Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Electronic Notes in Theoretical Computer Science 253 (2009) 143–165
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Modeling and Reasoning about an Attacker with Cryptanalytical Capabilities

Bruno Montalto[1,2]

Department of Computer Science
Information Security Group
ETH Zürich, Switzerland
and
SQIG - Instituto de Telecomunicações
Department of Mathematics, IST
TU Lisbon, Portugal

Carlos Caleiro [1,3]

SQIG - Instituto de Telecomunicações
Department of Mathematics, IST
TU Lisbon, Portugal

Abstract

We propose a probabilistic framework for the analysis of security protocols. The proposed framework allows one to model and reason about attackers that extend the usual Dolev-Yao adversary with explicit probabilistic statements representing properties of cryptographic primitives and the attacker's (partial) information about secret messages. The expressive power of these probabilistic statements is illustrated, namely by representing a standard security notion like indistinguishability under chosen plaintext attacks. We present an entropy-based approach to estimate the probability of a successful attack on a protocol given the prescribed knowledge of the attacker. We prove that, for an attacker whose knowledge increases with the security parameter, computing this quantity is NP-hard in the security parameter. However, we are still able to analyze a few meaningful and illustrative examples. Finally, we obtain a result which may be used to prove that a certain amount of probabilistic knowledge (about the properties of the cryptography being used) is not enough for allowing an attacker to correctly uncover a secret with non-negligible probability.

Keywords: Security protocol, attacker, probabilistic statement, cryptographic property, Shannon entropy.

[1] This work was partially supported by FCT and EU FEDER, namely via the KLog PTDC/MAT/68723/2006 project of SQIG-IT. The first author acknowledges the successive support of a research grant awarded by Fundação Calouste Gulbenkian on account of the 2006 edition of the Prize Professor Jaime Campos Ferreira, followed by a BI grant of the KLog project, and finally the PhD grant SFRH/BD/44204/2008 by FCT.

[2] Email: brunoco@inf.ethz.ch

[3] Email: ccal@math.ist.utl.pt

1571-0661© 2009 Elsevier B.V. Open access under CC BY-NC-ND license.
doi:10.1016/j.entcs.2009.10.010

1 Introduction

The analysis of security protocols has been the subject of much research in the last few decades. Two fundamentally distinct approaches have been used to this end.

In the symbolic approach, introduced in the early 1980's by Dolev and Yao [14], messages are represented as terms and technical details of cryptographic primitives are ignored. These models adopt the perfect cryptography assumption. For example, encryption is viewed as a black-box operation, so that, from an encrypted message, the attacker may not obtain any partial knowledge about the original message. Such symbolic methods have been widely accepted among the scientific community, and several tools for automated analysis of security protocols (mostly attack-search engines) have been developed based on this approach [5,10,13]. In fact, one of the main reasons for the popularity of this approach is that such strong abstractions allow automated proofs of the security of protocols. Their main weakness, however, is that it is hard to prove that these abstractions are sound, since in practice cryptographic primitives have properties which the attacker may explore and attack: for example, the redundancy of certain messages may be explored by the attacker to guess a weak password [8].

In stark contrast with the formal approach is the computational one. Computational methods use a more complex framework, in which messages are treated as actual bitstrings and cryptographic primitives as functions acting on those bitstrings. In order to provide a more complete and realistic representation of a "real" attacker, the computational approach deals with concepts like complexity and probability [20,25]. Security proofs in the computational approach are generally stronger than in the formal approach, since they allow the attacker to explore vulnerabilities of the cryptographic primitives to find a suitable attack. However, the greater complexity of these methods makes it difficult to obtain such proofs in an automated way, and it is hard to analyze even simple protocols in this setting.

In recent years, there has been a considerable effort in bridging the gap between these two approaches [1,6,7,16,27]. Our work aims precisely at this task. We present a formal model featuring an attacker who is allowed to take advantage of partial probabilistic information about the messages exchanged in the execution of a protocol by exploring certain cryptographic properties. This information is represented in the form of probabilistic statements about secret messages seen as random variables. We assume any suitable underlying communication model representing a Dolev-Yao adversary, such as in [3]. For the sake of space, we do not dwell here on the details of any particular symbolic model, although we have reported on it in [23]. We also present a way to estimate the probability of success of an attack based on the probabilistic statements that are known by the attacker. Estimating this quantity is useful not only because it provides a quantitative measure of the security of a protocol, but also because it allows us to evaluate the security impact of certain cryptographic weaknesses. In fact, our model is based not on the specific details of the cryptographic primitives but rather on probabilistic statements about their properties and the partial information about secret messages that the attacker may uncover. Thus, by describing general cryptographic properties, we may use

our model to assess their impact on the security of a protocol.

The presentation is organized as follows. In Section 2 we describe our setting. Namely, we define how we represent messages in our framework and describe the capabilities of the attacker. We define the syntax of the probabilistic statements considered in our model and present some examples which illustrate what these statements may represent, including cryptographic properties. In particular, we prove a theorem which shows how IND-CPA (indistinguishability under chosen plaintext attacks) security, a standard property for asymmetric encryption schemes, may be translated to our framework. Section 3 concerns the estimation of probabilities based on the set of probabilistic statements available to the attacker. We interpret messages as random variables whose range (the set of values which they may assume) is a set of bitstrings. To this end we use Shannon's notion of entropy. We present some examples which illustrate these ideas and methods, and obtain a theorem which may be useful in proving that the partial knowledge (about the properties of the cryptography being used) of an attacker is not enough for correctly guessing a secret with non-negligible probability. In Section 4 we assess our work and present some questions for further research. We include an Appendix with detailed proofs and examples.

2 The framework

We consider a public network where principals exchange private messages. To prevent an attacker from breaking the security of the network, the principals use security protocols. These protocols specify how the principals construct the messages that they publish using cryptographic functions. In our framework, each message corresponds to a bitstring. We represent the set of bitstrings by $\mathbf{B} = \{0,1\}^*$. We use elements of $\mathbf{B}$ to encode numbers and finite sequences of bitstrings [4]. To represent the functions used by the principals to manipulate messages and execute the protocols, we use a finite set $\mathcal{F}$ of deterministic algorithms and another finite set $\mathcal{R}$ of probabilistic algorithms. Note that it is possible to simulate the execution of a general probabilistic algorithm by means of a deterministic algorithm and a probabilistic algorithm which represents the randomness involved in the calculations and depends only on the security parameter. For this reason, we will assume (without loss of generality) that all algorithms in $\mathcal{R}$ receive only the security parameter as input, and will dub them random generation algorithms.

Since we want to model an attacker A with additional cryptanalytical capabilities, we consider additional finite sets of deterministic algorithms, $\mathcal{F}^A \supseteq \mathcal{F}$, and random generation algorithms, $\mathcal{R}^A \supseteq \mathcal{R}$, which A may use to obtain and represent knowledge about the secret data involved in the protocols or to compose messages of his own. For the sake of reasoning about complexity issues and ultimately modeling and reasoning about computationally feasible attacks exploring cryptanalysis, we consider that each algorithm depends on a security parameter. We will be interested

[4] Recall that there are bijections between $\mathbf{B}$ and $\mathbb{Z}$ and between $\mathbf{B}$ and $\mathbf{B}^*$ which may be efficiently computed and inverted.

in modeling an attacker with limited computational power, *i.e.*, an attacker who can only perform a number of operations limited by a function (usually a polynomial) of the security parameter.

The next definition introduces the set of valid expressions for given sets $\mathcal{F}, \mathcal{R}$ of algorithms. We will denote this set by $\mathbf{Exp}(\mathcal{F}, \mathcal{R})$. These expressions represent the construction of a message by applying deterministic algorithms to bitstrings (either randomly generated or chosen by the principals).

Definition 2.1 Let $\mathcal{F}$ be a finite set of deterministic algorithms and $\mathcal{R}$ a finite set of random generation algorithms. The set $\mathbf{Exp}(\mathcal{F}, \mathcal{R})$ of *expressions* generated by the sets $\mathcal{F}, \mathcal{R}$ is defined inductively as follows:

- for each $j \in \mathbb{N}$ and $R \in \mathcal{R}$, $R^j \in \mathbf{Exp}(\mathcal{F}, \mathcal{R})$;
- $\mathbf{B} \subseteq \mathbf{Exp}(\mathcal{F}, \mathcal{R})$;
- if $E^1, \ldots, E^n \in \mathbf{Exp}(\mathcal{F}, \mathcal{R})$ and $F \in \mathcal{F}$, $F(E^1, \ldots, E^n) \in \mathbf{Exp}(\mathcal{F}, \mathcal{R})$.

When the sets $\mathcal{F}, \mathcal{R}$ of public algorithms and $\mathcal{F}^A, \mathcal{R}^A$ of algorithms available to an attacker A are clear from the context, we abbreviate $\mathbf{Exp} = \mathbf{Exp}(\mathcal{F}, \mathcal{R})$, $\mathbf{Exp}^A = \mathbf{Exp}(\mathcal{F}^A, \mathcal{R}^A)$. Intuitively, each expression in $\mathbf{Exp}$ (resp. $\mathbf{Exp}^A$) represents the construction of a message using public algorithms (resp. the algorithms available to the attacker A). Each random generation algorithm $R \in \mathcal{R}^A$ may be executed several times. Since each of these executions may output a different bitstring, we need to distinguish between them. Thus, for each $j \in \mathbb{N}$, the expression R^j represents a different execution of the random generation algorithm R.

We now introduce some useful notation. Consider an attacker A with limited computational power (*i.e.*, for each security parameter η, A may only perform a finite number of operations $f(\eta)$). It is easy to see that, for each η, A can only interpret and obtain information from expressions with at most a finite size. Thus, when modeling A, the set of expressions which need to be considered is finite; for instance, one may only consider expressions with up to $f(\eta)$ subexpressions [5]. Let us denote that set by $\mathbf{Exp}^A_\eta \subseteq \mathbf{Exp}^A$. For similar reasons, we may assume that the set of bitstrings which an expression in $\mathbf{Exp}^A_\eta$ may possibly represent is also a finite set $\mathbf{B}^A_\eta \subseteq \mathbf{B}$. For each η, we define the set

$$\Omega^A_\eta = \{f \mid f \colon \mathbf{Exp}^A_\eta \setminus \mathbf{B} \to \mathbf{B}^A_\eta\}.$$

We use the superscript A to highlight the fact that this set depends on the algorithms and bitstrings available to A. An element of Ω^A_η associates each expression in $\mathbf{Exp}^A_\eta \setminus \mathbf{B}$ to a bitstring in $\mathbf{B}^A_\eta$ - thus, a probability distribution on Ω^A_η completely determines the probability distribution of all random variables represented by expressions, as well as dependencies between them. For example, for $E, E' \in \mathbf{Exp}^A_\eta$,

[5] Note also that we need to assume a finite set of expressions of the form R^i for each algorithm $R \in \mathcal{R}^A$; we namely require that $i < f(\eta)$.

$b, b' \in \mathbf{B}_\eta^A$,

$$P[E = b \mid E' = b'] = \frac{P[\{f \in \mathbf{Exp}_\eta^A \mid f(E) = b, f(E') = b'\}]}{P[\{f \in \mathbf{Exp}_\eta^A \mid f(E') = b'\}]}.$$

We will denote by $\mathcal{D}_\eta^A$ the probability distribution induced on Ω_η^A by the algorithms in $\mathcal{F}^A$ and $\mathcal{R}^A$. If $E \in \mathbf{Exp}_\eta^A$, we write E_η for the random variable representing the bitstring obtained by constructing a message in the way prescribed by E and using security parameter η in the computations. If $\mathcal{D}$ is any probability distribution on Ω_η^A and $\{E_1, \ldots, E_n\} \subseteq \mathbf{Exp}_\eta^A$, we will write $E_n, \ldots, E_n \leftarrow \mathcal{D}$ to denote that the n-tuple of random variables $(E_1, \ldots, E_n)$ has the probability distribution determined by $\mathcal{D}$. In particular, if $\mathcal{D} = \mathcal{D}_\eta^A$ and $E^1, \ldots, E^n \leftarrow \mathcal{D}$, then $(E^1, \ldots, E^n) \sim (E_\eta^1, \ldots, E_\eta^n)$ (*i.e.*, both vectors have the same joint probability distribution). We will omit the security parameter η when it is clear from the context or is not relevant to the discussion.

We may now introduce our notion of probabilistic statement.

Definition 2.2 A *probabilistic statement* is a statement of the form

$$P[E_\eta^1 = b_1, \ldots, E_\eta^n = b_n \mid E_\eta^{*,1} = b_1^*, \ldots, E_\eta^{*,n^*} = b_{n^*}^*] = \rho,$$

where $\eta \in \mathbb{N}$ is a security parameter, $E^1, \ldots, E^n, E^{*,1}, \ldots, E^{*,n^*} \in \mathbf{Exp}_\eta^A$, $b_1, \ldots, b_n$, $b_1^*, \ldots, b_{n^*}^* \in \mathbf{B}_\eta^A$, and $\rho \in [0, 1]$.

Such probabilistic statements model cryptographic properties written as probabilistic relations between expressions. This means that, by knowing the bitstrings corresponding to certain expressions, the attacker is able to learn "something" about the probability distribution of the bitstrings corresponding to other expressions. If $\eta \in \mathbb{N}$, $E^1, \ldots, E^n$, $E^{*,1}, \ldots, E^{*,n^*} \in \mathbf{Exp}_\eta^A$, $b_1, \ldots, b_n$, $b_1^*, \ldots, b_{n^*}^* \in \mathbf{B}_\eta^A$ and $\rho \in [0, 1]$, we will say that

$$P[E^1 = b_1, \ldots, E^n = b_n \mid E^{*,1} = b_1^*, \ldots, E^{*,n^*} = b_{n^*}^*] = \rho$$

is verified by a probability distribution $\mathcal{D}$ on Ω_η^A if the equality holds when $E^1, \ldots, E^n, E^{*,1}, \ldots, E^{*,n^*} \leftarrow \mathcal{D}$.

We now present some simple properties of cryptographic primitives which may be represented by probabilistic statements.

Example 2.3 One of the simplest and most important properties one may think of is that decryption with a private key is the "inverse" operation of encryption with the corresponding public key. To represent this, consider a probabilistic asymmetric encryption scheme $(R_k, pub, priv, R_e, Enc, Dec)$, where:

- $R_k \in \mathcal{R}$ is the key generation algorithm;
- pub, $priv \in \mathcal{F}$ are deterministic algorithms such that $pub(R_k^i)$ (resp. $priv(R_k^i)$) returns the public (resp. private) key corresponding to R_k^i;

- $R_e \in \mathcal{R}$ is a random generation algorithm representing the randomness involved in the (probabilistic) encryption;
- Enc, $Dec \in \mathcal{F}$ are the encryption and decryption algorithms, respectively.

The encryption of a message represented by the expression E using a public key $pub(R_k^i)$ and random data R_e^j is represented by the expression $Enc(E, pub(R_k^i), R_e^j)$, and the decryption of E using the private key $priv(R_k^i)$ is represented by $Dec(E, priv(R_k^i))$, where R_k^i, R_e^j respectively represent a randomly generated key and random data used in the encryption. Thus, if the encryption/decryption algorithms are correct, the following probabilistic statements are valid:

$$P[Dec(Enc(E, pub(R_k^i), R_e^j), priv(R_k^i)) = b \mid E = b] = 1,$$

$$P[E = b \mid Dec(Enc(E, pub(R_k^i), R_e^j), priv(R_k^i)) = b] = 1.$$

Example 2.4 Let $(R_k, pub, priv, R_e, Enc, Dec)$ be the Elgamal encryption scheme. In this encryption scheme, messages are represented by elements of a group. Thus, we may consider an additional deterministic algorithm $mult \in \mathcal{F}^A$ such that $mult(E^1, E^2)$ represents the multiplication of the two messages corresponding to E^1 and E^2. The well-known malleability property of Elgamal can be represented by the following pair of properties.

$$P[mult(E^1, Enc(E^2, pub(R_k^i), R_e^j)) = b \mid Enc(mult(E^1, E^2), pub(R_k^i), R_e^j) = b] = 1,$$

$$P[Enc(mult(E^1, E^2), pub(R_k^i), R_e^j) = b \mid mult(E^1, Enc(E^2, pub(R_k^i), R_e^j)) = b] = 1.$$

Example 2.5 Many encryption functions reveal some information about the length of the underlying plaintext. Suppose that an encryption scheme is such that

> "If the length of a ciphertext is l, then there is a 0.5 probability that the length of the underlying plaintext is also l."

This may be represented by the cryptographic property

$$P[length(E) = l \mid length(Enc(E, pub(R_k^i), R_e^j)) = l] = 0.5,$$

where $length \in \mathcal{F}^A$ is an algorithm which returns a bitstring representing the length of the bitstring corresponding to a given expression and l is a bitstring representing an integer.

Other examples of probabilistic statements may be found in Example 3.7 and Example 3.9; their discussions are given in the Appendix.

We now show how to represent the notion of IND-CPA (indistinguishability under chosen plaintext attacks) security for asymmetric encryption schemes using the notion of probabilistic statements as defined above. Recall that a function $f\colon \mathbb{N} \to \mathbb{R}$ is *negligible* if, for every $c > 0$, there exists $n \in \mathbb{N}$ such that $k > n \Rightarrow f(k) \leq k^{-c}$.

In the IND-CPA experiment, a polynomial-time attacker queries a certain oracle with pairs of plaintexts. The oracle either always returns an encryption of the

first message, or always returns an encryption of the second message; the goal of the attacker is to find which. The intuition behind IND-CPA security is that, for a polynomial-time observer, the encryption function reveals nothing about the underlying plaintext. A precise definition follows.

Definition 2.6 Let $\mathcal{E} = (R_k, pub, priv, R_e, Enc, Dec)$ be an asymmetric encryption scheme. Consider the following experiment:

- a security parameter η is fixed
- a random key R_k^1 is generated by running $R_k(\eta)$
- the corresponding public key $pub(R_k^1)$ is computed and published
- a random bit R_b^1 is chosen using a random generation algorithm R_b; the value of b remains secret
- a probabilistic algorithm Alg receives as input a security parameter η and the public key $pub(R_k^1)$, and returns 0 or 1. Alg may use an oracle $\mathcal{O}$ such that, for any valid plaintexts m_1, m_2,

$$\mathcal{O}(m_1, m_2) = Enc(m_b, pub(R_k^1), R_e^i),$$

where a different R_e^i is sampled from $R_e(\eta)$ each time the oracle is called.

We say that $\mathcal{E}$ is *IND-CPA secure* if, for any such polynomial-time algorithm Alg,

$$P[\mathsf{Alg}(pub(R_k^1)) = 1 \mid R_b^1 = 1] - P[\mathsf{Alg}(pub(R_k^1)) = 1 \mid R_b^1 = 0] \tag{1}$$

is negligible as a function of η. We will also say that an algorithm Alg compromises the IND-CPA security of $\mathcal{E}$ if (1) is not negligible for Alg.

The next theorem describes this intuition in a rigorous (if a little involved) way. It expresses IND-CPA security in terms of probabilistic statements by using the well-known fact that an asymmetric encryption scheme is not IND-CPA secure iff an attacker can compromise its IND-CPA security by performing only one query to the oracle. The resulted is stated using the same notational conventions as above.

Theorem 2.7 *The following are equivalent:*

(1) $\mathcal{E}$ is not IND-CPA secure.

(2) There exist deterministic polynomial-time algorithms $Q, (\cdot)^0, (\cdot)^1, B$ and a probabilistic polynomial-time algorithm R such that

$$P[B(Q^0, Q^1, \mathcal{O}(Q^0, Q^1), R^1) = 1 \mid R_b^1 = 1]$$
$$-P[B(\eta, Q^0, Q^1, \mathcal{O}(Q^0, Q^1), R^1) = 1 \mid R_b^1 = 0]$$

is not negligible as a function of η, where $Q^0 \equiv Q(R_k^1, R^1)^0$ and $Q^1 \equiv Q(R_k^1, R^1)^1$.

(3) There is an algorithm which compromises the IND-CPA security of $\mathcal{E}$ by performing only one query to the oracle.

The proof can be found in the Appendix.

3 Probabilistic reasoning

In this section we present a method for estimating the probability of success of an attack strategy based on the cryptanalytical knowledge of the attacker. This knowledge is represented in the form of probabilistic statements which may express general, abstract properties of cryptographic primitives; our goal is to evaluate the impact of such properties on the security of the protocol. For this we estimate a probability distribution of the random variables represented by expressions - that is, a probability distribution on Ω_η^A - which verifies all the probabilistic statements known by the attacker but otherwise reveals as little information and remains as hard to attack as possible. To assess the impact of properties of the cryptographic primitives used on the security of a protocol one may then compute the probability of success of an attacker whose cryptanalytical knowledge contains those properties using the probability distribution mentioned above. Such an approach may also be used to analyze the security of a protocol when specific cryptographic primitives are used, by using the attacker's cryptanalytical knowledge to express known properties of those primitives.

We consider an attacker A with access to a function $p^A \colon \mathbb{N} \times ((\mathbf{Exp}^A \times \mathbf{B}^A)^*)^2 \to [0,1] \cup \{\bot\}$, which we use to represent his cryptanalytical knowledge. p^A represents the information available to A in the form of probabilistic statements: that is,

$$p^A[\eta, (((E^1, b_1), \ldots, (E^n, b_n)), ((E^{*,1}, b_1^*), \ldots, (E^{*,n^*}, b_{n^*}^*)))] =$$

$$P[E_\eta^1 = b_1, \ldots, E_\eta^n = b_n \mid E_\eta^{*,1} = b_1^*, \ldots, E_\eta^{*,n^*} = b_{n^*}^*]$$

when the attacker's knowledge allows him to compute this probability, and $\bot$ otherwise. In order to model computationally realistic attackers, one may require, for example, that p^A be efficiently computable.

We will denote the (finite) set of probabilistic statements known by the attacker A (*i.e.*, the statements he can obtain by using the function p^A as described above) for security parameter η by $\mathbf{S}_\eta^A$. Henceforth we will describe the knowledge of the attacker about the cryptographic primitives directly as probabilistic statements, using the sets $\mathbf{S}_\eta^A$ instead of the function p^A.

We want to estimate a probability distribution on Ω_η^A given the set $\mathbf{S}_\eta^A$ (representing A's knowledge about the cryptographic primitives). Exactly what is a reasonable estimation is difficult to define, and similar problems have been subject of extensive research, namely on inductive logic [12,17]. In our approach we use Shannon's notion of entropy [24]. Shannon's entropy is widely used in information theory as the standard measure for uncertainty about the outcome of a random experiment, and several information theoretic approaches have been proposed in the context of information security [4,11,22]. As usual, $H(X)$ denotes the entropy of a random variable X, and $H(X \mid Y)$ denotes the entropy of X conditioned on another random variable Y.

Definition 3.1 Given sets $\mathbf{Exp}_\eta^A$, $\mathbf{B}_\eta^A$ and $\mathbf{S}_\eta^A$, we say that $\mathcal{D}$ is a *distribution of maximum entropy* on Ω_η^A for $\mathbf{S}_\eta^A$ if the following conditions are verified:

- if
$$(P[E^1_\eta = b_1, \ldots, E^n_\eta = b_n \mid E^{*,1}_\eta = b^*_1, \ldots, E^{*,n^*}_\eta = b^*_{n^*}] = \rho) \in \mathbf{S}^A_\eta,$$
then
$$P[E^1 = b_1, \ldots, E^n = b_n \mid E^{*,1} = b^*_1, \ldots, E^{*,n^*} = b^*_{n^*}] = \rho$$
when $E^1, \ldots, E^n, E^{*,1}, \ldots, E^{*,n^*} \leftarrow \mathcal{D}$.
- if $\mathcal{D}^*$ is another distribution which respects the first property, and F, F^* are random variables over Ω^A_η with distribution $\mathcal{D}, \mathcal{D}^*$, then $H(F) \geq H(F^*)$.

The following example illustrates the behavior of such a distribution given a few simple probabilistic statements about two Bernoulli random variables.

Example 3.2 Consider two random variables A, B, each of which has values 0 or 1. We will denote the events $A = 1$ and $A = 0$ by A and $\bar{A}$, respectively, and adopt similar conventions for B. We will add probabilistic statements to a set $\mathbf{S}$ and analyze the resulting changes in the distribution of maximum entropy $\mathcal{D}(\mathbf{S})$ which verifies the statements in $\mathbf{S}$. We will use e as the base of the logarithms for the calculus of entropy; the values presented are approximate.

	$\mathcal{D}(\mathbf{S})$				
$\mathbf{S}$	$P[A, B]$	$P[A, \bar{B}]$	$P[\bar{A}, B]$	$P[\bar{A}, \bar{B}]$	$H(A, B)$
$\{P[A] = 0.6\}$	0.3	0.3	0.2	0.2	1.366
$\{P[A] = 0.6,$ $P[A \mid B]\} = 0.7\}$	0.336	0.264	0.144	0.256	1.346
$\{P[A] = 0.6,$ $P[A \mid B] = 0.7,$ $P[B \mid A] = 0.5\}$	0.3	0.3	0.128	0.272	1.34

We will now present an important property of the distribution of maximum entropy which provides a slightly simpler way of computing it and also helps to justify why it is as a reasonable estimation. The following definition will be useful.

Definition 3.3 Let $s \in \mathbf{S}^A_\eta$ be the probabilistic statement
$$P[E^1_\eta = b_1, \ldots, E^n_\eta = b_n \mid E^{*,1}_\eta = b^*_1, \ldots, E^{*,n^*}_\eta = b^*_{n^*}] = \rho,$$
and let
$$\Omega^A_\eta \supseteq F = \{f \in \Omega^A_\eta \colon f(E^1) = b_1, \ldots, f(E^n) = b_n\},$$
$$\Omega^A_\eta \supseteq F^* = \{f \in \Omega^A_\eta \colon f(E^{*,1}) = b^*_1, \ldots, f(E^{*,n^*}) = b^*_{n^*}\}.$$
We will say that the set $\{P^1_s, P^2_s, P^3_s\}$ is *the partition (of Ω^A_η) induced by s*, where $P^1_s = F^* \setminus F$, $P^2_s = F \cap F^*$, and $P^3_s = \Omega_\eta \setminus F^*$. Observe that $\bigcup_{i=1}^3 P^i_s = \Omega_\eta$ and $P^i_s \cap P^j_s = \emptyset$ for any $i \neq j$.

If $\mathbf{S}_\eta^A = \{s_1, \ldots, s_n\}$ is a finite set of cryptographic properties, we say that

$$\bigcup_{i_1=1}^{3} \ldots \bigcup_{i_n=1}^{3} \{P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}\}$$

is *the partition (of Ω_η^A) induced by $\mathbf{S}_\eta^A$*.

The next theorem illustrates the importance of this definition.

Theorem 3.4 *Fix a security parameter η. Let $\mathbf{S}_\eta^A = \{s_1, \ldots, s_n\}$ be the set of probabilistic statements known by the attacker for security parameter η, and consider the partition of Ω_η^A induced by $\mathbf{S}_\eta^A$. Suppose that there is a distribution of maximum entropy $\mathcal{D}$ for $\mathbf{S}_\eta^A$, and fix $(i_1, \ldots, i_n) \in \{1, 2, 3\}^n$. Then, if F is a random element of Ω_η^A sampled with probability distribution $\mathcal{D}$ (*i.e.*, $F \leftarrow \mathcal{D}$), we have*

$$f_1, f_2 \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n} \Rightarrow P[F = f_1] = P[F = f_2].$$

This theorem, whose proof can be found in the Appendix, shows that the estimation of maximum entropy $\mathcal{D}$ gives the same probability to elements of Ω_η^A which A's knowledge does not distinguish. This is a desirable property, since it indicates that in some sense $\mathcal{D}$ does not give the attacker more information than what he can infer from his knowledge. Another reason why this theorem is important is that it provides us with a (slightly) simpler way to compute the distribution $\mathcal{D}$.

Corollary 3.5 *Consider the set of cryptographic properties $\mathbf{S}_\eta^A = \{s_1, \ldots, s_n\}$. Suppose that the function $p^*\colon \{1, 2, 3\}^n \to \mathbb{R}$ is a maximum of*

$$-\sum_{i_1=1}^{3} \ldots \sum_{i_n=1}^{3} |P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}| \cdot p^*(i_1, \ldots, i_n) \log(p^*(i_1, \ldots, i_n)) \tag{2}$$

restricted to the set of functions $p\colon \{1, 2, 3\}^n \to \mathbb{R}$ such that

$$\begin{cases} \sum_{i_1=1}^{3} \ldots \sum_{i_n}^{3} |P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}| \cdot p(i_1, \ldots, i_n) = 1 \\ p(i_1, \ldots, i_n) \geq 0. \end{cases}.$$

Then the distribution $\mathcal{D}$ defined by $F \leftarrow \mathcal{D} \Rightarrow P[F = f] = p(i_1, \ldots, i_n)$, where each i_j is such that $f \in P_{s_j}^{i_j}$, is a distribution of maximum entropy for $\mathbf{S}_\eta^A$.

The problem of maximizing such a function in a set bounded by linear conditions is well studied in linear programming, and there are efficient algorithms for solving it [18]. Of course, since the set $\{(i_1, \ldots, i_n)\colon i_j \in \{1, 2, 3\}, j = 1, \ldots, n\}$ has size 3^n and n may grow as a function of η, this may still not be an efficient way of estimating the distribution. Indeed, the next result states that computing even one probability of this distribution is a NP-hard problem in terms of the number of statements in $\mathbf{S}$. We use the fact that the problem **3SAT** is NP-complete [19].

Theorem 3.6 *Let* **Prob** *be the following problem: given finite sets of expressions* **Exp**, *of bitstrings* **B** *and of rules* **S**, *compute*

$$P[E^1 = b_1, \ldots, E^n = b_n \mid E^{*,1}, \ldots, E^{*,n^*}]$$

where $E^1, \ldots, E^n, E^{*,1}, \ldots, E^{*,n^*} \leftarrow \mathcal{D}$ *and* $\mathcal{D}$ *is a distribution of maximum entropy for the sets* **S**, **Exp**, **B**.

The problem **3SAT** *reduces to* **Prob** *and the complexity of the reduction is polynomial in* $|\mathbf{S}|$. *In particular,* **Prob** *is NP-hard (in* $|\mathbf{S}|$*).*

This theorem states that estimating this probability is NP-hard in the number of cryptographic properties known by the attacker [6]. Despite this fact, it may still be possible to use this distribution of maximum entropy to estimate the probability of success of a given attack strategy, even if it involves relatively complex cryptanalysis.

Example 3.7 It is possible to represent in our model an attack to a 6-round version of DES using the well-known differential cryptanalysis technique. We are able to define a set of algorithms and probabilistic statements about them, and use these properties to estimate (using the distribution of maximum entropy) that, by encrypting 200 pairs of plaintexts, an attacker can guess 30 of the 56 key bits with probability 0.65. This technique is somewhat involved; we present a simplified description of it using our framework in the Appendix. A detailed description of this attack is given in [28].

For the next theorem we consider a polynomial-time attacker A whose goal is to obtain the bitstring represented by a certain expression E. We present a result which may be useful in showing that a certain amount of cryptanalytical knowledge is not enough for A to have a non-negligible probability of success. The result concerns the (asymptotic) entropy of (the random variable represented by) E conditional on the A's guess, in the distribution of maximum entropy on Ω_η^A given $\mathbf{S}_\eta^A$. We first present an intuitive description of the result. Suppose that A is able to use the knowledge obtained about messages exchanged in the network, together with its own knowledge about the cryptographic primitives, to correctly uncover the bitstring corresponding to the expression E with a non-negligible probability.

We present an attacker $\tilde{A}$ which emulates A, and thus has the same computational complexity and the same probability of success. However, $\tilde{A}$ may have access to different cryptographic properties and even use different algorithms. In particular, there is an expression $g \in \mathbf{Exp}^{\tilde{A}}$ which represents $\tilde{A}$'s guess [7]. The result then states that $\tilde{A}$'s cryptanalytical knowledge is sufficient to show that its guess is in a certain way non-negligibly correlated to the bitstring represented by E. A precise statement of the result follows. The proof is given in the Appendix.

[6] However, this does not necessarily imply that the problem is NP-hard in the security parameter, for it is possible that the $\mathbf{S}_\eta^A$ does not grow when we increase the security parameter.

[7] Note that this implies that $\tilde{A}$'s guess is completely determined by the random generation algorithms executed during the protocol either by $\tilde{A}$ or the honest principals.

Theorem 3.8 *Let $E \in \mathbf{Exp}$ be such that $\mathcal{A}(E_\eta)$ (the set of bitstrings which E_η may represent for security parameter η) verifies the following:*

- *given a bitstring b and a security parameter η, there are efficient algorithms for deciding whether or not $b \in \mathcal{A}(E_\eta)$;*
- *there are $N \in \mathbb{N}, k > 1$ satisfying $\eta > N \Rightarrow |\mathcal{A}(E_\eta)| > k^\eta$ (*i.e.*, $\mathcal{A}(E_\eta)$ grows exponentially in η)* [8] .

Suppose that there is a polynomial-time attacker A whose probability of guessing the bitstring represented by E_η is a non-negligible, computable function of η. Then, there exists an attacker $\tilde{A}$ and an expression $g \in \mathbf{Exp}^{\tilde{A}}$ representing $\tilde{A}$'s guess such that

$$H_{max}(E_\eta) - H(E \mid g) \tag{3}$$

*is non-negligible as a function of η, where $E, g \leftarrow \mathcal{D}$ and $\mathcal{D}$ is the distribution of maximum entropy on $\Omega_\eta^{\tilde{A}}$ which verifies the all statements in $\mathbf{S}_\eta^{\tilde{A}}$ (*i.e.*, the set of cryptanalytical properties known by $\tilde{A}$) and $H_{max}(E_\eta) = \log |\mathcal{A}(E_\eta)|$ is the maximum entropy of a probability distribution on $\mathcal{A}(E_\eta)$.*

This theorem may be useful in proving that a given set of known cryptographic properties $\mathbf{S}_\eta^A$ are not enough to render a protocol insecure. In fact, suppose that (3) is negligible for A's guess g. If that attacker has a non-negligible probability of success, there must be some other relevant, polynomial-time computable property which would provide non-negligible information about E_η. If given the sets $\mathcal{R}^A, \mathcal{F}^A$ there is no way to compute a guess g such that (3) is non-negligible, then the cryptographic properties used by A do not compromise the security of the protocol by themselves. Of course, when we consider specific cryptographic primitives, they may verify other properties which allow an attacker to be successful with non-negligible probability.

Example 3.9 Our last example uses a version of the Mastermind game, generalized in our framework as follows. We will use a random generation algorithm $c \in \mathcal{R}$ such that $c(\eta)$ is a random element of $\{1, \ldots, 2\eta\}$. For security parameter η, the first player creates a code $C \equiv (c^1 \ldots c^\eta)$ by executing the random generation algorithm η times. Another player A (from attacker) tries to guess C by querying the first player with codes $C^k \equiv (c^{1+k\eta} \ldots c^{\eta+k\eta})$. A obtains in return $R(C, C^k)$, given by:

$$R(C, C^k) = \#\{j \in \{1, \ldots, \eta\} \mid c^j = c^{j+k\eta}\}.$$

$R(C, C^k)$ (the number of red balls) represents the number of slots in which C^k has the same color as the correct code C [9] .

The following cryptographic properties will be used:

$$P[c^j = n \mid c^{j+k\eta} = n, R(C, C^k) = 0] = 0, j = 1, \ldots, \eta, \tag{4}$$

[8] Intuitively, this last condition demands that $\mathcal{A}(E_\eta)$ is so large that guessing the bistring corresponding to E_η is hard.

[9] The white balls may be ignored for the purposes of this example.

$$P[R(C, C^k) = 0 \mid c^1 = b_1, c^{1+k\eta} = b'_1, \ldots, c^{\eta} = b_\eta, c^{\eta+k\eta} = b'_\eta] = 1, \quad (5)$$

for all $b_1, b'_1, \ldots, b_\eta, b'_\eta$ such that $b_1 \neq b'_1, \ldots, b_\eta \neq b'_\eta$.

Consider the following attack strategy. A randomly generates codes C^k and uses them in its queries. If $R(C, C^k) \neq 0$, he discards the information obtained. Otherwise, he concludes from the equation above that $c^j \neq c^{k\eta+j}$ for each j (using (4)). Suppose that A performs $4\eta \log \eta$ such queries and hopes to find the right code by excluding, for each slot, all but one possible color (note that this attack has less than quadratic complexity). We conservatively estimate (see Appendix) the probability of success to be $1/e$, and it is easy to see, by calculations similar to the ones used in the proof of Theorem 3.8, that this implies a non-negligible decrease in the entropy of C.

By contrast, suppose that instead of (4), (5) we equip the attacker with property

$$P[c^j = n \mid c^{j+k\eta} = n, R(C, C^k) = \eta] = 1, j = 1, \ldots, \eta. \quad (6)$$

Note that, in the distribution of maximum entropy, this does not give the attacker any knowledge about the secret code if $R(C, C^k) \neq \eta$. Consider the following (natural) attack strategy. The attacker performs a polynomial number of queries, η^m, and then either returns the correct bitstring (if he found it during those queries) or some random bitstring otherwise. This guess is correct with at most probability $\frac{\eta^m}{(2\eta)^\eta}$ for some C; all other possible codes have the same probability of being correct. Again by a process similar to the one used in the proof of Theorem 3.8, the entropy reduction provided by the attacker's knowledge can then be shown to be a negligible function η, thus confirming that the property (6) does not compromise the security of the Masterind "protocol" against this strategy.

4 Conclusion

In this work we presented a formal model for analyzing the security of cryptographic protocols against attackers who may use cryptanalysis. This model allows the representation of cryptanalytical capabilities and partial probabilistic information about secret messages. It is a very general and flexible model: one may specifiy cryptographic properties and use them to describe or find an attack, but if one does not, then it behaves essentially as a symbolic model. For illustration purposes, we showed how IND-CPA security may be expressed in terms of the probabilistic statements considered. Finally, we proposed a way of estimating the attacker's probability of success. To obtain this estimation we define the probability distribution of maximum entropy and present a few properties and examples which justify this approach. This technique may also be used to study whether or not a weakness of the cryptographic primitives used compromises the security of a given protocol. Another advantage of the approach is that it allows, to some extent, a separate study of weaknesses of cryptographic primitives and the security of protocols, since one may simply write properties of the cryptographic primitives deemed relevant and verify if the protocol is still secure when the attacker explores them. Possible inter-

esting applications include the study of side-channel attacks, in which the attacker explores the physical implementation of algorithms to obtain partial information about secret data (such as a key).

Several problems and open questions remain. First of all, one needs to give a description of the properties of the cryptographic primitives used, which should typically be quite hard. Thus, for each cryptographic primitive, finding a suitable set of properties to describe its weaknesses is a non-trivial and interesting problem. Examples of attacks using properties of real cryptographic primitives are typically hard to manage given the complexity of the computations involved, though implementing the model could help overcome this difficulty. Finding an attack has exponencial complexity on the length of the attack, as is typically the case for most tools used for this task. Another interesting and relevant problem is to find alternative methods for estimating probability distributions of bitstrings based on the knowledge of the attacker. Though the proposal presented here has several advantages and is relatively natural, other methods may be more efficient or reflect better the power and knowledge of the attacker.

As mentioned in the introduction, the work presented here tries to bridge the gap between the symbolic and computational approaches to the analysis of security protocols. This is a line of work which has been widely explored since [1]. Different proposals have dealt with this problem from different perspectives. In [2], algebraic properties of the cryptographic functions are included in the model. Several formal models for analysing guessing attacks have been proposed [21,15], some of which include probabilistic analysis [3]. Work has also been aimed at the computational validation of the cryptographic assumptions of the Dolev-Yao model [16], and even at automatically obtaining or checking cryptographically sound proofs of the security of protocols - namely using the CryptoVerif tool [9] and the BPW model [26]. Of course, a thorough comparison of our approach with other proposals in the literature (both extensions of the Dolev-Yao such as those mentioned above and other information-theoretic approaches to security such as those presented in [4,11,22]) shall deserve close attention in the near future.

References

[1] Abadi, M. and P. Rogaway, *Reconciling two views of cryptography*, Journal of Cryptology (2002), 103–127.

[2] Abadi, M. and V. Cortier, *Deciding knowledge in security protocols under equational theories*, Proceedings 31st Int. Coll. Automata, Languages and Programming (ICALP 2004) (2004), 46–58.

[3] Adão, P., P. Mateus, T. Reis and L. Viganò, *Towards a quantitative analysis of security protocols*, Electronic Notes in Theoretic Computer Science **164** (2006), 3–25.

[4] Aldini, A. and A. Di Pierro, *Estimating the maximum information leakage*, International Journal of Information Security **7** (2008), 219–242.

[5] Armando, A., D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, L. Cuellar, P. Drielsma, P. Heám, O. Kouchnareko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò and L. Vigneron, *The AVISPA Tool for Automated Validation of Internet Security Protocols and Applications*, Lecture Notes in Computer Science **3576** (2005), Springer-Verlag.

[6] Backes, M. and B. Pfitzmann, *A cryptographically sound security proof of the Needham-Schröder-Lowe public-key protocols*, IEEE Journal on Selected Areas in Communications (2004), 2075-2086.

[7] Backes, M. and B. Pfitzmann, *Relating symbolic and cryptographic secrecy*, IEEE Transactions on Dependable and Secure Computing **2** (2005), 109–123.

[8] Bellovin, S.M. and M. Merritt, *Encrypted key exchange: Password-based protocols secure against dictionary attacks*, In Proc. IEEE Computer Society Symposium on Research in Security and Privacy (1992), 72–84.

[9] Blanchet, B. *A computationally sound mechanized prover for security protocols*, Proceedings of the 2006 IEEE Symposium on Security and Privacy (2006).

[10] Blanchet, B., *An efficiente cryptographic protocol verifier based on Prolog rules*, 14th IEEE Computer Security Foundations WS (2001), IEEE Computer Society, 82–96.

[11] Chatzikokolakis and C. Palamidessi, *A framework to analyze probabilistic protocols and its application to the partial secrets exchange*, Theoretical Computer Science **389** (2007), 512–527.

[12] Cox, R.T., "The Algebra of Probable Inference", Johns Hopkins University, 1961.

[13] Cremers, C. "Scyther - Semantics and Verification of Security Protocols", Ph.D. thesis (2006), Computer Science Department, Eindhoven University of Technology.

[14] Dolev, D. and A. Yao, *On The Security of Public Key Protocols*, IEEE Transactions on Information Theory (1983), 198–208.

[15] Drielsma, P., S. Mödersheim and L. Viganò, *A formalization of off-line guessing for security protocol analysis*, Lecture Notes in Artificial Intelligence **3452** (2005), Springer, 363–379.

[16] Herzog, J., "Computational Soundness for Standard Assumptions of Formal Cryptography", Ph.D. thesis, Massachussets Institute of Technology (2004).

[17] Fine, T.L., *Theories of probability: an examination of foundations*, New York: Academic Press, 1973.

[18] Karmakar, N., *A new polynomial-time algorithm for linear programming*, Combinatorica **4** (1984), 373–395.

[19] Karp, R. M., *Reducibility among combinatorial problems*, Complexity of Computer Computations, R. Miller and J. Thatcher, eds. Plenum Press, New York, 85–103.

[20] Lincoln, P., J. C. Mitchell, M. Mitchell and A. Scedrov, *A Probabilistic Polynomial-Time Framework For Protocol Analysis*, Proceedings of the 5th ACM Conference on Computer and Communications Security (1998), M. Reiter, 112–121.

[21] Lowe, G., *Analysing protocols subject to guessing attacks*, WS on Issues in the Theory of Security, 2002.

[22] Malacaria, P. and H. Chen, *Lagrange multipliers and maximum information leakage in different observational models*, Proceedings of the 3rd WS on Programming Languages and Analysis for Security, 2008.

[23] Montalto, B., "Modelling an Attacker with Cryptanalytical Capabilities", Ms.C. thesis (2008), IST, TU Lisbon, Portugal, `http://wslc.math.ist.utl.pt/ftp/pub/MontaltoB/08-M-MScThesis.pdf`.

[24] Shannon, C.E., *A mathematical theory of communication*, Bell System Technical Journal **27** (1948), 379–423.

[25] Shoup, V., *On formal models for secure key exchange (version 4)*, Revision of IBM Research Report RZ 3120 (1999).

[26] Sprenger, C. and David Basin, *Cryptographically-sound protocol-model abstractions*, Proceedings of Computer Security Foundations (CSF '08) (2008), IEEE Computer Society.

[27] Sprenger, C., M. Backes, D. Basin, B. Pfitzmann and M. Waidner, *Cryptographically sound theorem proving*, Proceedings of 19th Computer Science Foundation WS (2006).

[28] Stinson, D., "Cryptography: Theory and Practice", CRC Press, 1995.

Appendix

Theorem 2.7 *The following are equivalent:*

(1) $\mathcal{E}$ is not IND-CPA secure.

(2) There exist deterministic polynomial-time algorithms $Q, (\cdot)^0, (\cdot)^1, B$ and a probabilistic polynomial-time algorithm R such that

$$\begin{aligned} &P[B(Q^0, Q^1, \mathcal{O}(Q^0, Q^1), R^1) = 1 \mid R_b^1 = 1] \\ -&P[B(\eta, Q^0, Q^1, \mathcal{O}(Q^0, Q^1), R^1) = 1 \mid R_b^1 = 0] \end{aligned} \tag{7}$$

is not negligible as a function of η, where $Q^0 \equiv Q(R_k^1, R^1)^0$ and $Q^1 \equiv Q(R_k^1, R^1)^1$.

(3) There is an algorithm which compromises the IND-CPA security of $\mathcal{E}$ by performing only one query to the oracle.

Proof. Implication (3) $\Rightarrow$ (1) is trivial. Implication (2) $\Rightarrow$ (3) is also simple.

The proof of (1) $\Rightarrow$ (2) is harder. For simplicity, we will assume fixed a security parameter η and drop it from the notation when there is no ambiguity. We also abbreviate $K \equiv R_k^1$ and write $\mathcal{O}_b$ to refer to the oracle $\mathcal{O}$ using bit b in the computations, so that $\mathcal{O}_b(m^0, m^1) = Enc(\eta, m^b, K, R_e^i)$. Let Alg be an algorithm which compromises the IND-CPA security of $\mathcal{E}$. Observe that, since Alg is a polynomial-time algorithm, there is a fixed number c such that, for security parameter η, Alg performs at most η^c queries to the oracle. We may assume, without loss of generality, that Alg always performs exactly η^c queries to the oracle and that the random choices of the attacker throughout the computation (*i.e.*, the η^c steps) are determined by a single bitstring R_r^1 sampled from a random generation algorithm R_r.

We may write the first pair of plaintexts with which Alg queries the oracle as $(Q(1, K, R_r^1))^0, (Q(1, K, R_r^1))^1$, where $Q, (\cdot)^0, (\cdot)^1 \in \mathcal{F}^A$ are functions available to the attacker A. More generally, for $i = 1, \ldots, \eta^c$, let

$$q_i = Q(i, K, R_r^1, q_1^0, q_1^1, o_1, \ldots, q_{i-1}^0, q_{i-1}^1, o_{i-1}) \tag{8}$$

represent the pair of messages used in the i-th query to the oracle. Each q_i is computable in polynomial-time: in the worst case, q_i needs to emulate the executions of $q_1, \ldots, q_{i-1}$ using random data R_r^1 (note that the output may depend on computations performed by these algorithms). Since $i < \eta^c$, this involves computing at most a polynomial number of algorithms, each of which is computable in polynomial time.

Alg's response can be represented using an algorithm B_F as follows:

$$B_F(\eta, K, R_r^1, q_1^0, q_1^1, o_1, \ldots, q_{\eta^c}^0, q_{\eta^c}^1, o_{\eta^c}). \tag{9}$$

Since Q, B_F are polynomial-time algorithms, we may assume, without loss of generality, that they are total: if they are not, we may specify some predetermined result and stipulate that they will return that result if they do not return another answer after a certain polynomial number of operations.

Now suppose that, instead of using the same bit R_b^1 in all queries, the oracle $\mathcal{O}$ uses the bit b_i in the i-th query (in other words, in the i-th query Alg is querying $\mathcal{O}_{b_i}$). Let $\delta_{i \geq j} = 1$ if $i \geq j$ and 0 otherwise. Note that

$$\begin{aligned} &P[\mathsf{Alg}(\eta, K) = 1 \mid b_i = 1] - P[\mathsf{Alg}(\eta, K) = 1 \mid b_i = 0] \\ &= \sum_{j=1}^{\eta^c} (P[\mathsf{Alg}(\eta, K) = 1 \mid b_i = \delta_{i \geq j}] - P[\mathsf{Alg}(\eta, K) = 1 \mid b_i = \delta_{i \geq j+1}]), \end{aligned} \tag{10}$$

because the sum on the right-hand side is telescopic. The left-hand side is non-negligible by hypothesis, and thus so is the right hand side. The assumption that Q, B_F return an answer in polynomial time for every possible input is necessary here, since we are running these algorithms with inputs that would be impossible in the original setting.

Querying the oracle $\mathcal{O}_b$ with q^0, q^1 is the same as computing $Enc(\eta, q^b, K, R_e(\eta))$. Thus, if in some step of the algorithm Alg, instead of querying the oracle $\mathcal{O}_b$ with plaintexts q_i^0, q_i^1, we encrypt q_i^b with public key k and random data $r_i \leftarrow R_e(\eta)$, the probability distribution of the output does not change.

Consider the following modified version Alg' of the algorithm Alg. For each η, Alg' receives an additional argument $j \in \{1, \ldots, \eta^c\}$. Alg' mimics the execution of Alg, except for the following. In the first $j-1$ queries, Alg' computes an encryption of the first message of the pair (instead of querying the oracle $\mathcal{O}$) and uses that result in the following computations. In the j-th query, Alg' queries the oracle just as Alg would; after the j-th query, Alg' computes encryptions of the second message of the pair instead of querying the oracle.

It is easy to see that

$$P[\mathsf{Alg}'(\eta, k, j) = 1 \mid b = 1] - P[\mathsf{Alg}'(\eta, k, j) = 1 \mid b = 0]$$

$$= P[\mathsf{Alg}(\eta,k)=1 \mid b_i=\delta_{i\geq j}] - P[\mathsf{Alg}(\eta,k)=1 \mid b_i=\delta_{i\geq j+1}].$$

Suppose now that Alg'' is an algorithm which chooses $j=1,\ldots,\eta^c$ randomly (with uniform distribution) and then executes $\mathsf{Alg}'(\eta,k,j)$. We obtain:

$$P[\mathsf{Alg}''(\eta,k)=1 \mid b=1] - P[\mathsf{Alg}''(\eta,k)=1 \mid b=0]$$

$$= \sum_{l=1}^{\eta^c} P[j=l]\cdot(P[\mathsf{Alg}'(\eta,k,l)=1 \mid b=1] - P[\mathsf{Alg}'(\eta,k,l)=1 \mid b=0])$$

$$= \frac{1}{\eta^c}\sum_{j=1}^{\eta^c}(P[\mathsf{Alg}(\eta,k)=1 \mid b_i=\delta_{i\geq j}] - P[\mathsf{Alg}(\eta,k)=1 \mid b_i=\delta_{i\geq j+1}])$$

$$= \frac{1}{\eta^c}(P[\mathsf{Alg}(\eta,k)=1 \mid b=1] - P[\mathsf{Alg}(\eta,k)=1 \mid b=0]).$$

Thus, by (10) and the equalities above, we conclude that

$$P[\mathsf{Alg}''(\eta,k)=1 \mid b=1] - P[\mathsf{Alg}''(\eta,k)=1 \mid b=0]$$

is non-negligible.

It is clear that Alg'' is a polynomial-time algorithm. Furthermore, Alg'' involves only one query to the oracle, compromises the IND-CPA security of $\mathcal{E}$, and it is easy to see that Alg'' may be written in terms of algorithms $Q,(\cdot)^0,(\cdot)^1,B,R$ as desired: $R(\eta)$ returns the random generated data R_r, the random $j\in\{1,\ldots,\eta^c\}$ used by Alg'' and the random data used in the encryptions $R_e^1,\ldots,R_e^{\eta^c}$, Q generates the pairs of plaintexts with which Alg'' queries the oracle and B computes the final answer. This concludes the demonstration. □

Theorem 3.4 *Fix a security parameter η. Let $\mathbf{S}_\eta^A=\{s_1,\ldots,s_n\}$ be the set of probabilistic statements known by the attacker for security parameter η, and consider the partition of Ω_η^A induced by $\mathbf{S}_\eta^A$. Suppose that there is a distribution of maximum entropy $\mathcal{D}$ for $\mathbf{S}_\eta^A$, and fix $(i_1,\ldots,i_n)\in\{1,2,3\}^n$. Then, if F is a random element of Ω_η^A sampled with probability distribution $\mathcal{D}$ (*i.e.*, $F\leftarrow\mathcal{D}$), we have*

$$f_1,f_2\in P_{s_1}^{i_1}\cap\ldots\cap P_{s_n}^{i_n} \Rightarrow P[F=f_1]=P[F=f_2]. \tag{11}$$

Proof. For each $s\in\mathbf{S}_\eta^A$ and each $f\in\Omega_\eta^A$, we write $i(s,f)$ for the number $i\in\{1,2,3\}$ such that $f\in P_r^{i(s,f)}$.

We will consider an arbitrary distribution $\mathcal{D}$ which verifies all properties in $\mathbf{S}_\eta^A$ and obtain another distribution $\mathcal{D}'$. We will then show that properties in $\mathbf{S}_\eta^A$ still hold for $\mathcal{D}'$, which verifies (11) and has greater entropy than $\mathcal{D}$.

Let $F\leftarrow\mathcal{D}, F'\leftarrow\mathcal{D}'$. We define the distribution $\mathcal{D}'$ by

$$P[F'=f]=\frac{P[F\in P_{s_1}^{i(s_1,x)}\cap\ldots\cap P_{s_n}^{i(s_n,x)}]}{|P_{s_1}^{i(s_1,x)}\cap\ldots\cap P_{s_n}^{i(s_n,x)}|}.$$

It is clear that $\mathcal{D}'$ verifies (11). To see that $\mathcal{D}'$ verifies all properties in $\mathbf{S}_\eta^A$, let

$$s_j=(P[E_\eta^1=b_1,\ldots,E_\eta^n=b_n \mid E_\eta^{*,1}=b_1^*,\ldots,E_\eta^{*,n^*}=b_{n^*}^*]=\rho)\in\mathbf{S}_\eta^A.$$

Now, if $F'\leftarrow\mathcal{D}'$, we have

$$P[F'\in P_{s_j}^2 \mid F'\in P_{s_j}^1\cup P_{s_j}^2]=\frac{P[F'\in P_{s_j}^2]}{P[F'\in P_{s_j}^1\cup P_{s_j}^2]}.$$

For each sequence $(i_1,\ldots,i_n)\in\{1,2,3\}^n$, it is clear that $P[F'\in P_{s_1}^{i_1}\cap\ldots\cap P_{s_n}^{i_n}]=P[F\in P_{s_1}^{i_1}\cap\ldots\cap P_{s_n}^{i_n}]$.

From this, by setting $i_j=2$ and summing over all other i_k, we conclude that $P[F'\in P_{s_j}^2]=P[F\in P_{s_j}^2]$, and thus

$$P[F'(E^1)=b_1,\ldots,F'(E^n)=b_n \mid F'(E^{*,1})=b_1^*,\ldots,F'(E^{*,n^*})=b_{n^*}^*]$$

$$= P[F(E^1) = b_1, \ldots, F(E^n) = b_n \mid F(E^{*,1}) = b_1^*, \ldots, F(E^{*,n^*}) = b_{n^*}^*].$$

Since we know that $\mathcal{D}$ verifies properties $\mathbf{S}_\eta^A$, we conclude that $\mathcal{D}'$ does too.

We now check that $H(F') \geq H(F)$. From Jensen's inequality for convex functions, we know that

$$\sum_{i=1}^{n} x_i = k \Rightarrow -\sum_{i=1}^{n} x_i \log x_i \leq k \log \frac{n}{k}.$$

Thus, for each sequence $(i_1, \ldots, i_n) \in \{1,2,3\}^n$, we obtain

$$- \sum_{f \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}} P[F = f] \cdot \log P[F = f]$$

$$\leq P[F \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}] \cdot \log \frac{|P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}|}{P[X \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}]}$$

$$= \sum_{f \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}} \frac{P[F \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}]}{|P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}|} \cdot \log \frac{|P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}|}{P[F \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}]}$$

$$= - \sum_{f \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}} P[F' = f] \cdot \log P[F' = f].$$

Using the inequality above, the desired result comes from

$$H(F') = - \sum_{i_1, \ldots, i_n} \sum_{f \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}} P[F' = f] \cdot \log P[F' = f]$$

$$\geq - \sum_{i_1, \ldots, i_n} \sum_{f \in P_{s_1}^{i_1} \cap \ldots \cap P_{s_n}^{i_n}} (P[F = f] \cdot \log P[F = f] = H(F).$$

□

Theorem 3.6 *Let* **Prob** *be the following problem: given finite sets of expressions* **Exp**, *of bitstrings* **B** *and of rules* **S**, *compute*

$$P[E^1 = b_1, \ldots, E^n = b_n \mid E^{*,1}, \ldots, E^{*,n^*}]$$

where $E^1, \ldots, E^n, E^{*,1}, \ldots, E^{*,n^*} \leftarrow \mathcal{D}$ *and* $\mathcal{D}$ *is a distribution of maximum entropy for the sets* **S**, **Exp**, **B**.

The problem **3SAT** *reduces to* **Prob** *and the complexity of the reduction is polynomial in* $|\mathbf{S}|$. *In particular,* **Prob** *is NP-hard (in* $|\mathbf{S}|$*).*

Proof. We prove that, given an algorithm for deciding in polynomial-time (in $|\mathbf{S}|$) whether a certain probability is 0 or 1, one may obtain a polynomial-time algorithm for solving the problem **3SAT**. Let

$$\varphi = (l_{11} \vee l_{12} \vee l_{13}) \wedge \ldots \wedge (l_{n1} \vee l_{n2} \vee l_{n3})$$

be an instance of the problem **3SAT** (*i.e.*, a propositional formula consisting of a conjunction of disjunctions of three literals). Denote by $S(\varphi) = \{s_1, \ldots, s_k\}$ the set of propositional symbols present in φ.

We will set $\mathbf{B} = \{0,1\}$, $\mathbf{Exp} = \{s_1, \ldots, s_k, c_1, \ldots, c_n, \varphi\}$. Intuitively, the expression s_i represents the truth value of the propositional symbol s_i, c_i represents the truth value of the i-th clause $c_i = (l_{i1} \vee l_{i2} \vee l_{i3})$, and φ represents the truth value of the formula $\varphi = c_1 \wedge \ldots \wedge c_n$. For each $i = 1, \ldots, n$ and each $j = 1, 2, 3$, let $s_{ij} \in S(\varphi)$ be such that either $l_{ij} = s_{ij}$ or $l_{ij} = \neg s_{ij}$. Furthermore, let $b_{ij} = 1$ if $l_{ij} = s_{ij}$ and 0 if $l_{ij} = \neg s_{ij}$.

The set of rules **S** describes how the value of each expression c_i may be computed from the expressions $s_1, \ldots, s_k$ and how φ may be computed from $c_i, i = 1, \ldots, n$. Thus, **S** contains the properties:

$$\begin{gathered} P[c_i = 1 \mid s_{ij} = b_{ij}] = 1, j \in \{1,2,3\}, i \in \{1, \ldots, n\} \\ P[c_i = 0 \mid s_{i1} = 1 - b_{ij} \text{ for all } j \in \{1,2,3\}] = 1, i \in \{1, \ldots, n\}, \end{gathered} \tag{12}$$

and

$$\begin{gathered} P[\varphi = 0 \mid c_i = 0] = 1, i \in \{1, \ldots, n\}, \\ P[\varphi = 1 \mid c_1 = 1, \ldots, c_n = 1]. \end{gathered} \tag{13}$$

It is clear that the original instance of **3SAT** may be polynomially converted in this instance of **Prob**. Thus, we need only show that

$$\varphi \text{ is satisfiable } \iff P[\varphi = 1] > 0 \text{ in } \mathcal{D}(\mathbf{S}), \tag{14}$$

where $\mathcal{D}(\mathbf{S})$ is the distribution of maximum entropy which verifies all probabilistic statements in $\mathbf{S}$. In order to do this, we first note that each $f \in \Omega$ determines an assignment which attributes "true" to s_i if $f(s_i) = 1$ and "false" otherwise.

Let F be a random varible taking values in Ω, and let $F_S = (F(s_1), \ldots, F(s_k))$, $F_{C,\varphi} = (F(c_1), \ldots, F(c_n), F(\varphi))$, so that F is determind by $(F_S, f_{C,\varphi})$. It is clear from equations (12), (13) that if F verifies all probabilistic statements in $\mathbf{S}$ then $F_{C,\varphi}$ is determined by F_S. Thus, we have $H(F) = H(F_S)$, and it becomes clear that F_S has uniform probability distribution (when considering the distribution of maximum entropy $\mathcal{D}(\mathbf{S})$ on Ω).

(14) is now clear: if φ is satisfiable, then there is at least one assignment which satisfies it, and if f is such that $f_S = (f(s_1), \ldots, f(s_n))$ is the value of F_S which induces that assignment, then $f(\varphi) = 1$ and

$$P[\varphi = 1] \geq P[F_S = f_S] = 1/2^k > 0.$$

On the other hand, if φ is not satisfiable, then for all f we have $P[\varphi = 1, F_S = f_S] = 0$, and hence $P[\varphi = 1] = 0$. □

Example 3.7 (Cryptanalysis of DES) In this example we show that it is possible to represent in our model an attack to a 6-round version of DES using the well-known differential cryptanalysis technique. We refer the reader to [28] for a detailed description of the DES encryption scheme and this cryptanalysis technique. For simplicity, we will ignore the security parameter during this description.

We will use the following algorithms. Again for simplicity, instead of describing all the algorithms themselves, we define abbreviations representing the expressions in $\mathbf{Exp}^{\mathcal{A}}$ which we will use.

- $K \in \mathcal{R}$ is the key generation algorithm. We will assume throughout this example that K^1 represents the key being used in the encryption (which the attacker wants to uncover);
- $\mathbf{b}, \mathbf{b'}$ are two specific bitstrings which the attacker will use in the construction of pairs of plaintexts;
- $L^m(b), L^{*,m}(b)$ (resp. $R^m(b), R^{*,m}(b)$) represent two randomly generated 32-bits bitstrings whose x-or is the bitstring b. We write $P^m = L^m(\mathbf{b})R^m(\mathbf{b'})$, $P^{*,m} = L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'})$ to represent the pair of plaintexts obtained from the m-th execution of these algorithms.
- $\oplus \in \mathcal{F}$ computes the x-or of two bitstrings. We will use infix notation for $\oplus$;
- For each $i = 1, \ldots, 6$ and each $j = 1, \ldots, 8$, $IS_{i,j}(P, K)$ (resp. $OS_{i,j}(P, K)$) represents bitstrings used during the computation of the encryption of a plaintext P; $K^1_{i,j}$ represents six key bits used in a certain step of the encryption;
- $L^m_3, E^m_3, E^m_{O,j}, E^m_{I,j}, E^{*,m}_{I,j} \in \mathbf{Exp}^{\mathcal{A}}$ are certain expressions needed in the cryptanalysis procedure which depend on the input plaintexts $P^m, P^{*,m}$ and the corresponding ciphertexts. As such, the attacker can compute the bitstring corresponding to these expressions.
- For each $j = 1, \ldots, 8$, and each $b, b' \in \mathbb{Z}_2^6$, $c \in \mathbb{Z}_2^4$, $test_j(b, b', c)$ represents a set of 6-bit bitstrings.
- For each j, b, b', c as described above and each $k \in \mathbb{Z}_2^6$, $contains(test_j(b, b', c), k)$ is 1 if k belongs to the set represented by $test_j(b, b', c)$ and 0 otherwise.
- For $(j_1, \ldots, j_5) = (2, 5, 6, 7, 8)$, $filters(E^m_{I,j_1}, E^{*,m}_{I,j_1}, E^m_{O,j_1}, \ldots, E^m_{I,j_5}, E^{*,m}_{I,j_5}, E^m_{O,j_5})$ is either 0 or 1.

The main cryptographic properties which we will need are the following:

$$P[L^m_3 = b \mid E^m_3 = b] = \frac{1}{16}, \text{ for } b \in \mathbb{Z}_2^6; \tag{15}$$

$$\begin{aligned} P[OS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'}) \oplus OS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^{m'}) = E^m_{O,j} \\ \mid L^m_3 = b, E^m_3 = b] = 1 \text{ for } j \in \{2, 5, 6, 7, 8\}; \end{aligned} \tag{16}$$

$$P[IS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'}) = b \mid E^m_{I,j} = b] = 1; \tag{17}$$

$$P[IS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^{m'}) = b \mid E^{*,m}_{I,j} = b] = 1 \tag{18}$$

(note that the attacker cannot compute $IS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'})$ directly, as it depends on the unknown key $K^{m'}$);

$$\begin{aligned} P[&contains(test_j(IS_{j,k}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'}),\\ &\qquad IS_{j,k}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^{m'}), b_O), K^{m'}_{i,j})\\ &\mid OS_{6,k}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'}) \oplus OS_{6,k}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^{m'}) = b_O] = 1; \end{aligned} \tag{19}$$

$$\begin{aligned} P[&contains(test_j(IS_{j,k}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'}),\\ &\qquad IS_{j,k}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^{m'}), c), k] = \frac{1}{16}, \end{aligned} \tag{20}$$

where $c \in \mathbb{Z}_2^4$, $k \in \mathbb{Z}_2^6$ and we assume that either c is different from the bitstring represented by $OS_{6,k}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'}) \oplus OS_{6,k}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^{m'})$ or k is different from the bitstring represented by $K^{m'}_{i,j}$ [10];

$$\begin{aligned} P[&filters(E^m_{I,j_1}, E^{*,m}_{I,j_1}, E^m_{O,j_1}, \ldots, E^m_{I,j_5}, E^{*,m}_{I,j_5}, E^m_{O,j_5}) = 1\\ &\mid L^m_3 = b, E^m_3 = b] = 1 \end{aligned} \tag{21}$$

and

$$\begin{aligned} P[&filters(E^m_{I,j_1}, E^{*,m}_{I,j_1}, E^m_{O,j_1}, \ldots, E^m_{I,j_5}, E^{*,m}_{I,j_5}, E^m_{O,j_5}) = 0\\ &\mid L^m_3 = b', E^m_3 = b] = 5/8, b \neq b'. \end{aligned} \tag{22}$$

We will tacitly assume that the attacker knows probabilistic statements which imply that if one replaces a subexpression s of an expression E by another subexpression s' which represents the same bitstring then the bitstring represented by E does not change. For the sake of brevity, we omit such details here.

We now describe the strategy of the attacker. The attacker generates pairs of plaintexts $P^m = L^m(\mathbf{b})R^m(\mathbf{b'})$, $P^{*,m} = L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'})$. From these plaintexts and corresponding ciphertexts, he computes $E^m_{0,j}, E^m_{I,j}, E^{*,m}_{I,j}$ for $j = 2, 5, 6, 7, 8$, and uses the results to compute $filters^m \equiv filters(E^m_{I,j_1}, E^{*,m}_{I,j_1}, E^m_{O,j_1}, \ldots, E^m_{I,j_5}, E^{*,m}_{I,j_5}, E^m_{O,j_5})$. If the computation returns 0 for a certain m, the attacker disregards the pair of plaintexts $P^m, P^{*,m}$.

Now, according to (15), $P[E^m_3 = L^m_3] = 1/16$; by (21), if this is the case, then $filters^m = 1$. From (22), we also conclude that $P[filters^m = 1] = 3/8$. Thus, the attacker discards 5/8 of the pairs of plaintexts; of the remaining 3/8, 1/6 of them correspond to pairs $P^m, P^{*,m}$ such that $L^m_3 = E^m_3$. These are the plaintexts which the attacker will use to gain information about the secret key.

From (16) and the previous reasoning, we conclude that

$$\begin{aligned} P[&OS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^1) \oplus OS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^1) = E^m_{O,j}\\ &\mid filters^m = 1] = \frac{1}{6}. \end{aligned} \tag{23}$$

(17) and (18) tell us that

$$E^m_{I,j} = IS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^1),$$

$$E^{*,m}_{I,j} = IS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^1).$$

The attacker then computes $contains(test_j(E^m_{I,j}, E^{*,m}_{I,j}, E^m_{O,j}), k)$ for all $j = 2, 5, 6, 7, 8$ and all $k \in \mathbb{Z}_2^6$.

Now, if

$$E^m_{O,j} = OS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^1) \oplus OS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^1), \tag{24}$$

which happens with probability 1/6, we have

$$\begin{aligned} &contains^m_j(k) \equiv contains(test_j(E^m_{I,j}, E^{*,m}_{I,j}, E^m_{O,j}), k) =\\ &contains(test_j(IS_{j,k}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^1) \end{aligned}$$

[10] This property is an estimation rather than a precise value; using a more complete set of cryptographic properties, it is obtained in the distribution of maximum entropy from other properties of the encryption process.

and

$$IS_{j,k}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^1), b_O), k).$$

Thus, by (19), $contains_j^m(K_{6,j}^1) = 1$. If $k \neq K_{6,j}^1$ is some other bitstring, $P[contains_j^m(k) = 1] = 1/16$ and $P[contains_j^m(k) = 0] = 15/16$. If (24) is not verified, which happens with probability 5/6, we have $P[contains_j^m(k) = 1] = 1/16$ with probability 1/16 for all $k \in \mathbb{Z}_2^6$.

It is easy to see that in the distribution of maximum entropy:

- the results of $contains_j^m(k)$ and $contains_{j'}^m(k)$ are independent for $j \neq j'$;
- the results of $contains_j^m(k)$ and $contains_j^m(k')$ are independent for $k \neq k'$;
- the results of $contains_j^m(k)$ and $contains_j^{m'}(k)$ are independent for $m \neq m'$.

The attacker will try to guess the bits $K_{6,j}^1$ for $j = 2, 5, 6, 7, 8$, thus obtaining 30 of the 56 bits of the original key. Suppose that the attacker encrypts $N = 200$ pairs of plaintexts. We may estimate his probability of success as follows. The expected number of "filtered" pairs if $3N/8 = 75$. For each $j = 2, 5, 6, 7, 8$ and each filtered pair of plaintexts, we have $P[contains_j^m(k) = 1] = 1/6 + 5/6 \cdot 1/16 = 7/32$ if $k = K_{6,j}^1$ and $P[contains_j^m(k) = 1] = 1/16$ if $k \neq K_{6,j}^1$.

Fix $j \in \{2, 5, 6, 7, 8\}$, $K_{6,j}^1 \neq k \in \mathbb{Z}_2^6$, and recall that in the distribution of maximum entropy the results of $contains_j^m$ are independent for different values of m. Taking this fact and the previous estimations into account, we conclude that the probability that after 75 filtered tests there are more positive results $contains_j^m(k)$ than $contains_j^m(K_{6,j}^1)$ is given by

$$p = \sum_{i=1}^{75} \left(\binom{75}{i} \left(\frac{7}{32}\right)^i \left(\frac{1}{16}\right)^{75-i} \sum_{j=i+1}^{75} \binom{75}{j} \left(\frac{1}{16}\right)^j \left(\frac{15}{16}\right)^{75-j} \right) \approx 0.0013. \tag{25}$$

Thus, the probability that some bitstring $K_{6,j}^1 \neq k \in \mathbb{Z}_2^6$ has more positive results than $K_{6,j}^1$ can be estimated to be less than $63p \approx 0.082$. As such, the probability that, for all $j = 2, 5, 6, 7, 8$, the bitstring with more positive results $contains_j^m(k)$ is the one represented by $K_{6,j}^1$ is at least $(1 - 63p)^5 \approx 0.65$. □

Theorem 3.8 *Let $E \in \mathbf{Exp}$ be such that $\mathcal{A}(E_\eta)$ (the set of bitstrings which E_η may represent for security parameter η) verifies the following:*

- *given a bitstring b and a security parameter η, there are efficient algorithms for deciding whether or not $b \in \mathcal{A}(E_\eta)$;*
- *there are $N \in \mathbb{N}, k > 1$ satisfying $\eta > N \Rightarrow |\mathcal{A}(E_\eta)| > k^\eta$ (i.e., $\mathcal{A}(E_\eta)$ grows exponentially in η)* [11] .

Suppose that there is a polynomial-time attacker A whose probability of guessing the bitstring represented by E_η is a non-negligible, computable function of η. Then, there exists an attacker $\tilde{A}$ and an expression $g \in \mathbf{Exp}^{\tilde{A}}$ representing $\tilde{A}$'s guess such that

$$H_{max}(E_\eta) - H(E \mid g) \tag{26}$$

is non-negligible as a function of η, where $E, g \leftarrow \mathcal{D}$ and $\mathcal{D}$ is the distribution of maximum entropy on $\Omega_\eta^{\tilde{A}}$ which verifies the all statements in $\mathbf{S}_\eta^{\tilde{A}}$ (i.e., the set of cryptanalytical properties known by $\tilde{A}$) and $H_{max}(E_\eta) = \log|\mathcal{A}(E_\eta)|$ is the maximum entropy of a probability distribution on $\mathcal{A}(E_\eta)$.

Proof. The first step of the proof is to see that we may specify an attacker $\tilde{A}$ with the two properties described above.

For this, suppose that, whenever the attacker A would make a probabilistic decision, $\tilde{A}$ runs some random generation algorithm $R \in \mathcal{R}^{\tilde{A}}$ and records its result. He then makes his choice deterministically, as determined by the output of the random generation algorithm. Clearly, we may assume that the probability of each decision is the same for both attackers, A and $\tilde{A}$, and so their probability of success is also the same.

Since all of $\tilde{A}$'s probabilistic decisions and computations are represented by the results of random generation algorithms in $\mathcal{R}^{\tilde{A}}$, it is clear that we may write $\tilde{A}$'s guess as an expression $g \in \mathbf{Exp}^{\tilde{A}}$ which depends on the outputs of random generation algorithms executed during the protocol (either by the attacker or other users of the network). Since $\tilde{A}$ is a polynomial-time adversary, each random generation algorithm

[11] Intuitively, this last condition demands that $\mathcal{A}(E_\eta)$ is so large that guessing the bistring corresponding to E_η is hard.

cannot be executed more than a polynomial number of times, and it is easy to see that g can be computed in polynomial-time by simulating an execution of the attack (recall that all probabilistic decisions and randomly generated data are determined by the random generation algorithms).

Let s be the non-negligible, computable function such that $s(\eta)$ is the probability that the attacker correctly guesses the bitstring corresponding to E_η. $\tilde{A}$'s cryptanalytical knowledge contains the properties

$$P[(Equals(E,g))_\eta = 1] = s(\eta)$$

for each fixed security parameter η, where $Equals \in \mathcal{F}^{\tilde{A}}$ is an algorithm which receives two bitstrings and returns 1 if they are equal, 0 otherwise. This property states that the probability that the guess of the attacker is correct is given by $s(\eta)$, and thus is correct by hypothesis. Furthermore, we let the attacker know that the value of E is in $\mathcal{A}(E_\eta)$: *i.e.*, we define $e \in \mathcal{F}^{\tilde{A}}$ so that $e(\eta, b) = 1$ if $b \in \mathcal{A}(\eta)$ and 0 otherwise, and consider the properties

$$P[E = b \mid e(b) = 0] = 0,$$
$$P[e(E) = e(b) \mid E = b] = 1$$

for all $E \in \mathbf{Exp}_\eta^{\tilde{A}}$. We also need some properties of *Equals* to prove our result:

$$P[E' = b \mid E = b, Equals(E, E') = 1] = 1,$$

$$P[E' = b \mid E = b, Equals(E, E') = 0] = 0,$$

where $E, E' \in \mathbf{Exp}_\eta^{\tilde{A}}$, $b \in \mathbf{B}_\eta^{\tilde{A}}$.

These properties imply that

$$P[E = b \mid g = b] = s(\eta),$$
$$P[E = b \mid e(b) = 0] = 0$$

when E is sampled from the distribution of maximum entropy given $\tilde{A}$'s cryptanalytical knowledge; the second property is valid for all $b \in \mathbf{B}_\eta^{\tilde{A}}$.

Without any further knowledge of E, it follows that, in the distribution of maximum entropy,

$$P[E = b \mid Equals(g, b) = 0, e(b) = 1] = \frac{1 - s(\eta)}{|\mathcal{A}(E_\eta)| - 1}$$

for $b \in \mathbf{B}_\eta^{\tilde{A}}$.

Thus,

$$H(E \mid g) = -s(\eta)\log(s(\eta)) - (1 - s(\eta))\log\frac{1 - s(\eta)}{|\mathcal{A}(E_\eta)| - 1}$$
$$< -\frac{1}{\eta^c}\log\frac{1}{\eta^c} - (1 - \frac{1}{\eta^c})\log\frac{1 - \frac{1}{\eta^c}}{|\mathcal{A}(E_\eta)| - 1},$$

for sufficiently large η and some $c > 0$. By definition of H_{max},

$$H_{max}(E_\eta) = \log(|\mathcal{A}(E_\eta)|).$$

Now

$$\begin{aligned} H_{max}(E_\eta) - H(E \mid g) &\geq \log(|\mathcal{A}(E_\eta)|) + \frac{1}{\eta^c}\log\frac{1}{\eta^c} + (1 - \frac{1}{\eta^c})\log\frac{1 - \frac{1}{\eta^c}}{|\mathcal{A}(E_\eta)| - 1} \\ &= \log(|\mathcal{A}(E_\eta)|) + \frac{1}{\eta^c}\log\frac{1}{\eta^c} \\ &+ (1 - \frac{1}{\eta^c})\log(1 - \frac{1}{\eta^c}) - (1 - \frac{1}{\eta^c})\log(|\mathcal{A}(E_\eta)| - 1) \\ &= [\log(|\mathcal{A}(E_\eta)|) - \log(|\mathcal{A}(E_\eta)| - 1)] + \frac{1}{\eta^c}\log\frac{1}{\eta^c} \\ &+ (1 - \frac{1}{\eta^c})\log(1 - \frac{1}{\eta^c}) + \frac{1}{\eta^c}\log(|\mathcal{A}(E_\eta)| - 1). \end{aligned} \tag{27}$$

Dividing each of these parcels by η^{c-1} and using properties of logarithms, one obtains that

$$\frac{H_{max}(E_\eta) - H(E_\eta \mid g_\eta)}{\eta^{c-1}} > C'$$

for some constant C' and all sufficiently large η, which concludes the proof. □

Example 3.9 (Probability of Success) It is clear that the probability distribution of $c(\eta)$ is uniform in the set $\{1,\ldots,2\eta\}$ (considering the distribution of maximu entropy), and that different executions of c are independent (*i.e.*, c^m, $c^{m'}$ are independent for $m \neq m'$). It is also easy to see that the responses $R(C, C^k)$ are independent for different values of k. From the first consideration one may conclude that $P[c^m = c^{m'}] = 1/(2\eta)$ for $m \neq m'$. Equations (4), (5) imply that $R(C, C^k) = 0$ iff $c^j \neq c^{j+k\eta}$ for $j = 1,\ldots,\eta$. Thus, we obtain $P[R(C, C^k) = 0] = (\frac{2\eta-1}{2\eta})^\eta \to \frac{1}{\sqrt{e}}$.

After $4\eta\log\eta$ queries, we expect that approximately $N = \frac{4}{\sqrt{e}}\eta\log\eta$ of them, say $k_1,\ldots,k_N$, are "relevant" (*i.e.*, A's response, $R(C, C^{k_i})$, is 0). Let us then estimate the probability that I can obtain the correct bitstring from this number of relevant queries. Consider the set S_1 of the bitstrings represented by $c^{1+k_1\eta}$, ..., $c^{1+k_N\eta}$. If $\#S_1 = 2\eta - 1$, then the information available to the attacker allows to exclude all but one hypothesis for the value of c^1 (by equation (4)). The probability of this event is at least

$$\frac{(2\eta)^{B\eta\log\eta} - (2\eta-1)^{B\eta\log\eta}}{(2\eta)^{B\eta\log\eta}}$$

$$= 1 - (1 - \frac{1/2}{\eta})^{B\eta\log\eta} \to 1 - \frac{1}{\eta^{2/\sqrt{e}}},$$

where $B = \frac{4}{\sqrt{e}}$.

Since all executions of c are independent, it is easy to conclude that the same calculation can be made for all other "slots" 1 through η. Thus, the probability that the attacker is able to obtain the correct "color" for all slots can be estimated to be

$$(1 - \frac{1}{\eta^{2/\sqrt{e}}})^\eta > (1 - \frac{1}{\eta})^\eta \to \frac{1}{e}.$$

□