l, (\mathcal{E}_H(\vec{a}) \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A} \upharpoonleft \mathcal{P}) \overset{\mathcal{E}_L}{\rightsquigarrow} \vec{b}] \leq p(k),$$
where k denotes the implicit security parameter.
- We say that $\mathcal{P}$ *computationally allows the transmission of l bits with probability at most p*, written $\mathcal{P} \preceq_l^{\Gamma,\mathsf{comp}} p$ if, for every probabilistic polynomial-time $\mathcal{A} \in \mathsf{Adv}(\mathcal{P})$ and for every probabilistic polynomial-time $\mathcal{E}_H$ and $\mathcal{E}_L$ such that $(\mathcal{E}_H, \mathcal{E}_L) \in \mathsf{Env}_\Gamma(\mathcal{A} \upharpoonleft \mathcal{P})$, there exists a negligible function ϵ such that
$$\mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \xleftarrow{r} \{0,1\}^l, (\mathcal{E}_H(\vec{a}) \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A} \upharpoonleft \mathcal{P}) \overset{\mathcal{E}_L}{\rightsquigarrow} \vec{b}] \leq p(k) + \epsilon(k),$$
where k denotes the implicit security parameter.

The definition of transmissible information thus captures the following situation: First, l bits are chosen uniformly at random, and are subsequently given to $\mathcal{E}_H$ (as the variable $\vec{a}$). Second, the reactive system $(\mathcal{E}_H(\vec{a}) \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A} \upharpoonleft \mathcal{P})$ is executed, and the probability is considered that $\mathcal{E}_L$ correctly outputs these l bits, i.e., that the event $\vec{a} = \vec{b}$ occurs. This probability should be bounded by $p(k)$ for all valid adversaries and split-environments. The following two simple lemmas serve as a sanity check of this definition: (1) the probability of transmitting l bits cannot be increased by switching from an unbounded to a computationally bounded setting; (2) the probability of transmitting $l+1$ bits can be at most as large as the probability of transmitting l bits.

Lemma 1. Let $\mathcal{P}$ be a process, Γ an IO partition of $\mathcal{P}$, and $p\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ a function. Then $\mathcal{P} \preceq_l^{\Gamma,\mathsf{perf}} p$ implies $\mathcal{P} \preceq_l^{\Gamma,\mathsf{comp}} p$.

Proof. Every valid adversary $\mathcal{A} \in \mathsf{Adv}(\mathcal{P})$ and every valid split-environment $(\mathcal{E}_H, \mathcal{E}_L) \in \mathsf{Env}_\Gamma(\mathcal{A} \upharpoonleft \mathcal{P})$ in the computational case are also a valid adversary and a valid split-environment in the unbounded (perfect) case, respectively. Hence $\mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \xleftarrow{r} \{0,1\}^l, (\mathcal{E}_H(\vec{a}) \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A} \upharpoonleft \mathcal{P}) \overset{\mathcal{E}_L}{\rightsquigarrow} \vec{b}] \leq p(k)$ by assumption. The claim follows using the negligible function $\epsilon(k) := 0$. □

Lemma 2. Let $\mathcal{P}$ be a process, Γ an IO partition of $\mathcal{P}$, and $p\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ a function. Then $\mathcal{P} \preceq_l^{\Gamma} p$ implies $\mathcal{P} \preceq_{l+1}^{\Gamma} p$. This holds both for $\preceq^{\mathsf{perf}}$ and $\preceq^{\mathsf{comp}}$.

Proof. Assume that the process $\mathcal{P}$ does not allow the transmission of $l+1$ bits with probability at most p (either perfectly or computationally). Hence there exists a (probabilistic polynomial-time) adversary $\mathcal{A} \in \mathsf{Adv}(\mathcal{P})$ and (probabilistic polynomial-time) $\mathcal{E}_H$ and $\mathcal{E}_L$ with $(\mathcal{E}_H, \mathcal{E}_L) \in \mathsf{Env}_\Gamma(\mathcal{A} \upharpoonleft \mathcal{P})$ such that we have $q(k) > p(k)$ for some k in the perfect case, and $q(k) > p(k) + \frac{1}{pol(k)}$ for some polynomial pol for infinitely many values of k in the computational case, respectively, where $q(k) := \mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \xleftarrow{r} \{0,1\}^{l+1}, (\mathcal{E}_H(\vec{a}) \upharpoonright \mathcal{E}_L \upharpoonright \mathcal{A} \upharpoonright \mathcal{P}) \overset{\mathcal{E}_L}{\rightsquigarrow} \vec{b}]$.

We show that in this case we obtain a contradiction to the assumption of a probability bound of $\mathcal{P}$ for the transmission of l bits. To this end, define the process $\mathcal{E}_H^{'}$ as follows: given an l-bit input vector $\vec{a}'$, it chooses a bit q at random and behaves as $\mathcal{E}_H(\vec{a}'||q)$ does, i.e., it interacts with $\mathcal{P}$ as $\mathcal{E}_H(\vec{a}'||q)$ would. Similarly, define $\mathcal{E}_L^{'}$ as the process that behaves as $\mathcal{E}_L$ does, except that $\mathcal{E}_L^{'}$ only outputs the first l bits of the output $\vec{b}$ of $\mathcal{E}_L$, i.e., the last bit s is discarded. Since $\mathcal{E}_H$ and $\mathcal{E}_L$ run in probabilistic polynomial-time, so do $\mathcal{E}_H^{'}$ and $\mathcal{E}_L^{'}$. By construction of $\mathcal{E}_H^{'}$ and $\mathcal{E}_L^{'}$, we have $(\mathcal{E}_H^{'}, \mathcal{E}_L^{'}) \in \mathsf{Env}_\Gamma(\mathcal{A} \upharpoonright \mathcal{P})$ and

$$\begin{aligned} &\mathsf{Pr}[\vec{a}' = \vec{b}' : \vec{a}' \xleftarrow{r} \{0,1\}^l, (\mathcal{E}_H^{'}(\vec{a}') \upharpoonright \mathcal{E}_L^{'} \upharpoonright \mathcal{A} \upharpoonright \mathcal{P}) \overset{\mathcal{E}_L^{'}}{\rightsquigarrow} \vec{b}'] \\ = &\mathsf{Pr}[\vec{a}' = \vec{b}' : \vec{a}' \xleftarrow{r} \{0,1\}^l, q \xleftarrow{r} \{0,1\}, (\mathcal{E}_H(\vec{a}'||q) \upharpoonright \mathcal{E}_L \upharpoonright \mathcal{A} \upharpoonright \mathcal{P}) \overset{\mathcal{E}_L}{\rightsquigarrow} (\vec{b}', s)] \\ \geq &\mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \xleftarrow{r} \{0,1\}^{l+1}, (\mathcal{E}_H(\vec{a}) \upharpoonright \mathcal{E}_L \upharpoonright \mathcal{A} \upharpoonright \mathcal{P}) \overset{\mathcal{E}_L}{\rightsquigarrow} \vec{b}] \\ = &q(k) \ . \end{aligned}$$

Since $q(k) > p(k)$ for some k in the perfect case, and $q(k) > p(k) + \frac{1}{pol(k)}$ for some polynomial pol for infinitely many values of k in the computational case, we obtain a contradiction to the transmission bound for l bits. This concludes the proof. □

Lemma 2 asserts that the transmission probability does not increase if one tries to transmit more bits. This might seem weak a statement at first glance, since one might expect this probability to strictly decrease once the number of bits increases. For arbitrary reactive systems, however, this expectation is flawed: consider a system that keeps track of prior invocations; once l bits have been transmitted, it simply behaves like a lossless channel, immediately delivering all inputs from the high user to the low user. For this system, the probability of transmitting l' bits is the same for all $l' \geq l$.

4.2. *Preservation of Transmissible Information under Universal Composability*

We now show that our notion of transmissible information is preserved under universal composability. More precisely, consider two processes $\mathcal{P}_1$ and $\mathcal{P}_2$ such that $\mathcal{P}_1$ securely realizes $\mathcal{P}_2$. Then we show that if $\mathcal{P}_2$ allows the transmission of l bits with probability at most p, then $\mathcal{P}_1$ allows the transmission of l bits with probability at most p as well.

Theorem 1 (Preservation of Information Flow). Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two processes, Γ an IO partition of $\mathcal{P}_1$, and $p \colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ a function. Then we have

— If $\mathcal{P}_2 \preceq_l^{\Gamma,\mathsf{perf}} p$ and $\mathsf{UC}^{\mathsf{perf}}(\mathcal{P}_1, \mathcal{P}_2)$, then $\mathcal{P}_1 \preceq_l^{\Gamma,\mathsf{perf}} p$.
— If $\mathcal{P}_2 \preceq_l^{\Gamma,\mathsf{comp}} p$ and $\mathsf{UC}^{\mathsf{comp}}(\mathcal{P}_1, \mathcal{P}_2)$, then $\mathcal{P}_1 \preceq_l^{\Gamma,\mathsf{comp}} p$.

Proof. Assume for contradiction that $\mathcal{P}_1 \preceq_l^{\Gamma,x} p$ does not hold, where $x \in \{\mathsf{perf}, \mathsf{comp}\}$. In the following, we use brackets to denote additional restrictions on processes in the case $x = \mathsf{comp}$. Hence there exists a (probabilistic polynomial-time) adversary $\mathcal{A}_1 \in \mathsf{Adv}(\mathcal{P}_1)$

and (probabilistic polynomial-time) processes $\mathcal{E}_H$ and $\mathcal{E}_L$ with $(\mathcal{E}_H, \mathcal{E}_L) \in \mathsf{Env}_\Gamma(\mathcal{A} \upharpoonleft \mathcal{P}_1)$ such that we have $q(k) > p(k)$ for some k in the perfect case, and

$$q(k) > p(k) + \frac{1}{pol(k)}$$

for some polynomial pol for infinitely many values of k in the computational case, respectively, where $q(k) := \mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \overset{r}{\leftarrow} \{0,1\}^l, (\mathcal{E}_H(\vec{a}) \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A}_1 \upharpoonleft \mathcal{P}_1) \overset{\mathcal{E}_L}{\rightsquigarrow} \vec{b}]$.

Now $\mathsf{UC}^x(\mathcal{P}_1, \mathcal{P}_2)$ implies that there exists a (probabilistic, polynomial-time) process $\mathcal{A}_2 \in \mathsf{Adv}_{\mathcal{A}_1 \upharpoonleft \mathcal{P}_1}(\mathcal{P}_2)$ such that for every (probabilistic polynomial-time) environment $\mathcal{E} \in \mathsf{Env}(\mathcal{A}_1 \upharpoonleft \mathcal{P}_1)$ we have

$$\mathcal{E} \upharpoonleft \mathcal{A}_1 \upharpoonleft \mathcal{P}_1 \equiv \mathcal{E} \upharpoonleft \mathcal{A}_2 \upharpoonleft \mathcal{P}_2. \tag{1}$$

Consider the perfect case ($x = \mathsf{perf}$). Then (1) means equality of probability distributions, i.e., $\mathcal{E} \upharpoonleft \mathcal{A}_1 \upharpoonleft \mathcal{P}_1 = \mathcal{E} \upharpoonleft \mathcal{A}_2 \upharpoonleft \mathcal{P}_2$. Since $(\mathcal{E}_H, \mathcal{E}_L) \in \mathsf{Env}_\Gamma(\mathcal{A}_1 \upharpoonleft \mathcal{P}_1)$, we have that $\mathcal{E}_H \upharpoonleft \mathcal{E}_L \in \mathsf{Env}(\mathcal{A}_1 \upharpoonleft \mathcal{P}_1)$. We thus in particular obtain

$$\mathcal{E}_H \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A}_1 \upharpoonleft \mathcal{P}_1 = \mathcal{E}_H \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A}_2 \upharpoonleft \mathcal{P}_2.$$

This shows $\mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \overset{r}{\leftarrow} \{0,1\}^l, (\mathcal{E}_H(\vec{a}) \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A}_2 \upharpoonleft \mathcal{P}_2) \overset{\mathcal{E}_L}{\rightsquigarrow} \vec{b}] = q(k)$; since $q(k) > p(k)$ for some k, $\mathcal{P}_2 \preceq_l^{\Gamma,\mathsf{perf}} p$ does not hold.

Consider the computational case ($x = \mathsf{comp}$). Since $\mathcal{P}_2 \preceq_l^{\Gamma,\mathsf{comp}} p$ holds, we know that for every probabilistic polynomial-time $\mathcal{A} \in \mathsf{Adv}(\mathcal{P}_2)$ and for every probabilistic polynomial-time $\mathcal{E}_H^*$ and $\mathcal{E}_L^*$ such that $(\mathcal{E}_H^*, \mathcal{E}_L^*) \in \mathsf{Env}_\Gamma(\mathcal{A} \upharpoonleft \mathcal{P}_2)$, there exists a negligible function μ such that $\mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \overset{r}{\leftarrow} \{0,1\}^l, (\mathcal{E}_H^*(\vec{a}) \upharpoonleft \mathcal{E}_L^* \upharpoonleft \mathcal{A} \upharpoonleft \mathcal{P}_2) \overset{\mathcal{E}_L^*}{\rightsquigarrow} \vec{b}] \leq p(k) + \mu(k)$. We consider the adversary $\mathcal{A}_2 \in \mathsf{Adv}_{\mathcal{A}_1 \upharpoonleft \mathcal{P}_1}(\mathcal{P}_2) \subseteq \mathsf{Adv}(\mathcal{P}_2)$. Since $(\mathcal{E}_H, \mathcal{E}_L) \in \mathsf{Env}_\Gamma(\mathcal{A}_1 \upharpoonleft \mathcal{P}_1)$, we have that $\mathcal{E}_H \upharpoonleft \mathcal{E}_L \in \mathsf{Env}(\mathcal{A}_1 \upharpoonleft \mathcal{P}_1)$. Equation 1 immediately implies $\mathsf{Env}(\mathcal{A}_1 \upharpoonleft \mathcal{P}_1) = \mathsf{Env}(\mathcal{A}_2 \upharpoonleft \mathcal{P}_2)$ since otherwise the indistinguishability cannot even hold syntactically; hence $\mathcal{E}_H \upharpoonleft \mathcal{E}_L \in \mathsf{Env}(\mathcal{A}_2 \upharpoonleft \mathcal{P}_2)$. We thus in particular obtain

$$\mathsf{Pr}[\vec{a} = \vec{b} : \vec{a} \overset{r}{\leftarrow} \{0,1\}^l, (\mathcal{E}_H(\vec{a}) \upharpoonleft \mathcal{E}_L \upharpoonleft \mathcal{A} \upharpoonleft \mathcal{P}_2) \overset{\mathcal{E}_L}{\rightsquigarrow} \vec{b}] \leq p(k) + \epsilon(k)$$

for some negligible function ϵ.

Now consider the overall environment $\mathcal{E}$ in Equation 1 that behaves as $\mathcal{E}_H$ at the high ports of $\mathcal{P}_2$ and as $\mathcal{E}_L$ at the low ports of $\mathcal{P}_2$. If the output of $\mathcal{E}_L$ matches the input of $\mathcal{E}_H$, i.e., if $\vec{a} = \vec{b}$, $\mathcal{E}$ outputs 1, and 0 otherwise. Since $\mathcal{E}_H$ and $\mathcal{E}_L$ run in probabilistic polynomial-time, $\mathcal{E}$ runs in probabilistic polynomial-time.

We obtain $|\mathsf{Pr}[\mathcal{E} \upharpoonleft \mathcal{A}_1 \upharpoonleft \mathcal{P}_1 \rightsquigarrow 1] - \mathsf{Pr}[\mathcal{E} \upharpoonleft \mathcal{A}_2 \upharpoonleft \mathcal{P}_2 \rightsquigarrow 1] \geq \frac{1}{pol(k)} - \epsilon(k)$ for some polynomial pol for infinitely many values of k, which is not negligible. This contradicts Equation 1 and concludes the proof. □

5. Computing Bounds On The Transmissible Information

In this section, we prove a formal connection between channel capacity and transmissible information in the unconditional setting. To this end, we view processes as communication channels in the information-theoretic sense and define their capacity. We prove a variant of the converse of Shannon's coding theorem and use it to give upper bounds on the probability of correct transmission in terms of the channel capacity. This connection enables one to compute an upper bound on the transmissible information for a restricted class of protocols, using existing automated techniques from quantitative information flow. To begin with, we recall some basic information theory.

5.1. *Basic Information Theory*

Let A be a finite set and $d\colon A \to [0,1]$ a probability distribution. For a random variable $X\colon A \to X$ we denote by $Pr\,[X = x]$ the probability that X takes value $x \in \mathcal{X}$, i.e. $Pr\,[X = x] = \sum_{a \in X^{-1}(x)} d(a)$. We sometimes denote the corresponding distribution (of type $\mathcal{X} \to [0,1]$) by $Pr\,[X]$.

The *(Shannon) entropy* (Sha48) of X is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} Pr\,[X = x] \log_2 Pr\,[X = x] \enspace .$$

The entropy is a lower bound on the average number of bits required for representing the results of independent repetitions of the experiment associated with X. Given another random variable $Y\colon A \to \mathcal{Y}$, one denotes by $H(X|Y = y)$ the entropy of X given $Y = y$, that is, with respect to the conditional distribution $Pr\,[X|Y = y]$.

The *conditional entropy* $H(X|Y)$ of X given Y is defined as the expected value of $H(X|Y = y)$ over all $y \in \mathcal{Y}$, namely,

$$H(X|Y) = \sum_{y \in \mathcal{Y}} Pr\,[Y = y]\, H(X|Y = y) \enspace .$$

Entropy and conditional entropy are related by the equation $H(XY) = H(Y) + H(X|Y)$, where XY is the random variable defined as $XY(a) = (X(a), Y(a))$.

The *mutual information* $I(X;Y)$ of X and Y is defined as the reduction of uncertainty about X when one learns Y, namely,

$$I(X;Y) = H(X) - H(X|Y) \enspace .$$

The mutual information is a symmetric function in X and Y; it is 0 if and only if X and Y are independent; and it is upper-bounded by $\min\{H(X), H(Y)\}$.

A *discrete channel* is a conditional probability distribution $Pr\,[Y|X] : \mathcal{X} \times \mathcal{Y} \to [0,1]$. Here, X models the input to the channel and Y the output. The *capacity* of the channel $Pr\,[Y|X]$ is defined as $\max_{Pr\,[X]} I(X;Y)$, where $Pr\,[X]$ ranges over all distributions of X. A channel is *noiseless* if the output is determined by the input, i.e. if $H(Y|X) = 0$. A channel is *lossless* if the input is determined by the output, i.e. if $H(X|Y) = 0$.

5.2. *Channel Processes*

A process $\mathcal{P}$ with IO partition Γ is a *channel process* if $\mathcal{P}$'s communication behavior on the high and low SPPC channels can be captured by a discrete channel in the information-theoretic sense. More precisely, we require that $\mathcal{P}$ accepts only a finite set $\mathcal{X}$ of possible inputs on the high channels and produces only a finite set $\mathcal{Y}$ of possible outputs on the low channels, and that there is a conditional probability distribution $Pr\,[Y|X]$, with

$$Pr\,[Y = y|X = x] = \mathsf{Pr}[\mathcal{P}(x) \rightsquigarrow y].$$

Additionally, $\mathcal{P}$ has to ignore all other communication. We define the *capacity* $Cap_\Gamma(\mathcal{P})$ of $\mathcal{P}$ as the capacity of the information-theoretic channel $Pr\,[Y|X]$.

A process is an *n-use channel process* if it satisfies the requirements of a channel process, but only accepts n inputs (and produces n outputs) before it terminates any communication. We define the capacity of an n-use channel process as the capacity of the corresponding channel process without any restrictions on the number of usages.

5.3. *Information Theoretic Bounds on the Transmissible Information*

In this section, we establish an upper bound on the probability that an n-use channel process $\mathcal{P}$ perfectly allows the transmission of l bits. To this end, we first state three basic lemmas from information theory; their proofs can be found in (CT06). The first lemma is Fano's inequality, which provides a lower bound on the probability of error for arbitrary estimators. For the lemma, recall that random variables X, Y, Z form a *Markov chain*, denoted by $X \to Y \to Z$, if $Pr\,[X = x, Y = y, Z = z] = Pr\,[X = x]\,Pr\,[Y = y|X = x]\,Pr\,[Z = z|Y = y]$. In particular, the Markov property implies that $Pr\,[Z = z|Y = y] = Pr\,[Z = z|Y = y, X = x]$, i.e. the distribution of Z given Y does not depend on X.

Consider now the special case where Z is an estimator for X, i.e. it is intended to recover the value of X from Y. Fano's inequality implies a lower bound on the probability of error for any such estimator in terms of the conditional entropy $H(Y|X)$.

Lemma 3 (Fano's inequality). Let $X \to Y \to \hat{X}$ be a Markov chain and let $q = \mathsf{Pr}[X \neq \hat{X}]$. Then

$$1 + q \log_2 |X| \geq H(X|Y) \ .$$

The second lemma is the data processing inequality, which states that computation cannot increase information.

Lemma 4 (Data processing inequality). Let $X \to Y \to Z$ be a Markov chain. Then

$$I(X;Y) \geq I(X;Z) \ .$$

The third lemma states that the channel capacity is an upper bound on the mutual information between sequence of inputs and outputs that are transmitted through the channel.

Lemma 5. Let Y^m be the result of passing X^m (i.e. m-tuples of values of a random variable X) through a discrete channel $Pr\,[Y|X]$ of capacity C. Then

$$I(X^m; Y^m) \leq mC \ .$$

We are now ready to prove the main result of this section, which is a lower bound on the transmission probability in terms of the channel capacity, the number of channel uses, and the number of transmitted bits.

Theorem 2. Let $\mathcal{P}$ be an n-use channel process and Γ an IO partition of $\mathcal{P}$. Then we have

$$\mathcal{P} \preceq_l^{\Gamma,\mathsf{perf}} \frac{Cap_\Gamma(\mathcal{P})n + 1}{l}.$$

Proof. The transmission process can be modeled as the Markov-chain

$$Z \to X^m \to Y^m \to \hat{Z}$$

with $m \in \{0, \dots, n\}$. Here, the random variable Z captures the experiment $\vec{a} \overset{r}{\leftarrow} \{0,1\}^l$, the random variables X^m and Y^m capture the $m \leq n$ inputs and outputs of the n-use channel process $\mathcal{P}$, respectively, and $Z \to X^m$ and $Y^m \to \hat{Z}$ capture the behavior of $\mathcal{E}_H$ and $\mathcal{E}_L$, respectively. The Markov property is satisfied because (i) the high and low processes of the split environment $(\mathcal{E}_H, \mathcal{E}_L) \in \mathsf{Env}_\Gamma(\mathcal{A} \upharpoonright \mathcal{P})$ do not share a network channel, and because (ii) $\mathcal{P}$ is a $n-$use channel process and hence does not participate in communication with the adversary. That is, the dependency between the inputs to $\mathcal{E}_H$ and the outputs of $\mathcal{E}_L$ is entirely captured by the m inputs and outputs of $\mathcal{P}$.

A calculation similar to (CT06), pp 206, shows

$$\begin{aligned} H(Z) &= H(Z|\hat{Z}) + I(Z;\hat{Z}) & (2)\\ &\leq H(Z|\hat{Z}) + I(X^m;Y^m) & (3)\\ &\leq H(Z|\hat{Z}) + mC & (4)\\ &\leq 1 + q\,l + mC\ , & (5)\end{aligned}$$

where (2) is the definition of the mutual information, (3) follows from Lemma 4, (4) follows from Lemma 5, and (5) follows from Lemma 3 and $H(Z) = l$. Here, q denotes the probability of an decoding error, i.e. $q = \mathsf{Pr}[Z \neq \hat{Z}]$. Consequently, we have

$$p \leq \frac{nC + 1}{l}$$

for the probability $p = 1 - q$ of correctly transmitting l bits using $\mathcal{P}$. □

Theorem 2 is a variant of the so-called weak converse of Shannon's coding theorem. It differs from the one given in (CT06) in that it does not assume deterministic encoders or decoders and in that it abstracts away from the actual code and its rate. The coding theorem shows that, for any fraction $\frac{l}{n} < C$, the probability of correctly transmitting l bits can be arbitrarily close to 1 if n is sufficiently large. This result shows that the bound of Theorem 2 can become tight for large n.

For small values of n and l, however, the information-theoretic bounds given by Theorem 2 are not necessarily tight: for the example of the binary erasure channel and $n = l = 1$, Theorem 2 yields a bound of $p \leq 1.5$. A direct calculation shows that in reality $p = 0.5 + 0.5 \cdot 0.5 = 0.75$. However, for more complex systems and for larger parameters n and l, such a direct computation of p may be too expensive.

$$\begin{tikzcd}
\mathcal{R} & \mathcal{Q} & \mathcal{P} \\
\mathit{Cap}_\Gamma(\mathcal{R}) \leq c \arrow[r, Rightarrow] & \mathit{Cap}_\Gamma(\mathcal{Q}) \leq c \arrow[d, Rightarrow] & \\
 & \mathcal{Q} \preceq_l^{\Gamma,\mathsf{perf}} p \arrow[d, Rightarrow] & \\
 & \mathcal{Q} \preceq_l^{\Gamma,\mathsf{comp}} p \arrow[r, Rightarrow] & \mathcal{P} \preceq_l^{\Gamma,\mathsf{comp}} p
\end{tikzcd}$$

Fig. 3. Overview of the proof of Theorem 3.

5.4. *Tools for Determining Channel Capacity*

In Section 5.3 we have given bounds on the probability of perfectly transmitting information in terms of the channel capacity of n-use processes. The channel capacity can be determined using a number of existing tools from information theory and quantitative information flow. For many important classes of strongly structured channels, e.g., symmetric channels, erasure channels, deletion channels, and buffer channels, closed expressions for the channel capacity are known, see e.g. (CT06; DG06) and references contained therein. For channels given in terms of a conditional probability distribution (e.g. represented as channel matrix) the channel capacity can be approximated using the Blahut-Arimoto algorithm. A recent approach proposes sampling to estimate the channel matrix from program code (CCG10).

A number of recent approaches enable the automatic computation of channel capacity from program code. The key observation here is that the capacity of a channel of range $\mathcal{Y}$ is upper-bounded by $\log_2 |\mathcal{Y}|$. This observation is has been used, e.g. for (Low02) quantifying channel capacity of CSP-processes. More recently, this result has served as a basis for automating the computation of channel capacity by counting the number of reachable states of a program. Existing approaches leverage techniques from automated verification, such as model-checking, #SAT-algorithms, and abstract interpretation (BKR09; KR10; HM10; MS11).

In combination with the techniques developed in this paper, these approaches can be used for the formal analysis of the transmissible information of programs with cryptographic primitives, as we will show next.

6. Formal Illustrative Example

In this section, we show how the techniques presented in this paper can be leveraged for deriving quantitative security guarantees for protocols that contain cryptographic primitives. The section serves as a formal counterpart to the illustrative example presented in Section 2.

The Example Process. We analyze the process $\mathcal{P}$ shown in Figure 4 with IO partition $\Gamma = \{c_H, c_L\}$. The process receives string inputs of length at most m bits on the high channel c_H and produces string outputs on the low channel c_L. Internally, it uses a public-key encryption scheme (Gen, E, D).

Upon first activation, $\mathcal{P}$ generates a key pair $(\mathit{pk}, \mathit{sk})$ using the key generation function

```
     input $msg \in \{0,1\}^{\leq m}$ on channel $c_H$
     begin
1        if first activation
2            $(pk, sk) := Gen(1^k)$ and $i := 0$
3        if $i < n$ do
4            $b \leftarrow L$
5            if $b = 0$
6                send$(loss, c_L)$
7            else
8                send$(E_{pk}(msg), c_L)$
9            $i \leftarrow i + 1$
10       od
     end
```

Fig. 4. The process $\mathcal{P}$

$Gen(1^k)$ and sets an invocation counter i to 0. $\mathcal{P}$ then first checks whether it has already been invoked n times, and aborts in this case. Depending on the value of b, the process either sends the message in encrypted form using the public key pk, or outputs $loss$. We denote the probability of $b = 0$ according to distribution L by p_{loss}. Finally, the invocation counter i is incremented. We claim the following theorem concerning the transmissible information of process $\mathcal{P}$.

Theorem 3. If (Gen, E, D) is a semantically secure public-key encryption scheme, then

$$\mathcal{P} \preceq_l^{\Gamma,\mathsf{comp}} \frac{n(1 - p_{loss}) \log m + 1}{l}.$$

The proof proceeds in three major steps, which we describe in detail below. The overall proof structure is depicted in Figure 3. To increase readability, we did not instantiate the values of p and c in this figure.

Determining the Capacity of a Stripped-down Version $\mathcal{R}$ of $\mathcal{P}$. As the first step of our analysis, we determine the information-theoretic characteristics of $\mathcal{P}$. In Figure 3, this corresponds to the statements right below the line.

For this, we proceed in two sub-steps, which are formalized by Lemmas 6 and 7 below. We define two processes $\mathcal{R}$ and $\mathcal{Q}$ as follows. $\mathcal{R}$ is obtained from $\mathcal{P}$ by replacing the command **send**$(E_{pk}(msg), c_L)$ in Line 8 of Figure 4 by the command **send**$(|msg|, c_L)$, where $|msg| \in \{1, \ldots, m\}$ represents the message length; $\mathcal{Q}$ is obtained from $\mathcal{P}$ by replacing the command **send**$(E_{pk}(msg), c_L)$ in Line 8 by the command **send**$(E_{pk}(0^{|msg|}), c_L)$.

Lemma 6. The process $\mathcal{R}$ is an n-use channel process with

$$Cap_\Gamma(\mathcal{R}) = (1 - p_{loss}) \log m.$$

Proof. To see that $\mathcal{R}$ is an n-use channel observe that, upon the first activation, i is set to 0, and that it is increased for each use of the channel until $i \geq n$. For determining the capacity of $\mathcal{R}$ observe that, with probability $1 - p_{loss}$, $\mathcal{R}$ sends the length of the inputs received on c_H to c_L, and that $\mathcal{R}$ outputs *loss* otherwise. This behavior precisely corresponds to the m-ary erasure channel (see also the picture in Section 2), which is known to have capacity $(1 - p_{loss}) \log m$, see (DG06; CT06). □

The data processing inequality informally states that arbitrary computation does not increase information. In particular, the channel capacity does not increase by replacing every $i \in \{1, \ldots, m\}$ with the encryption $E_{pk}(0^i)$.

Lemma 7. The process $\mathcal{Q}$ is an n-use channel process with

$$Cap_\Gamma(\mathcal{Q}) \leq (1 - p_{loss}) \log m.$$

Proof. Let $X \to Y$ with $\mathcal{X} = \{0,1\}^{\leq m}$ and $Y = \{0, \ldots, m, loss\}$ represent the channel implemented by $\mathcal{R}$, and let $Y \to Z$ denote the channel defined by $loss \mapsto loss$ and $i \mapsto E_{pk}(0^i)$, for all $i \in \{1, \ldots, m\}$. Then $X \to Y \to Z$ represents the channel implemented by $\mathcal{Q}$. The data processing inequality (Lemma 4) shows that $I(X;Y) \geq I(X;Z)$, from which the claim follows. □

Translating Capacity to Reactive Contexts. As the second step of our analysis, we transform the bounds on the capacity of $\mathcal{Q}$ into bounds on the transmissible information of $\mathcal{Q}$ in the computational setting. This corresponds to the vertical proof chain in the middle of Figure 3.

Lemma 8. For the process $\mathcal{Q}$, we obtain

$$\mathcal{Q} \preceq_l^{\Gamma,\mathsf{comp}} \frac{n(1 - p_{loss}) \log m + 1}{l}.$$

Proof. We apply Theorem 2 to the result of Lemma 7 to obtain $\mathcal{Q} \preceq_l^{\Gamma,\mathsf{perf}} (n(1 - p_{loss}) \log m + 1)/l$. The claim then follows from Lemma 1. □

Preservation of Transmissible Information Finally, it remains to be shown that

$$\mathcal{P} \preceq_l^{\Gamma,\mathsf{comp}} \frac{n(1 - p_{loss}) \log m + 1}{l} .$$

Since we already proved that $\mathcal{Q} \preceq_l^{\Gamma,\mathsf{comp}} (n(1 - p_{loss}) \log m + 1)/l$ holds, the preservation theorem (Theorem 1) immediately yields the desired result, provided that we can prove that $\mathsf{UC}^{\mathsf{comp}}(\mathcal{P}, \mathcal{Q})$ holds:

Lemma 9. If (Gen, E, D) is a semantically secure public-key encryption scheme, then $\mathsf{UC}^{\mathsf{comp}}(\mathcal{P}, \mathcal{Q})$.

Proof. Our proof of $\mathsf{UC}^{\mathsf{comp}}(\mathcal{P}, \mathcal{Q})$ follows the standard lines of proofs of universal composability: we first isolate the cryptographic part of the process and use compositionality to obtain the desired property for the overall process.

To this end, we first rewrite $\mathcal{P}$ into a process $\mathcal{P}$' that interacts with a suitable encryption subprocess instead of computing encryptions itself. We consider a real encryption subprocess $\mathcal{E}_{\mathsf{Real}}$ and a simple ideal functionality $\mathcal{E}_{\mathsf{Ideal}}$ of public-key encryption. We show

```
input $msg \in \{0,1\}^{\leq m}$ on $enc_{in}$
begin
    if first activation
        $(pk, sk) := Gen(1^k)$
    send$(E_{pk}(msg), enc_{out})$
end
```

Fig. 5. The real encryption functionality $\mathcal{E}_{\mathsf{Real}}$

```
input $msg \in \{0,1\}^{\leq m}$ on $enc_{in}$
begin
    if first activation
        $(pk, sk) := Gen(1^k)$
    send$(E_{pk}(0^{|msg|}), enc_{out})$
end
```

Fig. 6. The ideal encryption functionality $\mathcal{E}_{\mathsf{Ideal}}$

```
input $msg \in \{0,1\}^{\leq m}$ on channel $c_H$
begin
    if first activation
        $(pk, sk) := Gen(1^k)$ and $i := 0$
    if $i < n$ do
        $b \leftarrow L$
        if $b = 0$
            send$(loss, c_L)$
        else
            send$(msg, enc_{in})$
            receive$(x, enc_{out})$
            send$(x, c_L)$
        $i \leftarrow i + 1$
    od
end
```

Fig. 7. The rewritten process $\mathcal{P}'$

that $\mathcal{P} \equiv \mathcal{P}' \rceil \mathcal{E}_{\mathsf{Real}}$ and $\mathcal{Q} \equiv \mathcal{P}' \rceil \mathcal{E}_{\mathsf{Ideal}}$, i.e., $\mathcal{P}'$ behaves as $\mathcal{P}$ or $\mathcal{Q}$, respectively, depending on whether the real or the ideal encryption subprocess is used. Finally, we exploit the semantic security of the encryption scheme to show that $\mathsf{UC}^{\mathsf{comp}}(\mathcal{E}_{\mathsf{Real}}, \mathcal{E}_{\mathsf{Ideal}})$. The composition theorem then implies $\mathsf{UC}^{\mathsf{comp}}(\mathcal{P}, \mathcal{Q})$.

Let the simple ideal functionality $\mathcal{E}_{\mathsf{Ideal}}$ for public-key encryption be defined according to Figure 6. $\mathcal{E}_{\mathsf{Ideal}}$ expects inputs on a channel enc_{in} and produces outputs on channel enc_{out}. In its first activation, $\mathcal{E}_{\mathsf{Ideal}}$ generates a key pair (pk, sk) using $Gen(1^k)$. Upon input msg, $\mathcal{E}_{\mathsf{Ideal}}$ outputs $E_{pk}(0^{|msg|})$. $\mathcal{E}_{\mathsf{Real}}$ is defined in Figure 5. It is identical to $\mathcal{E}_{\mathsf{Ideal}}$ except that it outputs $E_{pk}(msg)$ instead of $E_{pk}(0^{|msg|})$. The rewritten process $\mathcal{P}'$ differs from $\mathcal{P}$ only in Line 8 of Figure 4: instead of computing $E_{pk}(msg)$ itself, it sends msg on channel enc_{in} and expects on answer x on channel enc_{out}, which it then outputs on channel c_L. For completeness, the definition of $\mathcal{P}'$ is given in Figure 7.

Showing the Universal Composability Relation By construction, we obtain that $\mathcal{P} \equiv \mathcal{P}' \rceil \mathcal{E}_{\mathsf{Real}}$ and $\mathcal{Q} \equiv \mathcal{P}' \rceil \mathcal{E}_{\mathsf{Ideal}}$, since $\mathcal{P}'$ is just a syntactic rewriting of both processes when calling subprocesses $\mathcal{E}_{\mathsf{Real}}$ and $\mathcal{E}_{\mathsf{Ideal}}$, respectively. Moreover, one easily shows the following lemma that asserts that $\mathcal{E}_{\mathsf{Real}}$ securely realizes $\mathcal{E}_{\mathsf{Ideal}}$ computationally if the encryption scheme is semantically secure.

Lemma 10. If (Gen, E, D) is a semantically secure public-key encryption scheme, then $\mathsf{UC}^{\mathsf{comp}}(\mathcal{E}_{\mathsf{Real}}, \mathcal{E}_{\mathsf{Ideal}})$.

The lemma follows immediately from the semantic security property, see (PW01). In (PW01), universal composability is even shown for public-key encryption functionalities that additionally offer decryption requests; these are answered by the ideal functionality using table look-up.

Finally, since $\mathcal{P} \equiv \mathcal{P}' \upharpoonright \mathcal{E}_{\mathsf{Real}}$, $\mathcal{Q} \equiv \mathcal{P}' \upharpoonright \mathcal{E}_{\mathsf{Ideal}}$, and $\mathsf{UC}^{\mathsf{comp}}(\mathcal{E}_{\mathsf{Real}}, \mathcal{E}_{\mathsf{Ideal}})$ hold, the composition theorem of SPPC yields $\mathsf{UC}^{\mathsf{comp}}(\mathcal{P}, \mathcal{Q})$. □

A concluding remark. We conclude this section by substantiating the claim from the introduction that cryptographic messages might even carry *more* information than expected from the perspective of information theory. Consider a modification of process $\mathcal{P}$ from Figure 4 with an additional Line 9.5: **if** $i = n$ **send**(sk, c_L); i.e., the decryption key sk is additionally leaked in the n-th execution. Theorem 3 has shown that without divulging the secret key sk in the n-th execution, the process $\mathcal{P}$ allows for computationally transmitting l bits with probability at most $p := \frac{n(1-p_{loss})\log m+1}{l}$. For the sake of illustration, let $p_{loss} = 0$ and $m = 1$. Thus $p = 1/l$, i.e., at most one bit can be transmitted with probability 1. If sk is divulged in the n-th execution, however, even transmitting $l := n$ bits is possible with probability 1: $\mathcal{E}_H$ enters its input $\vec{a} = (a_1, \ldots, a_n)$ of length n bitwise into $\mathcal{P}$; $\mathcal{E}_L$ simply stores all encryptions $c_i := E_{pk}(a_i)$ it receives from $\mathcal{P}$, decrypts them to $b_i := D_{sk}(c_i)$ after it received sk, and outputs $\vec{b} = (b_1, \ldots, b_n)$. The correctness property of the encryption scheme ensures $\vec{a} = \vec{b}$ with probability 1; i.e., $\mathcal{P}$ allows n bits to be correctly transmitted. Thus sending a (short) k-bit string sk triggers the additional transmission of $n-1$ bits, where n is an arbitrary polynomial in k.

7. Related Work

The first approach to use information-theoretic capacity for the analysis of covert channels is (Mil87). A connection between Shannon's coding theorem and nondeterministic notions of information flow has been made in (WJ90). A generalization of Shannon's theorem and its converse to finite-state channels can be found in (Gal68) and (Ari73), respectively. The notion of channel capacity has been generalized to stateful channels with feedback (Gra92); however, a coding theorem for this model of communication is only conjectured. An approximation of channel capacity in terms of the number of possible behaviors of a CSP process is given in (Low02). In contrast to these works, our definition of transmissible information can be used for expressing information flow in the context of stateful reactive systems with cryptographic primitives; moreover, it enjoys an explicit connection to channel capacity in the stateless case.

A study of conditions for the safe use of one-way functions in a programming language is presented in (Vol00). The considered security property captures the secrecy of a specific secret rather than non-interference. The first investigation of non-interference in a computational setting can be found in (Lau01). The presented definition is non-reactive and specific to encryption as the only cryptographic primitive. Definitions of information flow that allow for complexity-theoretic reasoning about extended cases such as reactive scenarios and additional cryptographic primitives can be found in (BP02; BP03; FR08; FGR09). However, these definitions do not capture quantitative aspects of information flow. The only prior work to combine quantitative information flow with computational assumptions is (Bac05). This work proposes a measure of the transmitted information based on the computational distance between two low-user observations under varying high-user behaviors. This distance, however, was not shown to correspond to the amount of transmitted bits, and no relationship to information theory was presented. Finally, a recent study considers public-key cryptosystems in which information about the key is leaked (KS10). The main result shows that complexity-theoretic security is not compromised if only a logarithmic (in the security parameter) number of bits are leaked. In contrast, in this paper we consider the deliberate communication between two processes rather than the unintended leakage of the key.

In Section 5.4, we have already discussed tools for determining the channel capacity of programs. For completeness, we briefly present a number of closely related approaches. A type system for statically deriving quantitative bounds on the information that a straight-line program leaks is presented in (CHM05; CHM07). The type system is complemented by a formula that characterizes the leakage of loops in terms of the loop's output and the number of iterations (Mal07). These approaches capture leakage with respect to a fixed probability distribution on the inputs. It would be interesting to see whether they can be extended to derive bounds on the channel capacity, which requires the quantification over all input probability distributions. Finally, the dynamic analysis from (ME08) enables one to derive bounds on the information that is leaked in a single run of the program. The results are not directly applicable to the problem at hand, as this would require the derivation of bounds that hold for the entirety of program runs.

Finally, we mention that the techniques developed in this paper can be used to check a protocol's adherence to a quantitative declassification policy. In particular, bounds on the number of disclosed bits can be seen as a special case of restricting "what" information is released (SS09). For a formal connection of qualitative (typically relational) specifications of declassification policies and quantitative policies, see (BKR09; VC11).

8. Conclusion and Future Work

We have presented a novel definition of quantitative information flow, called transmissible information, that is suitable for reasoning about information-theoretically secure or non-cryptographic systems, as well as about cryptographic systems with their polynomially bounded adversaries, error probabilities, etc. We have shown that transmissible information is preserved under universal composability, which enables its seamless integration with state-of-the art compositional security proofs of cryptographic protocols. We have furthermore proven a connection between transmissible information in the unconditional setting and channel capacity, based on the weak converse of Shannon's coding theorem. This connection enables us to compute an upper bound on the transmissible information for a restricted class of protocols, using existing techniques for quantitative information-flow analysis. Finally, we applied our results to derive quantitative security guarantees for a simple public-key encryption-based example.

On the information-theoretic side, we consider it interesting future work to investigate whether existing bounds for the transmission over finite-state channels (Kie74) can be turned into bounds of transmissible information of more general cryptographic systems. On the cryptographic side, a particularly interesting task is to mechanize security analyses (such as the one we performed in Section 6), e.g., using transformational type systems. We hope that advances on both sides will eventually lead to tools for the mechanized, quantitative information flow analysis of more complex cryptographic systems.

References

Suguru Arimoto. On the Converse to the Coding Theorem for Discrete Memoryless Channels. *IEEE Trans. Information Theory*, 19:357–359, 1973.

Michael Backes. Quantifying probabilistic information flow in computational reactive systems. In *Proc. 10th European Symposium on Research in Computer Security (ESORICS)*, LNCS 3679, pages 336–354. Springer, 2005.

Michael Backes, Boris Köpf, and Andrey Rybalchenko. Automatic Discovery and Quantification of Information Leaks. In *Proc. 30th IEEE Symposium on Security and Privacy (SSP)*, pages 141–153. IEEE Computer Society, 2009.

Michael Backes and Birgit Pfitzmann. Computational probabilistic non-interference. In *Proc. 7th European Symposium on Research in Computer Security (ESORICS)*, volume 2502 of *LNCS*, pages 1–23. Springer, 2002.

Michael Backes and Birgit Pfitzmann. Intransitive non-interference for cryptographic purposes. In *Proc. 24th IEEE Symposium on Security & Privacy*, pages 140–152, 2003.

Michael Backes, Birgit Pfitzmann, and Michael Waidner. The reactive simulatability (RSIM) framework for asynchronous systems. *Information and Computation*, 205(12):1685–1720, 2007.

Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 136–145, 2001. Extended version in Cryptology ePrint Archive, Report 2000/67, `http://eprint.iacr.org/`.

Konstantinos Chatzikokolakis, Tom Chothia, and Apratim Guha. Statistical measurement of information leakage. In *16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, LNCS 6015, pages 390–404. Springer, 2010.

David Clark, Sebastian Hunt, and Pasquale Malacaria. Quantitative Information Flow, Relations and Polymorphic Types. *J. Log. Comput.*, 18(2):181–199, 2005.

David Clark, Sebastian Hunt, and Pasquale Malacaria. A static analysis for quantifying information flow in a simple imperative language. *Journal of Computer Security*, 15(3):321–371, 2007.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley, second edition, 2006.

Suhas N. Diggavi and Matthias Grossglauser. On Information Transmission over a Finite Buffer Channel. *IEEE Transactions on Information Theory*, 52(3):1226–1237, 2006.

Anupam Datta, Ralf Küsters, John C. Mitchell, and Ajith Ramanathan. On the relationships between notions of simulation-based security. *J. Cryptol.*, 21(4):492–546, 2008.

Cédric Fournet, Gurvan Le Guernic, and Tamara Rezk. A security-preserving compiler for distributed programs: from information-flow policies to cryptographic mechanisms. In *Proc. ACM Conference on Computer and Communications Security (CCS)*, pages 432–441. ACM Press, 2009.

Cédric Fournet and Tamara Rezk. Cryptographically sound implementations for typed information-flow security. In *Proc. 35th ACM Symposium on Principles of Programming Languages (POPL)*, pages 323–335. ACM Press, 2008.

Robert G. Gallager. *Information Theory and Reliable Communication.* John Wiley and Sons, Inc., 1968.

Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

James W. Gray. Toward a Mathematical Foundation for Information Flow Security. *Journal of Computer Security*, 1(3-4):255–294, 1992.

Jonathan Heusser and Pasquale Malacaria. Quantifying information leaks in software. In *26th Annual Computer Security Applications Conference, (ACSAC)*, pages 261–269. ACM, 2010.

John C. Kieffer. A Lower Bound on the Probability of Decoding Error for the Finite-State Channel. *IEEE Trans. Information Theory*, 20:549–551, 1974.

Boris Köpf and Andrey Rybalchenko. Approximation and Randomization for Quantitative Information-Flow Analysis. In *Proc. 23rd IEEE Computer Security Foundations Symposium (CSF)*, pages 3–14. IEEE, 2010.

Boris Köpf and Geoffrey Smith. Vulnerability Bounds and Leakage Resilience of Blinded Cryptography under Timing Attacks. In *Proc. 23rd IEEE Computer Security Foundations Symposium (CSF)*, pages 44–56. IEEE, 2010.

Peeter Laud. Semantics and program analysis of computationally secure information flow. In *Proc. 10th European Symposium on Programming (ESOP)*, pages 77–91, 2001.

Gavin Lowe. Quantifying Information Flow. In *Proc. IEEE Computer Security Foundations Workshop (CSFW)*, pages 18–31. IEEE Computer Society, 2002.

Pasquale Malacaria. Assessing security threats of looping constructs. In *Proc. ACM Symposium on Principles of Programming Languages (POPL '07)*, pages 225–235. ACM Press, 2007.

Stephen McCamant and Michael D. Ernst. Quantitative information flow as network flow capacity. In *Proc. Conf. on Programming Language Design and Implementation (PLDI)*, pages 193–205, 2008.

Johnathan K. Millen. Covert Channel Capacity. In *Proc. IEEE Symposium on Security and Privacy (S&P)*, pages 60–66. IEEE Computer Society, 1987.

Ziyuan Meng and Geoffrey Smith. Calculating bounds on information leakage using two-bit patterns. In *6th ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, 2011.

Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proc. 22nd IEEE Symposium on Security & Privacy*, pages 184–200, 2001. Extended version of the model (with Michael Backes) IACR Cryptology ePrint Archive 2004/082, `http://eprint.iacr.org/`.

Claude E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

Andrei Sabelfeld and David Sands. Declassification: Dimensions and principles. *Journal of Computer Security*, 17(5):517–548, 2009.

Jeffrey A. Vaughan and Stephen Chong. Inference of expressive declassification policies. In *32nd IEEE Symposium on Security and Privacy (S&P)*, pages 180–195. IEEE, 2011.

Dennis Volpano. Secure introduction of one-way functions. In *Proc. 13th IEEE Computer Security Foundations Workshop (CSFW)*, pages 246–254, 2000.

J. Todd Wittbold and Dale M. Johnson. Information Flow in Nondeterministic Systems. In *Proc. IEEE Symposium on Security and Privacy (S&P '90)*, pages 144–161. IEEE Computer Society, 1990.