Modeling and Reasoning about an Attacker with Cryptanalytical Capabilities

Bruno Montalto[1,2] 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 their (partial) knowledge of the properties of cryptographic primitives. 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 cryptographic protocol given the prescribed knowledge of the attacker. Although we prove that this quantity is typically NP-hard to compute, we still manage to show its usefulness in analyzing a few meaningful 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 uncovering a secret with non-negligible probability.

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

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 [15], 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. One of the clearest evidences of the usefulness of such methods is the *man-in-the-middle* attack to the Needham-Schroeder protocol, discovered by Lowe in 1995 [25] using the Dolev-Yao model.

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

[2] Email: bmontal@math.ist.utl.pt

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

Several tools for automated analysis of security protocols (mostly attack-search engines) have been developed based on this approach and are currently being used in industrial scale projects [6] [14] [31]. 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. In fact, realistic cryptographic primitives have properties which the attacker may explore and attack: for an example, encryption functions may reveal the length of the original message.

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 [24] [7] [8] [13]. 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 analyse even simple protocols in this setting.

In recent years, there has been a considerable effort in bridging the gap between these two approaches [2] [8] [20] [30]. 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 symbolic representation of a Dolev-Yao adversary, such as [5]. 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 [27]. 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 from them. Thus, by describing general cryptographic properties, we may use our model to assess the impact of those properties 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. This translation allows us to obtain an equivalent but simpler statement of IND-CPA security. Section 3 concerns the computation of probabilities based on the set of probabilistic statements available to the attacker. Our goal is to find a way of estimating the probability distribution of the secret and randomly generated messages that satisfies all the statements known by the attacker, but does not introduce any additional structure and remains as hard to attack as possible. To achieve this goal we will 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, when that is the case, 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.

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 is a finite sequence of bitstrings. We represent the set of finite sequences of bitstrings by $\mathbf{B} = (\{0,1\}^*)^*$. For simplicity, we will often refer to elements of $\mathbf{B}$ simply as bitstrings. 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. 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 probabilistic 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 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[4].

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

- for each $j \in \mathbb{N}$ and $R \in \mathcal{S}$, $R^j \in \mathbf{Exp}(\mathcal{G}, \mathcal{S})$;
- $\mathbf{B} \subseteq \mathbf{Exp}(\mathcal{G}, \mathcal{S})$;
- if $E_1, \ldots, E_n \in \mathbf{Exp}(\mathcal{G}, \mathcal{S})$ and $F \in \mathcal{G}$, $F(E_1, \ldots, E_n) \in \mathbf{Exp}(\mathcal{G}, \mathcal{S})$.

[4] 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 the sets $\mathcal{R}, \mathcal{R}^A$ receive only the security parameter as input. We will refer to such algorithms as random generation algorithms.

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 only public algorithms (resp. all 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. Note also that in the previous definition we use the same symbol F in two different contexts: as part of the representation of an expression $E \in \mathbf{Exp}$ and to represent the function calculated by the algorithm F. We use this notation throughout this article; the context should avoid ambiguities.

Before introducing our way of defining cryptographic properties we need to fix some notation. Observe that, for each security parameter η, an attacker with limited computational power (*i.e.*, an attacker who, for a given security parameter η, may only perform a$f(\eta)$ of operations) can only interpret and obtain information from expressions with at most $f(\eta)$ subexpressions. Thus, when modeling such an adversary, the set of expressions which needs to be considered is finite. Let us denote that set by $\mathbf{Exp}^A_\eta \subseteq \mathbf{Exp}^A$. 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 $i < f(\eta)$. 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

$$\Omega^A_\eta = \prod_{E \in \mathbf{Exp}^A_\eta \setminus \mathbf{B}} \mathbf{B}^A_\eta.$$

We use the superindex A to highlight the fact that this set depends on the algorithms and bitstrings available to the attacker A. Intuitively, an element of Ω^A_η associates each expression in $\mathbf{Exp}^A_\eta$ to a bitstring in $\mathbf{B}^A_\eta$. Thus, if $\mathbf{Exp}^A_\eta = \{E^1, \ldots, E^N\}$, an element of Ω^A_η is an N-uple of bitstrings, each corresponding to an expression. Since an element of Ω^A_η completely determines the bitstrings corresponding to each expression, a probability distribution on Ω^A_η completely determines the probability distribution of each random variable represented by an expression, as well as possible dependencies between them. We will say that $\mathcal{D}_\eta$ is the probability distribution induced on Ω^A_η by the algorithms in $\mathcal{F}^A$ and $\mathcal{R}^A$. If $\mathcal{D}'$ is any probability distribution on Ω^A_η, we will write $(E^1_\eta, \ldots, E^N_\eta) \leftarrow \mathcal{D}'$ to denote the sampling of an element of Ω^A_η from $\mathcal{D}'_\eta$. Note that the expressions E^i_η may be interpreted as random variables. In particular, if $\mathcal{D}' = \mathcal{D}$, then each E^i_η may be interpreted as a random variable representing the bitstring obtained by constructing a message in the way prescribed by the expression E^i using security parameter η in the calculations. Similarly, we write $E^1_\eta, \ldots, E^n_\eta \leftarrow \mathcal{D}_\eta$ when $E^1, \ldots, E^n \subseteq \mathbf{Exp}^A_\eta$. If $x \in \Omega^A_\eta$ and $E \in \mathbf{Exp}^A_\eta$, we will also write x_E to denote the bitstring associated to the expression E by x. 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 way of defining cryptographic properties.

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

$$P[E^1_\eta = b_1, \ldots, E^n_\eta = b_n \mid E^{*,1}_\eta = b^*_1, \ldots, E^{*,n^*}_\eta = b^*_{n^*}] = \rho,$$

where $\eta \in \mathbb{N}$ is a security parameter, $E^1, \ldots, E^n$, $E^{*,1}, \ldots, E^{*,n^*} \in \mathbf{Exp}^A$, $b_1, \ldots, b_n$, $b^*_1, \ldots, b^*_{n^*} \in \mathbf{B}^A$, and $\rho \in [0,1]$.

Thus, a cryptographic property can be written as a probabilistic relation 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 $\eta \in \mathbb{N}$, $E^1, \ldots, E^n$, $E^{*,1}, \ldots, E^{*,n^*} \in \mathbf{Exp}^A_\eta$, $b_1, \ldots, b_n$, $b^*_1, \ldots, b^*_{n^*} \in \mathbf{B}^A_\eta$ and $\rho \in [0,1]$, we will say that

$$P[E^1_\eta = b_1, \ldots, E^n_\eta = b_n \mid E^{*,1}_\eta = b^*_1, \ldots, E^{*,n^*}_\eta = b^*_{n^*}] = \rho,$$

is verified by a probability distribution $\mathcal{D}'$ in Ω^A_η if the equality holds when $(E^1, \ldots, E^n, E^{*,1}, \ldots, E^{*,n^*}) \leftarrow \mathcal{D}'$.

Note that not all probabilistic statements denote cryptographic weaknesses: indeed, certain cryptographic properties are necessary for the cryptographic primitives to work properly. For example, consider a non-deterministic 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 $pub(R^i_k)$ (resp. $priv(R^i_k)$) returns the public (resp. private) key corresponding to R_k;
- $R_e \in \mathcal{R}$ is a random generation algorithm representing the randomness involved in the encryption;
- Enc, $Dec \in \mathcal{F}$ are the encryption and decryption algorithms, respectively.

In this case, the encryption of a message represented by the expression E using a public key $pub(R^i_k)$ is represented by the expression $Enc(E, pub(R_k), R_e)$, and the decryption of E using the private key $priv(R^i_k)$ is represented by $Dec(E, priv(R_k))$. Thus, the following cryptographic properties are always verified.

$$P[(Dec(Enc(E, pub(R^i_k), R^j_e), priv(R^i_k)))_\eta = b \mid E_\eta = b] = 1,$$

$$P[E_\eta = b \mid (Dec(Enc(E, pub(R^i_k), R^j_e), priv(R^i_k)))_\eta = b] = 1.$$

Example 2.3 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.

$$P[mult(E^1, Enc(E^2, pub(R^i_k), R^j_e)) = b \mid Enc(mult(E^1, E^2), pub(R^i_k), R^j_e) = b] = 1,$$

$$P[Enc(mult(E^1, E^2), pub(R^i_k), R^j_e) = b \mid mult(E^1, Enc(E^2, pub(R^i_k), R^j_e)) = b] = 1.$$

Example 2.4 Many encryption functions reveal some information about the length of the underlying plaintext. For example, suppose that an encryption scheme is such that

"There is a 0.5 probability that the length of a ciphertext is the same as the length of the underlying plaintext."

This may be represented by the cryptographic property

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

where $\mathit{length} \in \mathcal{F}^A$ gives the length of a given message and l represents an integer.

We now represent in our setting the notion of IND-CPA (indistinguishability under chosen plaintext attacks) security, and use the representation to show that the traditional definition of this concept is equivalent to a simpler one. 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 $k > n \Rightarrow f(k) \leq k^{-c}$.

Definition 2.5 Suppose that $\mathcal{E} = (R_k, \mathit{pub}, \mathit{priv}, R_e, \mathit{Enc}, \mathit{Dec})$ is an asymmetric encryption scheme. Consider the following experiment:

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

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

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

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

$$P[A(\eta, \mathit{pub}(R_k^1)) = 1 \mid b = 1] - P[A(\eta, \mathit{pub}(R_k^1)) = 1 \mid b = 0] \tag{1}$$

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

The 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 in polynomial time. The intuition behind IND-CPA security is that, for a polynomial-time observer, the encryption function reveals nothing about the underlying plaintext: looking at an encryption and two distinct plaintexts, it is impossible to say that one of the plaintexts is significantly more likely to be the correct decryption than the other. The next theorem describes this intuition in a rigorous (if a little involved) way, and also presents a simpler definition of IND-CPA security.

Theorem 2.6 *Suppose that $\mathcal{E} = (R_k, pub, priv, R_e, Enc, Dec)$ is an asymmetric encryption scheme. Then 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

$$\begin{aligned} &P[B(\eta, Q^0, Q^1, \mathcal{O}(Q^0, Q^1), R^1) = 1 \mid b = 1] \\ -&P[B(\eta, Q^0, Q^1, \mathcal{O}(Q^0, Q^1), R^1) = 1 \mid b = 0] \end{aligned} \tag{2}$$

is not negligible as a function of η, where $Q^0 \equiv (Q(\eta, k, R^1))^0$ and $Q^1 \equiv (Q(\eta, k, R^1))^1$, $R^1 \leftarrow R(\eta)$ and k is the public key used in the experiment.

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

The proof is done in the Appendix.

3 Probabilistic reasoning

In this section we discuss how to estimate a probability distribution of the random variables represented by expressions based on the probabilistic statements known by the attacker. This is useful for estimating the probability of success of an attack strategy (by using the attacker's knowledge to describe known properties of cryptographic primitives used in the protocol). Furthermore, note that we may express general abstract cryptographic properties. As such, estimating this probability distribution for given cryptographic properties (instead of using cryptographic properties verified by predetermined cryptographic primitives) allows us to study whether or not those properties compromise the security of a given protocol. We consider an attacker with access to a function $p\colon \mathbb{N} \times ((\mathbf{E} \times \mathbf{B})^*)^2 \to [0,1] \cup \{\perp\}$, which he uses to compute probabilities. We also require that p either returns a correct probability or $\perp$. For simplicity, we will write

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

instead of

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

We want to find a way of determining a reasonable estimation of values of p given the knowledge provided to the attacker. Exactly what is a reasonable estimation is difficult to define, and such problems have been subject of extensive research, namely on inductive logic [21]. In our approach we use the cryptographic properties obtained by the function p to build an estimated distribution on the set Ω_η^A using Shannon's notion of entropy [29]. Shannon's entropy is widely used in information theory as the standard measure for uncertainty about the outcome of a random experiment. It also has good properties when measuring the uncertainty of a random variable X in presence of information about another random variable Y, whose outcome may or may not be related to the outcome of X. 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 $\mathbf{Exp}^A_\eta$, $\mathbf{B}^A_\eta$ and the function p, we say that $\mathcal{D}^H_\eta$ is a *distribution of maximum entropy* in Ω^A_η for the function p if the following conditions are verified:

- if $(E^1, \ldots, E^n, E^{*,1}, \ldots, E^{*,n^*}) \leftarrow \mathcal{D}^H_\eta$ and
$$p_\eta[E^1 = b_1, \ldots, E^n = b_n \mid E^{*,1} = b^*_1, \ldots, E^{*,n^*} = b^*_{n^*}] = \rho \neq \perp,$$
then
$$P[E^1 = b_1, \ldots, E^n = b_n \mid E^{*,1} = b^*_1, \ldots, E^{*,n^*} = b^*_{n^*}] = \rho.$$
- if $\mathcal{D}^*$ is another distribution which respects the first property, and X, X^* are random variables over Ω_η with distribution $\mathcal{D}^H_\eta, \mathcal{D}^*$, then $H(X) \geq H(X^*)$.

Henceforth we will describe the properties of cryptographic primitives known by an attacker directly as probabilistic statements, without resorting to the function p. The following example illustrates the behavior of such a distribution given a few simple probabilistic statements about Bernoulli random variables.

Example 3.2 Let us 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 $\mathbf{R}$ and analyse the resulting changes in the distribution of maximum entropy $\mathcal{D}^H(\mathbf{R})$ when we require that it verifies the statements in $\mathbf{R}$. We will use e as the base of the logarithms for the calculus of entropy; the values presented are approximate.

	$\mathcal{D}^H(\mathbf{R})$				
$\mathbf{R}$	$P[A,B]$	$P[A,\bar{B}]$	$P[\bar{A},B]$	$P[\bar{A},\bar{B}]$	$H((A,B,C))$
$\{\ \}$	0.25	0.25	0.25	0.25	1.386
$\{P[A] = 0.6\}$	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

Let us denote the (finite) set of cryptographic properties known by the attacker A for security parameter η by $\mathbf{R}^A_\eta$. If all outputs of the function p are correct (*i.e.*, $\mathcal{D}_\eta$ verifies all cryptographic properties computed by p), then there is at least one distribution (the "real" one) which verifies all properties in $\mathbf{R}^A_\eta$. If the set of distributions which verify all properties in $\mathbf{R}^A_\eta$ is non-empty, then the problem of finding the distribution of maximum entropy is essentially that of maximizing

$$\sum_{(E,b)\in\mathbf{Exp}_\eta\times\mathbf{B}^A_\eta} x(E,b)\log x(E,b)$$

restricted to the set

$$\{x = \prod_{(E,b)\in\mathbf{Exp}_\eta\times\mathbf{B}^A_\eta} x(E,b) \colon \sum_{(E,b)\in\mathbf{Exp}_\eta\times\mathbf{B}^A_\eta} x(E,b) = 1, x(E,b) \geq 0 \text{ for all } E, b\}.$$

This is a continuous function in a non-empty, compact set. Thus, there is always a maximum, and at least one distribution of maximum entropy exists. We will now present an important propertiy of $\mathcal{D}^H_\eta$ which help to justify why we choose it as a reasonable estimation and also provides a slightly simpler way of computing it. The following definition will be useful.

Definition 3.3 Let $r \in \mathbf{R}^A_\eta$ be the probabilistic statement

$$P[E^1 = b_1, \ldots, E^n = b_n \mid E^{*,1} = b^*_1, \ldots, E^{*,n^*} = b^*_{n^*}] = \rho,$$

and let

$$\Omega_\eta \supseteq A = \{x \in \Omega_\eta \colon x_{E^1} = b_1, \ldots, x_{E^N} = b_N\},$$
$$\Omega_\eta \supseteq A^* = \{x \in \Omega_\eta \colon x_{E^{*,1}} = b^*_1, \ldots, x_{E^{*,n^*}} = b^*_{n^*}\}.$$

We will say that the set $\{P^1_r, P^2_r, P^3_r\}$ is *the partition (of Ω_η) induced by* r, where $P^1_r = A^* \setminus A$, $P^2_r = A \cap A^*$, and $P^3_r = \Omega_\eta \setminus A^*$. Observe that $\bigcup_{i=1}^3 P^i_r = \Omega_\eta$ and $P^i_r \cap P^j_r = \emptyset$ for any $i \neq j$.

If $\mathbf{R}^A_\eta = \{r_1, \ldots, r_n\}$ is a finite set of cryptographic properties, we say that

$$\bigcup_{i_1=1}^{3} \cdots \bigcup_{i_n=1}^{3} \{P^{i_1}_{r_1} \cap \ldots \cap P^{i_n}_{r_n}\}$$

is *the partition (of Ω_η) induced by* $\mathbf{R}^A_\eta$.

The next theorem illustrates the importance of this definition. For each $r \in \mathbf{R}^A_\eta$ and each $x \in \Omega_\eta$, we write $i(r,x)$ for the number $i \in \{1,2,3\}$ such that $x \in P^{i(r,x)}_r$.

Theorem 3.4 *Fix a security parameter η. Let $\mathbf{R}^A_\eta = \{r_1, \ldots, r_N\}$ be the set of probabilistic statements known by the attacker for security parameter η, and consider the partition of Ω_η induced by $\mathbf{R}^A_\eta$. Suppose that there is a distribution of maximum entropy $\mathcal{D}^H_\eta$ for $\mathbf{R}^A_\eta$, and fix $(i_1, \ldots, i_N) \in \{1,2,3\}^N$. Then, for $X \leftarrow \mathcal{D}^H$, we have*

$$x, y \in P^{i_1}_{r_1} \cap \ldots \cap P^{i_N}_{r_N} \Rightarrow P[X = x] = P[X = y]. \qquad (3)$$

This theorem, whose proof can be found in the appendix, shows that the estimation $\mathcal{D}^H$ gives the same probability to elements of Ω_η which the knowledge of the attacker does not distinguish. This is a desirable property, since it indicates that in some sense $\mathcal{D}^H$ 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 $\mathcal{D}^H_\eta$.

Corollary 3.5 *Consider the set of cryptographic properties* $\mathbf{R}^A_\eta = \{r_1, \ldots, r_N\}$*. Suppose that* $p = (p_k)_{k \in \{1,2,3\}^N}$ *is a maximum of*

$$-\sum_{i_1=1}^{3} \cdots \sum_{i_N=1}^{3} |P^{i_1}_{r_1} \cap \ldots \cap P^{i_N}_{r_N}| \cdot p_{(i_1,\ldots,i_N)} \log p_{(i_1,\ldots,i_N)}$$

restricted to

$$\begin{cases} \sum_{i_1=1}^{3} \cdots \sum_{i_N}^{3} |P^{i_1}_{r_1} \cap \ldots \cap P^{i_N}_{r_N}| \cdot p_{(i_1,\ldots,i_N)} = 1 \\ p_{(i_1,\ldots,i_N)} \geq 0. \end{cases}.$$

Then the distribution $\mathcal{D}$ *defined by* $X \leftarrow \mathcal{D} \Rightarrow P[X = x] = p_{(i_1,\ldots,i_N)}$*, where each* i_j *is such that* $x \in P^{i_j}_{r_j}$*, is a distribution of maximum entropy for* $\mathbf{R}^A_\eta$*.*

The problem of maximizing a function in a set bounded by linear conditions is well studied in linear programming, and there are efficient algorithms for solving it [22]. 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 is still not 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 rules in $\mathbf{R}$). We use the fact that the problem **3SAT** is NP-complete [23].

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

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

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

The problem **3SAT** *reduces to* **Prob***. In particular,* **Prob** *is NP-hard (in* $|\mathbf{R}|$*).*

This theorem states that estimating this probability is NP-hard in the number of cryptographic properties known by the attacker[5]. 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. This is a chosen-plaintext attack: the attacker attempts to obtain partial information about the secret key being used by encrypting several pairs of carefully chosen plaintexts. 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.

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

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 [32].

For the next theorem we consider a polynomial-time attacker 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 the attacker to have a non-negligible probability of success. The result concerns the (asymptotic) entropy of E, given the attacker's strategy, in the distribution of maximum entropy given the cryptographic properties with which we equip the attacker. We first present a description of the result. Suppose that A is able to use the knowledge obtained about messages exchanged in the network, together with its own cryptanalytical capabilities, to correctly uncover the bitstring corresponding to the expression E with a non-negligible probability. We assume two things about the expression E:

- there are efficient algorithms for deciding whether or not a given bitstring is an element of $\mathcal{A}(E_\eta)$ (the set of bitstrings which E_η may represent): *i.e.*, there is a (deterministic) polynomial-time algorithm e such that, for $\eta \in \mathbb{N}$, $b \in \mathbf{B}$,

$$e(\eta, b) = \begin{cases} 1 \text{ if } b \in \mathcal{A}(E_\eta) \\ 0 \text{ otherwise.} \end{cases} \tag{4}$$

- $|\mathcal{A}(E_\eta)|$ grows exponentially as a function of η: more precisely, there are $N \in \mathbb{N}, k > 1$ satisfying

$$\eta > N \Rightarrow |\mathcal{A}(E_\eta)| > k^\eta. \tag{5}$$

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

Theorem 3.8 *Let $E \in \mathbf{Exp}$ be such that there is a polynomial-time algorithm e and $N \in \mathbb{N}, k > 1$ satisfying conditions (4) and (5). 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*

$$H_{max}(E_\eta) - H(E \mid g) \tag{6}$$

is non-negligible as a function of η, where $E, g \leftarrow \mathcal{D}^H_\eta$ and $\mathcal{D}^H_\eta$ is the distribution of *maximum entropy on $\Omega_\eta^{\tilde{A}}$ which verifies the cryptanalytical properties known by $\tilde{A}$.*

This theorem may be useful in proving that an attacker, given a set of known cryptographic properties, cannot compromise the security of a protocol. In fact, suppose that (6) is negligible for an algorithm g representing some attack strategy. If that attacker has a non-negligible probability of success, there must be some other relevant, polynomial-time computable property which gives non-negligible information about E_η from the attacker's knowledge. If given the knowledge of the attacker there is no uncovering algorithm g such that (6) is non-negligible, then the cryptographic properties which we allow the attacker to explore in his attack strategy do not compromise the security of the protocol by themselves. Of course, when we consider specific cryptographic primitives, they may verify other properties which (together with the properties described in the setting) 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 $\{1, \ldots, 2\eta\}$[6]. For security parameter η, a player A creates a code $C \equiv (c^1 \ldots c^\eta)$ by executing the random generation algorithm η times. Another player I tries to guess C by querying A with codes $C^k \equiv (c^{1+k\eta} \ldots c^{\eta+k\eta})$. A returns $R(C, C^k)$, given by:

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

$R(C, C^k)$ (the number of red balls) represents the number of slots in which C^k has the same color as the correct code C. We interpret I (from intruder) as the attacker, trying to uncover a secret message.

The following cryptographic properties will be used:

$$P[c^j = n \mid c^{j+k\eta} = n, R(C, C^k) = 0] = 0, j = 1, \ldots, \eta, \tag{7}$$

$$P[R(C, C^k) = 0 \mid c^1 = b_1, c^{1+k\eta} = b_1', \ldots, c^\eta = b_\eta, c^{\eta+k\eta} = b_\eta'] = 1, \tag{8}$$

where $b_1 \neq b_1'$, ..., $b_\eta \neq b_\eta'$.

I queries A with randomly generated codes C^k. If $R(C, C^k) \neq 0$, he discards the information obtained. Otherwise, he concludes from the equation above that $c^j \neq c^{k\eta+j}$. Suppose that I 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 6) the probability of success to be $\frac{1}{e}$, 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 (7), (8) we equip the attacker with properties

$$P[c^j = n \mid c^{j+k\eta} = n, R(C, C^k) = \eta] = 1, j = 1, \ldots, \eta. \tag{9}$$

[6] Recall that there are bijections between $\mathbb{Z}$ and $\mathbf{B}$ which may be efficiently calculated and inverted.

The attacker knows that if $R(C, C^k) = \eta$ then $C = C^k$; however, in the distribution of maximum entropy, this does not give the attacker any knowledge about the secret code if $R(C, C^k) \neq \eta$.

With a polynomial number of queries, the attacker can only obtain the bitstrings represented by $R(C, C^k)$ for a polynomial number of different k's.

Consider the following (natural) attack strategy. The attacker performs a polynomial number of queries, η^C, 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 $\frac{\eta^C}{(2\eta)^\eta}$ 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 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 (9) 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 find an attack, but if one does not it behaves essentially as a symbolic model. For illustrations purposes, we showed how IND-CPA security may be expressed in terms of the probabilistic statements considered, and concluded that its usual definition may be replaced by a simpler one. 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 based on the set of known cryptographic properties. We presented a few properties of this distribution which make it a reasonable approach to model an attacker with cryptanalytical knowledge. This technique may also be used to study whether a weakness of the cryptographic primitives used compromises or not the security of a given protocol. Another advantage of this is that it allows, to some extent, a separate study of weaknesses of cryptographic primitives and the security of protocols. In fact, the model verifies the security of a protocol against an attacker who may use certain cryptographic weaknesses. Thus, one may simply write all properties of the cryptographic primitives deemed relevant and verify if the protocol is still secure when the attacker explores them.

Several problems and open questions still remain. First of all, for the model to be effective, one needs to give a complete description of the cryptographic properties of the functions used, which should tipically 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. Finding criteria for deciding which properties are relevant, at least in the context of specific protocols, is also an interesting problem. Examples of attacks using properties of real cryptographic primitives are tipically hard to manage, as the complexity of the computations involved make even writing the properties a non-trivial exercise. Implementing the model should help overcome this difficulty. The task of finding an attack has exponencial complexity on the length of the attack, as is tipically the case for most tools used for this task. Other possible applications may be found in 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). 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 many advantages and is relatively natural, other methods may be more efficient or reflect better the power and knowledge of the attacker given his cryptanalytical capabilities.

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 [2]. Different proposals have dealt with this problem from different perspectives, either by including algebraic properties of the cryptographic functions in the model [3], or by incorporating guessing and probabilities [5] [26] *inter alia*, some even brought towards implementation, as in the CryptoVerif tool [9]. Of course, a thorough comparison of our approach with other proposals in the literature shall deserve close attention in the near future.

5 Bibliographical references

References

[1] Abadi, M. and A. Gordon, *A Calculus for Cryptographic Protocols: the Spi Calculus*, Proc. 4th ACM Conference on Computer and Communication Security (1997), 36–47

[2] Abadi, M. and P. Rogaway, *Reconciling Two Views of Cryptography*, Journal of Cryptology (2002), 103–127

[3] 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

[4] Adão, P., "Formal Methods for the Analysis of Security Protocols", Ph.D. thesis (2006), IST, UTL

[5] Adão, P., P. Mateus, T. Reis, L. Viganò, *Towards a quantitative analysis of security protocols*, 19th IEEE Computer Security Foundations Workshop, 2006

[6] 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

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

[8] Backles, M. and B. Pfitzmann, *Relating symbolic and cryptogrphic secrecy*, IEEE Transactions on Dependable and Secure Computing **2** (2005), 109-123

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

[10] Bond, M. and J. Clulow, *Extending security protocol analysis: new challenges*, ENTCS **125** (2005), 13–24

[11] Caleiro, C., D. Basin and L. Viganò, *Deconstructing Alice & Bob*, Workshop on Automated Reasoning for Security Protocol Analysis (2005)

[12] Cormen, T. H., C. Stein, R. L. Rivest and C. E. Leiserson, "Introduction to Algorithms", McGraw-Hill Higher Education, 2nd edition, 2001

[13] Cortier, V. and B. Warinschi, *Computationally sound, automated proofs for security protocols*, Proceedings of the 14th European Symposium on Programming, Lecture Notes in Computer Science **3444** (2005), 157-171

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

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

[16] Elgamal, T., *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory (1985), Springer-Verlag, 10–18

[17] , Halpern, J. Y. and R. Pucella, *Modeling adversaries in a logic for reasoning about security protocols*, Proceedings of the Formal Aspects of Security, First International Conference (2002), Lecture Notes in Computer Science **2629**, Springer-Verlag, Berlin/Heidelberg/New York, 115–132

[18] Halpern, J. Y. and R. Pucella, *Probabilistic algorithmic knowledge*, Logical Methods in Computer Science (2003), 118–130

[19] Halpern, J. Y., Y. Moses and M. Y. Vardi, *Algorithmic knowledge*, Proceedings 5th Conference on Theoretical Aspects of Reasoning About Knowledge (1994), 255–266

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

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

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

[23] Karp, R. M., *Reducibility Among Combinatorial Problems*, Complexity of Computer Computations, R. Miller and J. Thatcher, eds. Plenum Press, New York, 85–103

[24] 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, journal = Proceedings of the 5th ACM Conference on Computer and Communications Security (CCS), 112–121

[25] Lowe, Gavin, *An Attack on the Needham-Schroeder Public-Key Authentication Protocol*, Information Processing Letters **56** (1995), 131–133

[26] Lowe, Gavin, *Analysing protocols subject to guessing attacks*, Workshop on Issues in the Theory of Security, 2002

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

[28] Paulson, L.C., *Proving properties of security protocols by induction*, 10th IEEE Computer Security Foundations Workshop (1997), 70–83

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

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

[31] , Song, D., *Athena: a new efficient automatic checker for security protocol analysis*, Proceedings of the Twelfth IEEE Computer Security Foundations Workshop (1999), IEEE Computer Society Press, 192–202

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

[33] Warinschi, B., *A Computational Analysis of the Needham-Schröeder-Lowe Protocol*, Proceedings of the 16th IEEE Computer Security Foundations Workshop (CSFW), (2003), IEEE Computer Society Press, 248–262

6 Appendix

Proof (Theorem 2.6) Implication (3) $\Rightarrow$ (1) is trivial. (2) $\Rightarrow$ (3) is also simple.

The proof of (1) $\Rightarrow$ (2) is a little harder. In the following, we will write $\mathcal{O}_b$ to refer to the oracle $\mathcal{O}$ using bit b in the computations, so that $\mathcal{O}_b(m_0, m_1) = Enc(\eta, m_b, k, R_e(\eta))$ is an encryption of m_b with key k. Let A be an algorithm which compromises the IND-CPA security of $\mathcal{E}$. Observe that, since A is a polynomial-time algorithm, there is fixed number c such that, for security parameter η, A performs at most η^c queries to the oracle. We may assume without loss of generality that A always performs exactly η^c queries to the oracle.

We may write the first pair of plaintexts with which A queries the oracle as $(Q(\eta, 1, k, r_1))^0, (Q(\eta, 1, k, r_1))^1$, where $Q, (\cdot)^0, (\cdot)^1 \in \mathcal{F}^A$, $r_1 \leftarrow R$ and $R \in \mathcal{R}^A$ is a random generation algorithm which represents the randomness involved in the computations. More generally, for $i = 1, \ldots, \eta^c$, let

$$q_i = Q(\eta, i, k, r_1, q_1^0, q_1^1, o_1, \ldots, r_{i-1}, q_{i-1}^0, q_{i-1}^1, o_{i-1}, r_i) \tag{10}$$

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 $r_1, \ldots, r_{i-1}$ (since the output may depend on computations performed by these algorithms). But since $i < \eta^c$, this involves computing at most a polynomial number of algorithms, each of which is computable in polynomial time.

A's response can be represented using an algorithm $B_F \in \mathcal{F}^A$ as follows:

$$B_F(\eta, k, r_1, q_1^0, q_1^1, o_1, \ldots, r_{\eta^c}, q_{\eta^c}^0, q_{\eta^c}^1, o_{\eta^c}, r_{\eta^c+1}). \tag{11}$$

Since Q_i, 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 value if they do not return another answer after a certain polynomial number of operations.

Now suppose that, instead of using the same bit b 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 A is querying $\mathcal{O}_{b_i}$). Let $\delta_{i \geq j} = 1$ if $i \geq j$ and 0 otherwise. Note that

$$\begin{aligned} & P[A(\eta, k) = 1 \mid b_i = 1] - P[A(\eta, k) = 1 \mid b_i = 0] \\ = & \sum_{j=1}^{\eta^c} (P[A(\eta, k) = 1 \mid b_i = \delta_{i \geq j}] - P[A(\eta, k) = 1 \mid b_i = \delta_{i \geq j+1}]), \end{aligned} \tag{12}$$

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_i, 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 A, 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, the probability distribution of the output does not change.

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

It is easy to see that

$$P[A'(\eta, k, j) = 1 \mid b = 1] - P[A'(\eta, k, j) = 1 \mid b = 0]$$

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

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

$$P[A''(\eta, k) = 1 \mid b = 1] - P[A''(\eta, k) = 1 \mid b = 0]$$

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

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

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

Thus, by (12) and the equalities above, we conclude that

$$P[A''(\eta, k) = 1 \mid b = 1] - P[A''(\eta, k) = 1 \mid b = 0].$$

is non-negligible.

It is clear that A'' is a polynomial-time algorithm. Furthermore, A'' involves only one query to the oracle, and compromises the IND-CPA security of $\mathcal{E}$, and it is easy to see that A'' may be written in terms of algorithms $Q, (\cdot)^0, (\cdot)^1, B, R$ (R is a random generation algorithm which generates a random $j \in \{1, \ldots, \eta^c\}$ and the random bitstrings $r_1, \ldots, R_{\eta^c}$, Q generates the pairs of plaintexts with which A'' queries the oracle and B computes the final answer), which concludes the demonstration. $\square$

Proof (Theorem 3.4) We will consider an arbitrary distribution $\mathcal{D}$ which verifies all properties in $\mathbf{R}_\eta$ and obtain another distribution $\mathcal{D}'$. We will then show that properties in $\mathbf{R}_\eta$ still hold for $\mathcal{D}'$, which also verifies (3) and has greater entropy than $\mathcal{D}$.

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

$$P[X' = x] = \frac{P[X \in P_{r_1}^{i(r_1,x)} \cap \ldots \cap P_{r_N}^{i(r_N,x)}]}{|P_{r_1}^{i(r_1,x)} \cap \ldots \cap P_{r_N}^{i(r_N,x)}|}.$$

To see that $\mathcal{D}'$ verifies all properties in $\mathbf{R}_\eta$, let

$$r_j = (p_\eta[E^1 = b_1, \ldots, E^n = b_n \mid E^{*,1} = b_1^*, \ldots, E^{*,n^*} = b_{n^*}^*] = \rho) \in \mathbf{R}_\eta.$$

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

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

For each sequence $(i_1, \ldots, i_N) \in \{1,2,3\}^N$, it is clear that $P[X' \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}] = P[X \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}]$.

From this, by setting $i_j = 2$, we conclude that $P[X' \in P_{r_j}^2] = P[X \in P_{r_j}^2]$, and thus

$$\begin{aligned} &P[X'_{E^1} = b_1, \ldots, X'_{E^n} = b_n \mid X'_{E^{*,1}} = b_1^*, \ldots, X'_{E^{*,n^*}} = b_{n^*}^*] \\ &= P[X_{E^1} = b_1, \ldots, X_{E^n} = b_n \mid X_{E^{*,1}} = b_1^*, \ldots, X_{E^{*,n^*}} = b_{n^*}^*]. \end{aligned}$$

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

We now check that $H(X') \geq H(X)$. 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

$$\begin{aligned} &- \sum_{x \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}} P[X = x] \cdot \log P[X = x] \\ &\leq P[X \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}] \cdot \log \frac{|P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}|}{P[X \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}]} \\ &= \sum_{x \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}} \frac{P[X \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}]}{|P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}|} \cdot \log \frac{|P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}|}{P[X \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}]} \\ &= - \sum_{x \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}} P[X' = x] \cdot \log P[X' = x]. \end{aligned}$$

Using the inequality above, the desired result comes from

$$\begin{aligned} H(X') &= - \sum_{i_1, \ldots, i_N} \sum_{x \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}} P[X' = x] \cdot \log P[X' = x] \\ &\geq - \sum_{i_1, \ldots, i_N} \sum_{x \in P_{r_1}^{i_1} \cap \ldots \cap P_{r_N}^{i_N}} (P[X = x] \cdot \log P[X = x] = H(X). \end{aligned}$$

□

Proof (Theorem 3.6) 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 $S(\varphi) = \{s_1, \dots, s_k\}$ the set of propositional symbols present in φ.

We will set $\mathbf{B} = \{0, 1\}$, $\mathbf{E} = \{s_1, \dots, s_k, c_1, \dots, c_n, \varphi\}$. 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 $\varphi = c_1 \wedge \dots \wedge c_n$. For each $i = 1, \dots, n$ and each $j = 1, 2, 3$, let $s_{ij} \in S(\phi)$ 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 $\mathbf{R}$ describes how each E^{c_i} may be computed from the expressions $E^{s_1}, \dots, E^{s_k}$ and how E^{φ} may be computed from $E^{c_i}, i = 1, \dots, n$. Thus, $\mathbf{R}$ contains the properties:

$$\begin{gathered} P[c_i = 1 \mid s_{ij} = b_{ij}] = 1, j \in \{1, 2, 3\}, i \in \{1, \dots, n\} \\ P[c_i = 0 \mid s_{i1} = 1 - b_{ij} \text{ for } j \in \{1, 2, 3\}] = 1, i \in \{1, \dots, n\}, \end{gathered} \tag{13}$$

and

$$\begin{gathered} P[\varphi = 0 \mid c_i = 0] = 1, i \in \{1, \dots, n\} \\ P[\varphi = 1 \mid c_1 = 1, \dots, E^{c_n} = 1]. \end{gathered} \tag{14}$$

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

$$\varphi \text{ is satisfiable } \iff P[\varphi = 1] > 0 \text{ in } \mathcal{D}^H(\mathbf{R}). \tag{15}$$

In order to do this, we first note that each

$$x = (s_1, \dots, s_k, c_1, \dots, c_n, \varphi) \in \Omega$$

determines an assignment which attributes "true" to s_i if $e^{s_i} = 1$ and "false" otherwise.

Let X be a random varible taking values in Ω. Let $X_S = (s_1, \dots, s_k)$, $X_{C,\varphi} = (c_1, \dots, c_n, \varphi)$, so that $X = (X_S, X_{C,\varphi})$. It is clear that from equations (13), (14) that $X_{C,\varphi}$ is determined by X_S. Thus, we have $H(X) = H(X_S)$, and it becomes clear that X_S has uniform probability distribution in the distribution of maximum entropy $\mathcal{D}^H$.

(15) is now clear: if ϕ is satisfiable, then there is at least one assignment which satisfies it, and if x_S is the value of X_S which induces that assignment, then $P[\varphi = 1] \geq P[X_S = x_S] = \frac{1}{2^k} > 0$. On the other hand, if f ϕ is not satisfiable, then for all x_S we have $P[\varphi = 1, X_S = x_S] = 0$, and hence $P[\varphi = 1] = 0$. $\square$

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 [32] 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 $\mathbf{Exp}^A$ 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 given 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;
- $L^m_3, E^m_3, E^m_{O,j}, E^m_{I,j}, E^{*,m}_{I,j} \in \mathbf{Exp}^A$ 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:

- $$P[L^m_3 = b \mid E^m_3 = b] = \frac{1}{16}, \text{ for } b \in \mathbb{Z}_2^6; \tag{16}$$
- $$\begin{aligned} P[OS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'}) \oplus OS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^{m'}) = E^m_{O,j} \\ \mid L^m_3 = b, E^m_3 = b] = 1, \end{aligned} \tag{17}$$

  where $j \in \{2, 5, 6, 7, 8\}$;
- $$P[IS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^{m'}) = b \mid E^m_{I,j} = b] = 1, \tag{18}$$
- $$P[IS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^{m'}) = b \mid E^{*,m}_{I,j} = b] = 1 \tag{19}$$

  (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 $K^{m'}$);

- $$\begin{aligned}P[&contains(test_j(IS_{j,k}(L^m(\mathbf{b})R^m(\mathbf{b'}),K^{m'}),\\ &IS_{j,k}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}),K^{m'}),b_O),K^{m'}_{i,j})\\ &\mid OS_{6,k}(L^m(\mathbf{b})R^m(\mathbf{b'}),K^{m'})\oplus OS_{6,k}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}),K^{m'})=b_O]=1;\end{aligned} \tag{20}$$

- $$\begin{aligned}P[&contains(test_j(IS_{j,k}(L^m(\mathbf{b})R^m(\mathbf{b'}),K^{m'}),\\ &IS_{j,k}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}),K^{m'}),c),k]=\frac{1}{16},\end{aligned} \tag{21}$$

  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 $K^{m'}_{i,j}$ [7]

- $$\begin{aligned}P[&filters(E^m_{I,j_1},E^{*,m}_{I,j_1},E^m_{O,j_1},\ldots,E^m_{I,j_5},E^{*,m}_{I,j_5},E^m_{O,j_5})=1\\ &\mid L^m_3=b,E^m_3=b]=1\end{aligned} \tag{22}$$

- $$\begin{aligned}P[&filters(E^m_{I,j_1},E^{*,m}_{I,j_1},E^m_{O,j_1},\ldots,E^m_{I,j_5},E^{*,m}_{I,j_5},E^m_{O,j_5})=0\\ &\mid L^m_3=b',E^m_3=b]=5/8, b\neq b'.\end{aligned} \tag{23}$$

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 (16), $P[E^m_3 = L^m_3] = 1/16$; by (22), if this is the case, then $filters^m = 1$. From (23), 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 (17) and the previous reasoning, we conclude that

$$\begin{aligned}P[&OS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}),K^1)\oplus OS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}),K^1)=E^m_{O,j}\\ &\mid filters^m=1]=\frac{1}{6}.\end{aligned} \tag{24}$$

[7] 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.

(18) and (19) 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

$$E^m_{O,j} = OS_{6,j}(L^m(\mathbf{b})R^m(\mathbf{b'}), K^1) \oplus OS_{6,j}(L^{*,m}(\mathbf{b})R^{*,m}(\mathbf{b'}), K^1), \tag{25}$$

which happens with probability 1/6, we have

$$contains^m_j(k) \equiv contains(test_j(E^m_{I,j}, E^{*,m}_{I,j}, E^m_{O,j}), k) =$$

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

and thus, by (20), $contains^m_j(K^1_{6,j}) = 1$. If $k \neq K^1_{6,j}$ is some other bitstring, $P[contains^m_j(k) = 1] = 1/16$ and $P[contains^m_j(k) = 0] = 15/16$. If (25) is not verified, which happens with probability 5/6, we have $P[contains^m_j(k) = 1] = 1/16$ 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 $contains^m_j(k), contains^m_{j'}(k)$ are independent for $j \neq j'$;
- the results of $contains^m_j(k), contains^m_j(k')$ are independent for $k \neq k'$;
- the results of $contains^m_j(k), contains^{m'}_j(k)$ are independent for $m \neq m'$.

The attacker will try to guess the bits $K^1_{6,j}$ 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 $P[contains^m_j(k) = 1] = 1/6 + 5/6 \cdot 1/16 = 7/32$ if $k = K^1_{6,j}$ and $P[contains^m_j(k) = 1] = 1/16$ if $k \neq K^1_{6,j}$.

Fix $j \in \{2, 5, 6, 7, 8\}$, $K^1_{6,j} \neq k \in \mathbb{Z}_2^6$, and recall that in the distribution of maximum entropy the results of $contains^m_j$ 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 $contains^m_j(k)$ than $contains^m_j(K^1_{6,j})$ is given by

$$p = \sum_{i=1}^{75} (\binom{75}{i} (\frac{7}{32})^i (\frac{1}{16})^{75-i} \sum_{j=i+1}^{75} \binom{75}{j} (\frac{1}{16})^j (\frac{15}{16})^{75-j}) \approx 0.0013. \tag{26}$$

Thus, the probability that some bitstring $K^1_{6,j} \neq k \in \mathbb{Z}_2^6$ has more positive results than $K^1_{6,j}$ 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 $contains^m_j(k)$ is the one represented by $K^1_{6,j}$ is at least $(1 - 63p)^5 \approx 0.65$. □

Proof (Theorem 3.8) The first step of the proof is to see that we may specify an attacker A' with the two properties described above.

For this, suppose that, whenever the attacker A would make a probabilistic decision (whether for choosing his next action in the protocol, deciding to wait for some step of a protocol to be executed or for some internal computation), $\tilde{A}$ 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 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).

$\tilde{A}$'s cryptanalytical knowledge contains the property

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

(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 let $e \in \mathcal{F}^{A'}$ (e is as defined in equation (4)), and consider the properties

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

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

Now, given $g = b$ for some b, these properties imply that

$$P[E = b] = s(\eta),$$

$$P[E \notin \mathcal{A}(E_{\eta})] = 0.$$

when $E \leftarrow \mathcal{D}_{\eta}^{H}$ is sampled from the distribution of maximum entropy given A''s cryptanalytical knowledge.

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

$$P[E = b'] = \frac{1 - s(\eta)}{|\mathcal{A}(E_{\eta})| - 1}$$

for $b' \neq b$, $b' \in \mathcal{A}(E_{\eta})$.

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

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

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 demonstration. □

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 $P[c^m = c^{m'}] = \frac{1}{2\eta}$, $m \neq m'$. Equations (7), (8) 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 (7)). 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 $B = \frac{4}{\sqrt{e}}$.

Now, 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

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

$\square$