An extended abstract of this paper appears in *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997. This is the full paper.

A Concrete Security Treatment of Symmetric Encryption

M. Bellare* A. Desai* E. Jokipii* P. Rogaway†

September 2000

Abstract

We study notions of security and schemes for symmetric (ie. private key) encryption in a concrete security framework.

We give several different notions of security and analyze the concrete complexity of reductions among them. Next we provide concrete security analyses of various methods of encrypting using a block cipher, including two of the most popular methods, Cipher Block Chaining and Counter Mode. We establish tight bounds (meaning matching upper bounds and attacks) on the success of adversaries as a function of their resources.

* Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-Mail: {mihir, adesai, ej}@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/{mihir, adesai, ej}. Supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

† Dept. of Computer Science, Engineering II Bldg., University of California at Davis, Davis, CA 95616, USA. E-mail: rogaway@cs.ucdavis.edu. URL: http://wwwcsif.cs.ucdavis.edu/~rogaway. Supported in part by NSF CAREER Award CCR-9624560.

Contents

1 Introduction

An encryption scheme enables Alice to send a message to Bob in such a way that an adversary Eve does not gain significant information about the message content. This is the classical problem of cryptography. It is usually considered in one of two settings. In the *symmetric* (private-key) one, encryption and decryption are performed under a key shared by the sender and receiver. In the *asymmetric* (public-key) setting the sender has some public information and the receiver holds some corresponding secret information.

In this paper we have two goals. The first is to study notions of security for symmetric encryption in the framework of concrete security. This means we will look at the concrete complexity of reductions between different notions. We want to prove both upper and lower bounds. In this way we can establish tight relations between the notions and can compare notions (even though polynomially reducible to each other) as stronger or weaker.

The second goal is to provide a concrete security analysis of some specific symmetric encryption schemes. The schemes we consider are in pervasive use, and yet have never received any formal analysis (concrete or otherwise) in the tradition of provable security. We want to remedy this. Once again the goal is to find tight bounds on the success probability of an adversary as a function of the resources it expends. This involves proving both an upper bound and a matching lower bound.

BACKGROUND. The pioneering work of Goldwasser and Micali [15] was the first to introduce formal notions of security for encryption. Specifically, they presented two notions of security for asymmetric encryption, "semantic security" and "polynomial security," and proved them equivalent with respect to polynomial-time reductions. Micali, Rackoff and Sloan [22] showed that (appropriate versions of) these notions were also equivalent to another notion, suggested by Yao [26]. A uniform complexity treatment of notions of asymmetric encryption is given by Goldreich [11]. Some adaptations of these notions to the symmetric setting are presented by Luby in [20, Chapters 11–12].

Goldwasser and Micali [15] also specified an asymmetric encryption scheme whose security (in the senses above) is polynomial-time reducible from quadratic residuosity. Subsequently many other schemes have emerged (eg. [9, 1, 26, 13, 7]), based on various hard problems.

CONCRETE SECURITY. The viewpoint in all the works above is that two notions of security are equivalent if there is a polynomial-time reduction between them; and a scheme is declared provably secure if there is some polynomial-time reduction from a hard problem to it. These are certainly basic questions, but we believe that, once the answers are known, it is important to classify notions and schemes in a more precise way.

To make an analogy, caring only about polynomial-time reducibility in cryptography is a bit like caring only whether a computational problem is or is not in P. Yet we know there are a lot of interesting questions (including most of the field of algorithms, and much of complexity theory) centered around getting further information about problems already known to be in P. Such information helps to better understand the problem and is also essential for practical applications.

Paying attention to the concrete complexity of polynomially-equivalent notions in cryptography has similar payoffs. In particular, when reductions are not security-preserving it means that one must use a larger security parameter to be safe, reducing efficiency. Thus, in the end, one pays for inefficient reductions in either assurance or running time.

Our approach for doing concrete security is that of Bellare, Kilian and Rogaway [6], wherein one parameterizes the resources involved and measures adversarial success by an explicit function on them. The approach is non-asymptotic and applicable to functions with a finite domain.

We will be concerned not only with proving security by exhibiting concrete bounds, but also with showing that these bounds are the best possible, which is done by exhibiting matching attacks.

We follow works of Bellare et al. [5, 3], who did this for certain message authentication schemes.

Though this paper is concerned with concrete security for symmetric encryption, we believe that, in general, concrete security is one of the major emerging avenues for productive research in theoretical cryptography.

NOTIONS OF SECURITY. We will consider four definitions of security for symmetric encryption and examine the complexity of reductions between them. Each of our definitions actually capture *two* notions: one against chosen-plaintext attack (CPA) and the other against chosen-ciphertext attack (CCA). The first definition, which we call "left-or-right indistinguishability" (LOR) is new, and the second, "real-or-random indistinguishability" (ROR) is a variant of it. The next two definitions, "find-then-guess security" (FTG) and "semantic security" (SEM) are adaptations of the definitions of Goldwasser and Micali [15] to the symmetric setting. [1]

In order to model CPA we must give the adversary the ability to see ciphertexts. In the public key setting it can create them itself given the public key, but in the symmetric key setting the encryption key is secret so we must modify the model and provide the adversary with an oracle for the encryption function. The presence of the encryption oracle is one reason it would be untrue to regard the notion of symmetric encryption as a special case of asymmetric encryption. To model CCA, we must give the adversary, in addition to an encryption oracle, an oracle for the decryption function.

As indicated above, our approach to concrete security is via parameterization of the resources of the adversary A. We distinguish between A's running time, t (by convention, we include in this the space for A's program and the time to answer all of A's oracle queries); the number of queries, q_e, made by A to an encryption oracle; the amount of ciphertext A sees in response to its encryption oracle queries, μ_e; and, in the case of CCA, also the number of queries, q_d, made by A to a decryption oracle; and the amount of plaintext A sees in response to its decryption oracle queries, μ_d. With an eye towards practical applications, it is important to treat all of these resources separately. (Previous works would neglect q_e, μ_e, q_d, μ_d, since they are bounded by t. But as resources they are very different, because, typically, obtaining legitimate plaintext-ciphertext pairs is more problematic than performing local computation.) The security of a scheme under any of the notions is specified by giving bounds on an "advantage function" for that scheme. The advantage function is the maximum, over all adversaries restricted to some indicated resources, of the "advantage" that the adversary has (compared to simply guessing) in "breaking" the scheme. Of course what it means to "break" a scheme varies across the different notions.

REDUCTIONS AMONG THE NOTIONS. In this work, we only look at the complexity of reductions among notions under the same attack, $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA}\}$. That is, either both the notions being compared are defined against CPA or both are defined against CCA. It follows from our results here and the work of [4, 10, 19] that there can be no reductions from any of the notions of security against CPA to any of the notions of security against CCA.

We show that LOR-ATK and ROR-ATK are equivalent, up to a small constant factor in the reduction. (That is, we have security-preserving reductions between them.) We also show a security-preserving reduction from these notions to FTG-ATK. However, the reduction from FTG-ATK to LOR-ATK (or ROR-ATK) is not security-preserving. However, we show that the reduction we give is tight; one cannot hope to do better. We complete the picture by showing that SEM-ATK and FTG-ATK are equivalent.

From the above results it is clear that when one wants to prove the security of some encryption scheme $\mathcal{SE}$ it is best to give a tight reduction from ROR-ATK or LOR-ATK, since that implies

[1] In [15] the term "polynomial security" is used for the notion analogous to what we call "find-then-guess security."

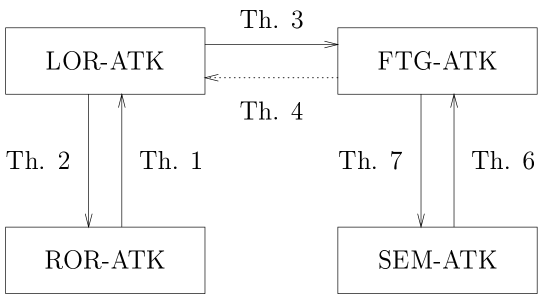

Figure 1: *Relating the notions for ATK $\in \{CPA, CCA\}$. A solid line from notion **A** to notion **B** means that there is a security-preserving reduction from **A** to **B**. A broken line indicates that the reduction is not security-preserving.*

good reductions to the rest. A summary of the reductions is given in Figure 1.

Although concrete security has been considered before in the context of scheme analysis [6, 5, 3, 8], this is the first work that considers it also for the purpose of relating different notions of security. That is, this is the first time *notions* are classified as weaker or stronger according to the complexity of the reductions between them.

Actually these results extend easily to the asymmetric setting. We focus on the symmetric mainly because that is the domain in which lie the schemes we want to analyze.

Security of Encryption Schemes. We analyze the security of some classic symmetric encryption schemes. Specifically, we look at three different modes of encryption with a block cipher (eg., DES): CBC (Cipher Block Chaining mode); CTR (Counter mode); and XOR (a stateless variant of the CTR mode).

In these schemes the underlying primitive is a pseudorandom function (PRF) or pseudorandom permutation (PRP) family F in which a particular function F_K, specified by a key K, maps l-bits to L-bits for fixed l, L. (For permutations, $l = L$.) To encrypt a message the applications of F_K are iterated in some scheme-dependent way. We wish to see how the security of the encryption scheme depends on the assumed security of the PRF family. We define the concrete security of PRF and PRP families as in [6], via parameterization of the time t' and the number of oracle queries q'. We define the advantage function for the PRF or PRP family, similar to the one defined for encryption schemes. The question then is: assuming F is a "good" PRF family (meaning it has a small advantage function for reasonable values of t', q'), what are values of t, q_e, μ_e such that the advantage function with those resources for the encryption scheme are small? We seek upper and lower bounds. (The latter represent the best known attacks.)

For the CTR scheme we show that if the underlying PRF family has an advantage function value ϵ' for resources t', q', then the advantage function value for the scheme is at most $2\epsilon'$, for resources $t = t'$, $\mu = q'l$ and any q. For the XOR scheme we show that with the above meanings, the advantage function value for the scheme is at most $2\epsilon' + \delta_{\mathrm{XOR}}$, where $\delta_{\mathrm{XOR}} = \mu(q-1)/(L2^l)$ and other resources are as before. We analyze CBC assuming the underlying family to be a PRP family, since the scheme must indeed be used with permutations. With t', q', ϵ' now understood to be associated to F as a PRP family, we show that for CBC, the corresponding advantage function value is at most $2\epsilon' + \delta_{\mathrm{CBC}}$, where $\delta_{\mathrm{CBC}} = (\mu^2 - \mu l)/(l^2 2^l)$ and the other resources are as before. In all cases, we show that these results are tight, up to a constant. Notice that even if the underlying PRF (or PRP) family is ideal (meaning, $\epsilon' = 0$), it is still possible for an adversary attacking the XOR or CBC schemes to derive some advantage. This is not true for CTR and hence we conclude that it has the best security.

In all the above the security is in the LOR-CPA sense. From what we said before this gives comparable bounds for security under any of the other three notions against CPA. There are simple (and well-known) attacks to show that none of the three schemes we look at are secure against CCA.

MORE RELATED WORK. We have already mentioned the most important related work, namely [15]. Here we provide some more detailed comparisons and histories and also discuss other work.

Since our results imply that the notions we consider are equivalent under polynomial time reductions, they can be viewed, at one level, as providing the analogue of [15] for the symmetric case. Luby [20] defines what is essentially find-then-guess security for symmetric encryption, and he mentions encryption using a pseudorandom function whose output length is the number of bits you wish to encrypt. In treating the asymmetric setting, [11] says that the symmetric case can be dealt with similarly. One ingredient missing in this view is that to model CPA one must, in the symmetric setting, supply the adversary with some means to encrypt. We extend polynomial and semantic security by providing the adversary with an encryption oracle. Stronger notions of asymmetric encryption than those of [15, 22] have appeared in the form of non-malleability [10] and chosen-ciphertext security [24, 25]. It can be gathered from results [4, 10, 19] obtained subsequent to this work that FTG-CCA implies all these other notions.

Works like [20, 12, 16] pay attention to concrete security to some extent but do not really go "all the way," in the sense that at some level their notions are still only caring about whether something is polynomial or not. Also the flavor is different from us in that their concern is more the security you can get for a certain investment of randomness, and the treatment remains asymptotic. Curiously, some earlier works had a more concrete treatment: in the asymmetric encryption arena, Alexi et. al. [1] were careful to specify the complexity of their reductions, a habit many later works unfortunately dropped.

The construction of a pseudorandom generator from a one-way function [17] provides a solution for symmetric encryption starting from a one-way function. In the current work existence is not the issue; we are interested in concrete security and the analysis of some particular schemes.

A concrete security analysis of the CBC MAC is provided in [6]. (The CBC MAC should not be confused with CBC encryption: The former is a message authentication code.) We build on their techniques, but those techniques do not directly solve the problems here. CBC mode encryption is standardized in [2, 18, 23].

2 Notions of Security

If $A(\cdot,\cdot,\ldots)$ is any probabilistic algorithm then $a \leftarrow A(x_1, x_2, \ldots)$ denotes the experiment of running A on inputs $x_1, x_2, \ldots$ and letting a be the outcome, the probability being over the coins of A. Similarly, if A is a set then $a \leftarrow A$ denotes the experiment of selecting a point uniformly from A and assigning a this value.

SYNTAX OF (SYMMETRIC) ENCRYPTION SCHEMES. A *(symmetric) encryption scheme* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three algorithms. The randomized *key generation* algorithm $\mathcal{K}$ takes input a security parameter $k \in \mathsf{N}$ and returns a key K; we write $K \stackrel{R}{\leftarrow} \mathcal{K}(k)$. The *encryption* algorithm $\mathcal{E}$ could be randomized or stateful. It takes the key K and a *plaintext* M to return a *ciphertext* C; we write $C \stackrel{R}{\leftarrow} \mathcal{E}_K(M)$. (If randomzed it flips coins anew on each invocation. If stateful, it uses and then updates a state that is maintained across invocations.) The *decryption* algorithm $\mathcal{D}$ is deterministic and stateless. It takes the key K and a string C to return either the corresponding plaintext M or the symbol $\perp$; we write $x \leftarrow \mathcal{D}_K(C)$ where $x \in \{0,1\}^* \cup \{\perp\}$. We require that $\mathcal{D}_K(\mathcal{E}_K(M)) = M$ for all $M \in \{0,1\}^*$.

We now give four definitions of security, each modeling both, chosen-plaintext attack and chosen-ciphertext attack (in the sense of Rackoff and Simon [25]). In each case, we allow the adversary access to an encryption oracle in some form; this is one feature distinguishing these definitions from previous ones. We will describe our definitions for stateless encryption schemes and later indicate how to modify them for stateful ones.

LEFT-OR-RIGHT INDISTINGUISHABILITY. The adversary is allowed queries of the form (x_0, x_1) where x_0, x_1 are equal-length messages. Two games are considered. In the first, each query is responded to by encrypting the left message; in the second, it is the right message. Formally, we define the *left-or-right* oracle $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$, where $b \in \{0,1\}$, to take input (x_0, x_1) and do the following: if $b = 0$ it computes $C \leftarrow \mathcal{E}_K(x_0)$ and returns C; else it computes $C \leftarrow \mathcal{E}_K(x_1)$ and returns C. We consider an encryption scheme to be "good" if a "reasonable" adversary cannot obtain "significant" advantage in distinguishing the cases $b = 0$ and $b = 1$ given access to the left-or-right oracle.

To model chosen-ciphertext attacks we allow the adversary to also have access to a decryption oracle. Note that if the adversary queries the decryption oracle at a ciphertext output by the left-or-right oracle, then it can obviously easily win the game. Therefore, we disallow it from doing so. Any other query is permissible.

Definition 1 [LOR-CPA, LOR-CCA] *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. Let $b \in \{0,1\}$ and $k \in \mathsf{N}$. Let A_{cpa} be an adversary that has access to the oracle $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ and let A_{cca} be an adversary that has access to the oracles $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ and $\mathcal{D}_K(\cdot)$. Now, we consider the following experiments:*

Experiment $\mathbf{Exp}^{\text{lor-cpa-}b}_{\mathcal{SE},A_{\mathrm{cpa}}}(k)$	Experiment $\mathbf{Exp}^{\text{lor-cca-}b}_{\mathcal{SE},A_{\mathrm{cca}}}(k)$
$K \stackrel{R}{\leftarrow} \mathcal{K}(k)$	$K \stackrel{R}{\leftarrow} \mathcal{K}(k)$
$d \leftarrow A_{\mathrm{cpa}}^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))}(k)$	$d \leftarrow A_{\mathrm{cca}}^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b)),\mathcal{D}_K(\cdot)}(k)$
Return d	***Return*** d

Above it is mandated that A_{cca} never queries $\mathcal{D}_K(\cdot)$ on a ciphertext C output by the $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ oracle, and that the two messages queried of $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ always have equal length. We define the advantages of the adversaries via

$$\begin{aligned}
\mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE},A_{\mathrm{cpa}}}(k) &= \Pr[\,\mathbf{Exp}^{\text{lor-cpa-1}}_{\mathcal{SE},A_{\mathrm{cpa}}}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{lor-cpa-0}}_{\mathcal{SE},A_{\mathrm{cpa}}}(k) = 1\,] \\
\mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE},A_{\mathrm{cca}}}(k) &= \Pr[\,\mathbf{Exp}^{\text{lor-cca-1}}_{\mathcal{SE},A_{\mathrm{cca}}}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{lor-cca-0}}_{\mathcal{SE},A_{\mathrm{cca}}}(k) = 1\,]\,.
\end{aligned}$$

We define the advantage functions of the scheme as follows. For any integers $t, q_e, \mu_e, q_d, \mu_d$,

$$\begin{aligned}
\mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &= \max_{A_{\mathrm{cpa}}}\{\mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE},A_{\mathrm{cpa}}}(k)\} \\
\mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d) &= \max_{A_{\mathrm{cca}}}\{\mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE},A_{\mathrm{cca}}}(k)\}
\end{aligned}$$

where the maximum is over all $A_{\mathrm{cpa}}, A_{\mathrm{cca}}$ with "time complexity" t, each making at most q_e queries to the $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ oracle, totalling at most $\mu_e/2$ bits, and, in the case of A_{cca}, also making at most q_d queries to the $\mathcal{D}_K(\cdot)$ oracle, totalling at most μ_d bits. The scheme $\mathcal{SE}$ is said to be LOR-CPA secure (resp. LOR-CCA secure) if the function $\mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE},A}(\cdot)$ (resp. $\mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE},A}(\cdot)$) is negligible for any adversary A whose time complexity is polynomial in k. ▌

The "time complexity" is the worst case total execution time of the experiment, plus the size of the code of the adversary, in some fixed RAM model of computation. We stress that the total execution time of the experiment includes the time of *all* operations in the experiment, including the time for key generation and the computation of answers to oracle queries. Thus when the time complexity is polynomially bounded, so are all the other parameters. This convention for measuring time complexity and other resources of an adversary is used for all definitions in this paper. The advantage function is the maximum probability that the security of the scheme $\mathcal{SE}$ can be compromised by an adversary using the indicated resources.

Real-or-Random Indistinguishability. The idea is that an adversary cannot distinguish the encryption of text from the encryption of an equal-length string of garbage. (By transitivity, the adversary cannot distinguish from each other the encryption of any two equal-length strings.) Formally, we define the *real-or-random* oracle $\mathcal{E}_K(\mathcal{RR}(\cdot,b))$, where $b \in \{0,1\}$, to take input x and do the following: if $b=1$ it computes $C \leftarrow \mathcal{E}_K(x)$ and returns C; else it computes $C \leftarrow \mathcal{E}_K(r)$ where $r \stackrel{R}{\leftarrow} \{0,1\}^{|x|}$ and returns C. (It is understood that the oracle picks any coins that $\mathcal{E}$ might need if $\mathcal{E}$ is randomized, or updates its state appropriately if $\mathcal{E}$ is stateful.) The encryption scheme is "good" if no "reasonable" adversary cannot obtain "significant" advantage in distinguishing the cases $b=0$ and $b=1$ given access to the oracle.

Definition 2 [ROR-CPA, ROR-CCA] *Let $\mathcal{SE} = (\mathcal{K},\mathcal{E},\mathcal{D})$ be a symmetric encryption scheme. Let $b \in \{0,1\}$ and $k \in \mathsf{N}$. Let $A_{\rm cpa}$ be an adversary that has access to the oracle $\mathcal{E}_K(\mathcal{RR}(\cdot,b))$ and let $A_{\rm cca}$ be an adversary that has access to the oracles $\mathcal{E}_K(\mathcal{RR}(\cdot,b))$ and $\mathcal{D}_K(\cdot)$. Now, we consider the following experiments:*

Experiment $\mathbf{Exp}^{\text{ror-cpa-}b}_{\mathcal{SE},A_{\rm cpa}}(k)$	Experiment $\mathbf{Exp}^{\text{ror-cca-}b}_{\mathcal{SE},A_{\rm cca}}(k)$
$K \stackrel{R}{\leftarrow} \mathcal{K}(k)$	$K \stackrel{R}{\leftarrow} \mathcal{K}(k)$
$d \leftarrow A_{\rm cpa}^{\mathcal{E}_K(\mathcal{RR}(\cdot,b))}(k)$	$d \leftarrow A_{\rm cca}^{\mathcal{E}_K(\mathcal{RR}(\cdot,b)),\mathcal{D}_K(\cdot)}(k)$
Return d	***Return*** d

Above it is mandated that $A_{\rm cca}$ never queries $\mathcal{D}_K(\cdot)$ on a ciphertext C output by the $\mathcal{E}_K(\mathcal{RR}(\cdot,b))$ oracle. We define the advantages of the adversaries via

$$\begin{aligned}
\mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE},A_{\rm cpa}}(k) &= \Pr[\,\mathbf{Exp}^{\text{ror-cpa-}1}_{\mathcal{SE},A_{\rm cpa}}(k)=1\,] - \Pr[\,\mathbf{Exp}^{\text{ror-cpa-}0}_{\mathcal{SE},A_{\rm cpa}}(k)=1\,]\\
\mathbf{Adv}^{\text{ror-cca}}_{\mathcal{SE},A_{\rm cca}}(k) &= \Pr[\,\mathbf{Exp}^{\text{ror-cca-}1}_{\mathcal{SE},A_{\rm cca}}(k)=1\,] - \Pr[\,\mathbf{Exp}^{\text{ror-cca-}0}_{\mathcal{SE},A_{\rm cca}}(k)=1\,]\,.
\end{aligned}$$

We define the advantage functions of the scheme as follows. For any integers $t, q_e, \mu_e, q_d, \mu_d$,

$$\begin{aligned}
\mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e) &= \max_{A_{\rm cpa}}\{\mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE},A_{\rm cpa}}(k)\}\\
\mathbf{Adv}^{\text{ror-cca}}_{\mathcal{SE}}(k,t,q_e,\mu_e,q_d,\mu_d) &= \max_{A_{\rm cca}}\{\mathbf{Adv}^{\text{ror-cca}}_{\mathcal{SE},A_{\rm cca}}(k)\}
\end{aligned}$$

where the maximum is over all $A_{\rm cpa}, A_{\rm cca}$ with time complexity t, each making at most q_e queries to the $\mathcal{E}_K(\mathcal{RR}(\cdot,b))$ oracle, totalling at most μ_e bits, and, in the case of $A_{\rm cca}$, also making at most q_d queries to the $\mathcal{D}_K(\cdot)$ oracle, totalling at most μ_d bits. The scheme $\mathcal{SE}$ is said to be ROR-CPA secure (resp. ROR-CCA secure) if the function $\mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE},A}(\cdot)$ (resp. $\mathbf{Adv}^{\text{ror-cca}}_{\mathcal{SE},A}(\cdot)$) is negligible for any adversary A whose time complexity is polynomial in k. ▮

Find-then-Guess Security. This is an adaptation of the notion of polynomial security as given in [15, 22]. We imagine an adversary that runs in two stages. During the find stage, the adversary endeavors to come up with a pair of equal-length messages, x_0 and x_1, whose encryptions it wants to try to tell apart. It also retains some state information s that it may want to preserve to help it later. In the guess stage, it is given a random ciphertext y for one of the plaintexts x_0, x_1, together with the state information s. The adversary "wins" if it correctly identifies which plaintext goes with y. The encryption scheme is "good" if "reasonable" adversaries cannot win significantly more than half the time.

Definition 3 [FTG-CPA, FTG-CCA] *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. Let $b \in \{0,1\}$ and $k \in \mathsf{N}$. Let A_{cpa} be an adversary that has access to the oracle $\mathcal{E}_K(\cdot)$ and let A_{cca} be an adversary that has access to the oracles $\mathcal{E}_K(\cdot)$ and $\mathcal{D}_K(\cdot)$. Now, we consider the following experiments:*

Experiment $\mathbf{Exp}^{\text{ftg-cpa-}b}_{\mathcal{SE},A_{\text{cpa}}}(k)$	Experiment $\mathbf{Exp}^{\text{ftg-cca-}b}_{\mathcal{SE},A_{\text{cca}}}(k)$
$K \stackrel{R}{\leftarrow} \mathcal{K}(k)$	$K \stackrel{R}{\leftarrow} \mathcal{K}(k)$
$(x_0, x_1, s) \leftarrow A^{\mathcal{E}_K(\cdot)}_{\text{cpa}}(k, \mathsf{find})$	$(x_0, x_1, s) \leftarrow A^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)}_{\text{cca}}(k, \mathsf{find})$
$y \leftarrow \mathcal{E}_K(x_b)$	$y \leftarrow \mathcal{E}_K(x_b)$
$d \leftarrow A^{\mathcal{E}_K(\cdot)}_{\text{cpa}}(k, \mathsf{guess}, y, s)$	$d \leftarrow A^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)}_{\text{cca}}(k, \mathsf{guess}, y, s)$
Return d	***Return*** d

Above it is mandated that A_{cca} does not query $\mathcal{D}_K(\cdot)$ on the ciphertext y in the guess *stage and that the two messages (x_0, x_1) have equal length. We define the advantages of the adversaries via*

$$\begin{aligned}\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE},A_{\text{cpa}}}(k) &= \Pr[\,\mathbf{Exp}^{\text{ftg-cpa-1}}_{\mathcal{SE},A_{\text{cpa}}}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ftg-cpa-0}}_{\mathcal{SE},A_{\text{cpa}}}(k) = 1\,] \\ \mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE},A_{\text{cca}}}(k) &= \Pr[\,\mathbf{Exp}^{\text{ftg-cca-1}}_{\mathcal{SE},A_{\text{cca}}}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ftg-cca-0}}_{\mathcal{SE},A_{\text{cca}}}(k) = 1\,]\,.\end{aligned}$$

We define the advantage functions of the scheme as follows. For any integers $t, q_e, \mu_e, q_d, \mu_d$,

$$\begin{aligned}\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &= \max_{A_{\text{cpa}}}\{\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE},A_{\text{cpa}}}(k)\} \\ \mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d) &= \max_{A_{\text{cca}}}\{\mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE},A_{\text{cca}}}(k)\}\end{aligned}$$

where the maximum is over all $A_{\text{cpa}}, A_{\text{cca}}$ with time complexity t, each making at most q_e queries to the $\mathcal{E}_K(\cdot)$ oracle, totalling at most $(\mu_e - |x_0|)$ bits, and, in the case of A_{cca}, also making at most q_d queries to the $\mathcal{D}_K(\cdot)$ oracle, totalling at most μ_d bits. The scheme $\mathcal{SE}$ is said to be FTG-CPA secure (resp. FTG-CCA secure) if the function $\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE},A}(\cdot)$ (resp. $\mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE},A}(\cdot)$) is negligible for any adversary A whose time complexity is polynomial in k. ▮

Semantic Security. Goldwasser and Micali [15] explain semantic security by saying that whatever can be efficiently computed about the plaintext given the ciphertext can also be computed in the absence of the ciphertext. We adapt the formalizations of [15, 22] to the symmetric setting.

Our adversary will run in two stages. During the select stage it endeavors to come up with an advantageous message distribution $\mathcal{M}$. We assume that the message distribution is *valid*, meaning that all strings in $\mathcal{M}$ with non-zero probability have the same length. In the adversary's predict stage it is given a random ciphertext y for a plaintext x_1, chosen according to the distribution $\mathcal{M}$, and it has to output a function f and a function value α. It hopes that $\alpha = f(x)$. An encryption scheme is semantically secure if no reasonable adversary can guess $f(x)$ with probability significantly better than the probability $\alpha = f(x_0)$, for some hidden x_0 drawn randomly from $\mathcal{M}$. This comparison-based method of measuring an adversary's advantage follows the approach Bellare et al [4] used to capture the notion of non-malleability.

Previous formalizations required the condition to hold for all functions f. In our concrete treatment we allow the function f and the probability distribution $\mathcal{M}$ to be selected by the adversary.

Definition 4 [SEM-CPA, SEM-CCA] *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. Let $k \in \mathsf{N}$. Let A_{cpa} be an adversary that has access to the oracle $\mathcal{E}_K(\cdot)$ and let A_{cca} be an adversary that has access to the oracles $\mathcal{E}_K(\cdot)$ and $\mathcal{D}_K(\cdot)$. Now, we consider the following experiments:*

Experiment $\mathbf{Exp}^{\text{sem-cpa-}b}_{\mathcal{SE},A_{\text{cpa}}}(k)$	*Experiment* $\mathbf{Exp}^{\text{sem-cca-}b}_{\mathcal{SE},A_{\text{cca}}}(k)$
$K \stackrel{R}{\leftarrow} \mathcal{K}(k)$	$K \stackrel{R}{\leftarrow} \mathcal{K}(k)$
$(\mathcal{M}, s) \leftarrow A^{\mathcal{E}_K(\cdot)}_{\text{cpa}}(k, \mathsf{select})$	$(\mathcal{M}, s) \leftarrow A^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)}_{\text{cca}}(k, \mathsf{select})$
$x_0 \leftarrow \mathcal{M}\,;\ x_1 \leftarrow \mathcal{M}$	$x_0 \leftarrow \mathcal{M}\,;\ x_1 \leftarrow \mathcal{M}$
$y \leftarrow \mathcal{E}_K(x_1)$	$y \leftarrow \mathcal{E}_K(x_1)$
$(f, \alpha) \leftarrow A^{\mathcal{E}_K(\cdot)}_{\text{cpa}}(k, \mathsf{predict}, y, s)$	$(f, \alpha) \leftarrow A^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)}_{\text{cca}}(k, \mathsf{predict}, y, s)$
If $\alpha = f(x_b)$ *then* $d \leftarrow 1$; *else* $d \leftarrow 0$	*If* $\alpha = f(x_b)$ *then* $d \leftarrow 1$; *else* $d \leftarrow 0$
Return d	***Return*** d

Above it is mandated that A_{cca} does not query $\mathcal{D}_K(\cdot)$ on the ciphertext y in the $\mathsf{predict}$ *stage. We define the advantages of the adversaries via*

$$\begin{aligned}
\mathbf{Adv}^{\text{sem-cpa}}_{\mathcal{SE},A_{\text{cpa}}}(k) &= \Pr[\,\mathbf{Exp}^{\text{sem-cpa-1}}_{\mathcal{SE},A_{\text{cpa}}}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{sem-cpa-0}}_{\mathcal{SE},A_{\text{cpa}}}(k) = 1\,] \\
\mathbf{Adv}^{\text{sem-cca}}_{\mathcal{SE},A_{\text{cca}}}(k) &= \Pr[\,\mathbf{Exp}^{\text{sem-cca-1}}_{\mathcal{SE},A_{\text{cca}}}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{sem-cca-0}}_{\mathcal{SE},A_{\text{cca}}}(k) = 1\,]
\end{aligned}$$

We define the advantage functions of the scheme as follows. For any integers $t, q_e, \mu_e, q_d, \mu_d$,

$$\begin{aligned}
\mathbf{Adv}^{\text{sem-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &= \max_{A_{\text{cpa}}}\{\mathbf{Adv}^{\text{sem-cpa}}_{\mathcal{SE},A_{\text{cpa}}}(k)\} \\
\mathbf{Adv}^{\text{sem-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d) &= \max_{A_{\text{cca}}}\{\mathbf{Adv}^{\text{sem-cca}}_{\mathcal{SE},A_{\text{cca}}}(k)\}
\end{aligned}$$

where the maximum is over all $A_{\text{cpa}}, A_{\text{cca}}$ with time complexity t, each making at most q_e queries to the $\mathcal{E}_K(\cdot)$ oracle, totalling at most $(\mu_e - |x_0|)$ bits, and, in the case of A_{cca}, also making at most q_d queries to the $\mathcal{D}_K(\cdot)$ oracle, totalling at most μ_d bits. The scheme $\mathcal{SE}$ is said to be SEM-CPA secure (resp. SEM-CCA secure), if the function $\mathbf{Adv}^{\text{sem-cpa}}_{\mathcal{SE},A}(\cdot)$ (resp. $\mathbf{Adv}^{\text{sem-cca}}_{\mathcal{SE},A}(\cdot)$) is negligible for any adversary A whose time complexity is polynomial in k. ▮

MODIFYING THE DEFINITIONS FOR THE STATEFUL CASE. Definitions of security for stateful encryption schemes are obtained by modifying the above definitions in the natural way, adjusting how one answers oracle queries. For example, in Definition 2, $A^{\mathcal{E}_K(\mathcal{RR}(\cdot,0))}_{\text{cpa}}$ now means A_{cpa} with an oracle that maintains a state σ, initially ε. Upon receiving a query x it picks coins r and sets (σ', y) to be $\mathcal{E}_K(x, \sigma, r)$. It returns y as the answer to the oracle query and updates the state via $\sigma \leftarrow \sigma'$. Notice that the ciphertext (meaning y) is returned, but the updated state is not. (Thus we are abusing notation when we write $A^{\mathcal{E}_K(\mathcal{RR}(\cdot,0))}_{\text{cpa}}$; we ought to write $A^{\mathcal{E}^2_K(\mathcal{RR}(\cdot,0))}_{\text{cpa}}$.) Notice that the encryption oracles now have "memory": between invocations, the state is modified and retained. The notation $A^{\mathcal{E}_K(\mathcal{RR}(\cdot,1))}_{\text{cpa}}$ can be similarly re-interpreted, and the same approach applies to the other definitions.

3 Reductions Among the Notions

Here we look at the reductions among the different notions of security. We look at both upper bounds and lower bounds. Since we are paying attention to concrete security bounds, we can use our results to decide how strong is a notion of security relative to other notions to which it is polynomially equivalent. This information is useful because it helps us identify the most desirable starting points for reductions. We implicitly use this information when we demonstrate the security of schemes via reductions from left-or-right indistinguishability.

We use the notation $A \Rightarrow B$ to indicate a security-preserving reduction from notion A to notion B. $A \rightarrow B$ indicates a reduction (not necessarily security-preserving) from A to B. $A \not\Rightarrow B$ and $A \not\rightarrow B$ are the natural interpretations given the above. For concision and clarity, we relate the notions, simultaneously with respect to CPA and CCA. We let the string atk be instantiated by the formal symbols cpa, cca, while ATK is then the corresponding formal symbol from CPA, CCA. In the proofs of our claims, we use the convention that if atk = cpa then $\mathcal{O}^{-1} = \epsilon$. (When we say $\mathcal{O}^{-1} = \epsilon$, we mean $\mathcal{O}^{-1}$ is the function which, on any input, returns the empty string.)

The first two theorems say that our first two notions, left-or-right indistinguishability and real-or-random indistinguishability, are of essentially equivalent strength, under any attack.

Theorem 1 [ROR-ATK $\Rightarrow$ LOR-ATK] *For any scheme* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$,

$$\begin{aligned} \mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &\leq 2 \cdot \mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) \\ \mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d) &\leq 2 \cdot \mathbf{Adv}^{\text{ror-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d). \end{aligned}$$

Proof: Assume that A_1 is an adversary attacking $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ in the LOR-ATK sense. We construct a new adversary A_2, using A_1, that attacks $\mathcal{SE}$ in the ROR-ATK sense.

Let $\mathcal{O}_2(\cdot)$ be A_2's encryption oracle and $\mathcal{O}^{-1}(\cdot)$ its decryption oracle. $A_2^{\mathcal{O}_2(\cdot), \mathcal{O}^{-1}(\cdot)}$ will run A_1, using its oracles to provide a simulation of A_1's oracles.

For $b \in \{0,1\}$ and $|x_0| = |x_1|$, define $\mathcal{O}_1(\mathcal{LR}(x_0, x_1, b))$ as $\mathcal{O}_2(x_b)$.

Algorithm $A_2^{\mathcal{O}_2(\cdot), \mathcal{O}^{-1}(\cdot)}(k)$

(1) Let $b \stackrel{R}{\leftarrow} \{0,1\}$

(2) If $b = 0$ then $d \leftarrow A_1^{\mathcal{O}_1(\mathcal{LR}(\cdot,\cdot,0)), \mathcal{O}^{-1}(\cdot)}(k)$, else $d \leftarrow A_1^{\mathcal{O}_1(\mathcal{LR}(\cdot,\cdot,1)), \mathcal{O}^{-1}(\cdot)}(k)$.

(3) If $b = d$ then return 1 else return 0.

From the above description, it is easy to see that the time and query complexities are as claimed.

We now compute A_2's advantage. We consider $\mathbf{Exp}^{\text{ror-atk-}b}_{\mathcal{SE}, A_2}(k)$, freely referring to the random variables underlying this experiment. We have,

$$\mathbf{Adv}^{\text{ror-atk}}_{\mathcal{SE}, A_2}(k) \;=\; \Pr[\,\mathbf{Exp}^{\text{ror-atk-1}}_{\mathcal{SE}, A_2}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ror-atk-0}}_{\mathcal{SE}, A_2}(k) = 1\,]$$

When $\mathcal{O}_2(\cdot) = \mathcal{E}_K(\mathcal{RR}(\cdot, 0))$, we have that $\mathcal{O}_1(\mathcal{LR}(\cdot,\cdot,0))$ and $\mathcal{O}_1(\mathcal{LR}(\cdot,\cdot,1))$ return identically distributed answers. So, $\Pr[\,\mathbf{Exp}^{\text{ror-atk-0}}_{\mathcal{SE}, A_2}(k) = 1\,] = 1/2$. Hence,

$$\begin{aligned} \mathbf{Adv}^{\text{ror-atk}}_{\mathcal{SE}, A_2}(k) &= \Pr[\,\mathbf{Exp}^{\text{ror-atk-1}}_{\mathcal{SE}, A_2}(k) = 1\,] - 1/2 \\ &= 1/2 \cdot \Pr[\,\mathbf{Exp}^{\text{lor-atk-1}}_{\mathcal{SE}, A_1}(k) = 1\,] + 1/2 \cdot \Pr[\,\mathbf{Exp}^{\text{lor-atk-0}}_{\mathcal{SE}, A_1}(k) = 0\,] - 1/2 \\ &= 1/2 \cdot \Pr[\,\mathbf{Exp}^{\text{lor-atk-1}}_{\mathcal{SE}, A_1}(k) = 1\,] + 1/2 \cdot \Big(1 - \Pr[\,\mathbf{Exp}^{\text{lor-atk-0}}_{\mathcal{SE}, A_1}(k) = 1\,]\Big) - 1/2 \\ &= 1/2 \cdot \Big(\Pr[\,\mathbf{Exp}^{\text{lor-atk-1}}_{\mathcal{SE}, A_1}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{lor-atk-0}}_{\mathcal{SE}, A_1}(k) = 1\,]\Big) \\ &= 1/2 \cdot \mathbf{Adv}^{\text{lor-atk}}_{\mathcal{SE}, A_1}(k) \end{aligned}$$

Since A_1 is an arbitrary adversary, the claimed relation in the advantage functions follows. ▌

Theorem 2 **[LOR-ATK $\Rightarrow$ ROR-ATK]** *For any scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$,*

$$\begin{aligned} \mathbf{Adv}^{\text{ror-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &\leq \mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) \\ \mathbf{Adv}^{\text{ror-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d) &\leq \mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d). \end{aligned}$$

Proof: Assume that A_2 is an adversary attacking $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ in the ROR-ATK sense. We construct a new adversary A_1, using A_2, that attacks $\mathcal{SE}$ in the LOR-ATK sense.

Let $\mathcal{O}_1(\cdot, \cdot)$ be A_1's encryption oracle and $\mathcal{O}^{-1}(\cdot)$ its decryption oracle. $A_1^{\mathcal{O}_1(\cdot,\cdot),\mathcal{O}^{-1}(\cdot)}$ will run A_2, using its oracles to provide a simulation of A_2's oracles.

For any string x, define $\mathcal{O}_2(x)$ to be $\mathcal{O}_1(r, x)$ where $r \stackrel{R}{\leftarrow} \{0,1\}^{|x|}$ is chosen anew each time the oracle is invoked.

Algorithm $A_1^{\mathcal{O}_1(\cdot,\cdot),\mathcal{O}^{-1}(\cdot)}(k)$

(1) Return $A_2^{\mathcal{O}_2(\cdot),\mathcal{O}^{-1}(\cdot)}(k)$

It is clear that the time and query complexities are as claimed. For A_1's advantage, we have,

$$\mathbf{Adv}^{\text{lor-atk}}_{\mathcal{SE},A_1}(k) \;=\; \Pr[\, \mathbf{Exp}^{\text{ror-atk-1}}_{\mathcal{SE},A_2}(k) = 1\,] - \Pr[\, \mathbf{Exp}^{\text{ror-atk-0}}_{\mathcal{SE},A_2}(k) = 1\,] \;=\; \mathbf{Adv}^{\text{ror-atk}}_{\mathcal{SE},A_2}(k)$$

Since A_2 is an arbitrary adversary, the claimed relation in the advantage functions follows. ▮

Left-or-right indistinguishability and real-or-random indistinguishability constitute a *stronger* notion of security than the traditional find-then-guess notion. Intuitively, the adversary's job is harder with find-then-guess because it has to single out a single message pair on which to perform. This is illustrated by Theorems 3 and 4 and Proposition 5.

The first theorem says that a scheme with a certain security in the left-or-right sense has essentially the same security in the find-then-guess sense.

Theorem 3 **[LOR-ATK $\Rightarrow$ FTG-ATK]** *For any scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$,*

$$\begin{aligned} \mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &\leq \mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE}}(k, t, q_e + 1, \mu_e) \\ \mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d) &\leq \mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE}}(k, t, q_e + 1, \mu_e, q_d, \mu_d). \end{aligned}$$

Proof: Assume that A_3 is an adversary attacking $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ in the FTG-ATK sense. We construct a new adversary A_1, using A_3, that attacks $\mathcal{SE}$ in the LOR-ATK sense.

Let $\mathcal{O}_1(\cdot, \cdot)$ be A_1's encryption oracle and $\mathcal{O}^{-1}(\cdot)$ be its decryption oracle. $A_1^{\mathcal{O}_1(\cdot,\cdot),\mathcal{O}^{-1}(\cdot)}$ will run A_3, using its oracles to provide a simulation of A_3's oracles.

For any string x, define $\mathcal{O}_3(x)$ to be $\mathcal{O}_1(x, x)$. We assume, wlog, that A_3 does not query $\mathcal{O}^{-1}(\cdot)$ on any ciphertext it has previously obtained by querying $\mathcal{O}_3(\cdot)$.

Algorithm $A_1^{\mathcal{O}_1(\cdot,\cdot),\mathcal{O}^{-1}(\cdot)}(k)$

(1) Let $(x_0, x_1, s) \leftarrow A_3^{\mathcal{O}_3(\cdot),\mathcal{O}^{-1}(\cdot)}(k, \mathsf{find})$

(2) Let $d \leftarrow A_3^{\mathcal{O}_3(\cdot),\mathcal{O}^{-1}(\cdot)}(k, \mathsf{guess}, \mathcal{O}_1(x_0, x_1), s)$

(3) If $d = 0$ then return 0, else return 1.

It is clear that the time and query complexities are as claimed. For A_1's advantage, we have,

$$\mathbf{Adv}^{\text{lor-atk}}_{\mathcal{SE},A_1}(k) \;=\; \Pr[\,\mathbf{Exp}^{\text{ftg-atk-1}}_{\mathcal{SE},A_3}(k)=1\,] - \Pr[\,\mathbf{Exp}^{\text{ftg-atk-0}}_{\mathcal{SE},A_3}(k)=1\,] \;=\; \mathbf{Adv}^{\text{ftg-atk}}_{\mathcal{SE},A_3}(k)$$

Since A_3 is an arbitrary adversary, the claimed relation in the advantage functions follows. ▌

The next theorem says that if a scheme has a certain security in the find-then-guess sense, then it is secure in the left-or-right sense, but the security shown is quantitatively lower.

Theorem 4 [FTG-ATK → LOR-ATK] *For any scheme $\mathcal{SE}=(\mathcal{K},\mathcal{E},\mathcal{D})$,*

$$\begin{aligned}\mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e) &\le q_e\cdot\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e)\\ \mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE}}(k,t,q_e,\mu_e,q_d,\mu_d) &\le q_e\cdot\mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}}(k,t,q_e,\mu_e,q_d,\mu_d).\end{aligned}$$

Proof: Assume that A_1 is an adversary attacking $\mathcal{SE}=(\mathcal{K},\mathcal{E},\mathcal{D})$ in the LOR-ATK sense. We construct a new adversary A_3, using A_1, that attacks $\mathcal{SE}$ in the FTG-ATK sense.

Let $\mathcal{O}_3(\cdot)$ be A_3's encryption oracle and $\mathcal{O}^{-1}(\cdot)$ be its decryption oracle. For $b\in\{0,1\}$ and $|x_0|=|x_1|$, define $\mathcal{O}_1(\mathcal{LR}(x_0,x_1,b))$ to be $\mathcal{O}_3(x_b)$.

Algorithm $A_3^{\mathcal{O}_3(\cdot),\mathcal{O}^{-1}(\cdot)}(k,\mathsf{find})$

(1) Let $i\xleftarrow{R}\{1,\cdots,q_e\}$

(2) Run A_1 answering its encryption oracle queries with $\mathcal{O}_1(\mathcal{LR}(\cdot,\cdot,0))$ and decryption oracle queries with $\mathcal{O}^{-1}(\cdot)$, until the point at which it makes its i-th encryption oracle query, which we denote (x_0^i,x_1^i). (That is, A_1 has now made this query and is waiting for the response from the encryption oracle.) Let s be A_1's runtime state at this point.

(3) Return (x_0^i,x_1^i,s)

Algorithm $A_3^{\mathcal{O}_3(\cdot),\mathcal{O}^{-1}(\cdot)}(k,\mathsf{guess},y,s)$

(1) Resume execution of A_1 in state s by answering its i-th encryption oracle query (namely (x_0^i,x_1^i)) by y, and stop before it makes another oracle query.

(2) Continue execution of A_1, answering all encryption oracle queries now via $\mathcal{O}_1(\mathcal{LR}(\cdot,\cdot,1))$ and decryption oracle queries via $\mathcal{O}^{-1}(\cdot)$, until A_1 halts.

(3) If A_1 outputs 1 then return 0, else return 1.

Clearly, the time and query complexities are as given. We compute A_3's advantage using a standard hybrid argument. Towards this, we define a sequence of q_e+1 experiments: for $j=0\ldots q_e$ define $\mathbf{Exp}^{\text{hyb-atk-}j}_{\mathcal{SE},A_1}(k)$ to be an experiment in which one chooses $K\xleftarrow{R}\mathcal{K}(k)$ and runs A_1, answering the first j encryption oracle queries of A_1 via $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,0))$ and the rest via $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,1))$, and furthermore, if atk = cca, answering its decryption oracle queries via $\mathcal{D}_K(\cdot)$. The output of the experiment is defined to be the output of A_1.

Now consider the experiment $\mathbf{Exp}^{\text{ftg-atk-}b}_{\mathcal{SE},A_3}(k)$, where A_3 is the algorithm above. In this experiment, if $b=0$ then $y=\mathcal{E}_K(x_0^i)$ and, in the simulation, A_1's output would be that of $\mathbf{Exp}^{\text{hyb-atk-}(i+1)}_{\mathcal{SE},A_1}(k)$. On the other hand, if $b=1$ then $y=\mathcal{E}_K(x_1^i)$ and, in the simulation, A_1's output would be the same as $\mathbf{Exp}^{\text{hyb-atk-}i}_{\mathcal{SE},A_1}(k)$. Since i is chosen randomly from $\{1,\cdots,q_e\}$ by A_3, we have,

$$\begin{aligned}\mathbf{Adv}^{\text{ftg-atk}}_{\mathcal{SE},A_3}(k) &= (1/q_e)\cdot\textstyle\sum_{i=0}^{q_e-1}\Big(\Pr[\,\mathbf{Exp}^{\text{hyb-atk-}i}_{\mathcal{SE},A_1}(k)=1\,]-\Pr[\,\mathbf{Exp}^{\text{hyb-atk-}(i+1)}_{\mathcal{SE},A_1}(k)=1\,]\Big)\\ &= (1/q_e)\cdot\Big(\Pr[\,\mathbf{Exp}^{\text{hyb-atk-0}}_{\mathcal{SE},A_1}(k)=1\,]-\Pr[\,\mathbf{Exp}^{\text{hyb-atk-}q_e}_{\mathcal{SE},A_1}(k)=1\,]\Big)\\ &= (1/q_e)\cdot\mathbf{Adv}^{\text{lor-atk}}_{\mathcal{SE},A_1}(k)\end{aligned}$$

Since A_1 is an arbitrary adversary, the claimed relation in the advantage functions follows. ▌

The following proposition says that the drop in security above is not due to any weakness in the reduction but is intrinsic— we present a scheme having a higher security in the find-then-guess sense than in the left-or-right sense, with the gap being the same as in the theorem above. Obviously we cannot make such a statement if there are no secure encryption schemes around at all, so the theorem assumes there exists a secure scheme, and then constructs a different scheme exhibiting the desired gap.

Proposition 5 [FTG-ATK $\not\Rightarrow$ LOR-ATK] *Suppose $\mathcal{SE}$ is a stateless encryption scheme, over a message space containing $\{0,1\}$. Then, there exists a stateless encryption scheme $\mathcal{SE}'$, such that,*

$$\begin{aligned}
\mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE}'}(k,t,q_e,q_e) &= \mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE}'}(k,t,q_e,q_e,0,0) \geq 0.632 \\
\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}'}(k,t,q_e,\mu_e) &\leq \mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e) + 1/q_e \\
\mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}'}(k,t,q_e,\mu_e,q_d,\mu_d) &\leq \mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}}(k,t,q_e,\mu_e,q_d,\mu_d) + 1/q_e
\end{aligned}$$

Furthermore, then there exists a stateful encryption scheme $\mathcal{SE}''$, such that,

$$\begin{aligned}
\mathbf{Adv}^{\text{lor-cpa}}_{\mathcal{SE}''}(k,t,q_e,q_e) &= \mathbf{Adv}^{\text{lor-cca}}_{\mathcal{SE}''}(k,t,q_e,q_e,0,0) = 1 \\
\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}''}(k,t,q_e,\mu_e) &\leq \mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e) + 1/q_e \\
\mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}''}(k,t,q_e,\mu_e,q_d,\mu_d) &\leq \mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}}(k,t,q_e,\mu_e,q_d,\mu_d) + 1/q_e .
\end{aligned}$$

Proof: Let $\mathcal{SE} = (\mathcal{E},\mathcal{D},\mathcal{K})$ be the given encryption scheme. We now define $\mathcal{SE}' = (\mathcal{E}',\mathcal{D}',\mathcal{K}')$ and show that it has the claimed properties. Set $\mathcal{K}' = \mathcal{K}$. Define encryption as:

Algorithm $\mathcal{E}'_K(x)$

(1) Pick $i \xleftarrow{R} \{1,\cdots,q_e\}$

(2) If $i = 1$ then return $0 \,\|\, x$, else return $1 \,\|\, \mathcal{E}_K(x)$

$\mathcal{D}'$ is as one would expect. Now consider the following adversary A_1 attacking $\mathcal{SE}'$ in the LOR-ATK sense. Let $\mathcal{O}_1(\cdot,\cdot)$ be A_1's encryption oracle and $\mathcal{O}^{-1}(\cdot)$ be its decryption oracle.

Algorithm $A_1^{\mathcal{O}_1(\cdot,\cdot),\mathcal{O}^{-1}(\cdot)}(k)$

(1) Fix a pair x_1, x_2 of distinct, equal length messages. (For concreteness $x_0 = 0$ and $x_1 = 1$, which we assumed are in the message space of $\mathcal{SE}$.)

(2) For $j = 1,\cdots,q_e$ do: $y_j \leftarrow \mathcal{O}_1(x_0,x_1)$

(3) If there is some j such that $y_j = 0 \,\|\, x_0$, then return 0; else return 1.

One can check that A_1's advantage is the probability that the i value chosen by $\mathcal{E}'_K$ is 1 in at least one of the q_e encryptions, namely $\mathbf{Adv}^{\text{lor-atk}}_{\mathcal{SE},A_1}(k) = 1-(1-1/q_e)^{q_e} \approx 1-1/e$.

Notice A_1 makes q_e queries, each consisting of two 1-bit messages, so its complexity is as claimed.

A find-then-guess adversary making q_e queries must hope that its challenge in the guess state, y, begins with a 0. If not, it can achieve no advantage over and above that of an adversary attacking $\mathcal{SE}$. With probability $1/q_e$ it is the case that y begins with 0, so $\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}'}(k,t,q_e,\mu_e)$ (respectively, $\mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}'}(k,t,q_e,\mu_e,q_d,\mu_d)$) is at most $\epsilon' + (1-\epsilon')/q_e \leq \epsilon' + 1/q_e$, where ϵ' is $\mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e)$ (respectively, $\mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}}(k,t,q_e,\mu_e,q_d,\mu_d)$).

Notice that $\mathcal{SE}'$ is stateless (as long as $\mathcal{SE}$ is stateless). If we allow the constructed scheme to be stateful we can slightly improve the constant factor in the gap between the securities, making ϵ' exactly 1. To do this we define a stateful encryption scheme $\mathcal{SE}'' = (\mathcal{E}'', \mathcal{D}'', \mathcal{K}'')$ which maintains a counter ctr, initially zero. The key generator $\mathcal{K}''$ outputs (i, a) where $i \stackrel{R}{\leftarrow} \{1, \ldots, q_e\}$ and $K \stackrel{R}{\leftarrow} \mathcal{K}(k)$. $\mathcal{E}''$ is as follows:

Algorithm $\mathcal{E}''_{i,K}(x, ctr)$

(1) $ctr \leftarrow ctr + 1$

(2) If $i = ctr$ then return $(ctr, 0 \,\|\, x)$, else return $(ctr, 1 \,\|\, \mathcal{E}_K(x))$

(Remember that according to our syntax for stateful schemes the output of the encryption algorithm is a pair consisting of the new state (here the updated counter) and the actual ciphertext.) $\mathcal{D}''$ is as one would expect. If we consider the same left-or-right adversary A_1 as above, executing now with scheme $\mathcal{SE}''$, we see that it is guaranteed to receive, in its q_e queries, a response whose first bit is 0. So the advantage function of $\mathcal{SE}''$ in the LOR-ATK sense is now 1, while it remains the same as that of $\mathcal{SE}'$ in the FTG-ATK sense. ▮

In the above, think of the advantage function value of $\mathcal{SE}$ as very small (essentially zero). The constructed scheme $\mathcal{SE}'$ can be broken with probability $\epsilon' = 0.632$, using q_e queries, in the left-or-right sense, meaning it is completely insecure under this notion. However, the probability of breaking it (with comparable resources) in the find-then-guess sense is $\epsilon \approx 1/q_e$. The probabilities obey the relation $q_e \epsilon = \Theta(\epsilon')$, showing that Theorem 4 is essentially tight. Furthermore, if one allows the scheme to be stateful, one can make ϵ' exactly one, so that $q_e \epsilon \approx \epsilon'$.

Semantic security is too complex to make it a good starting point for proving schemes secure. Still, as the next theorem indicates, it is nice that there is a strong reduction from semantic security to find-then-guess security. Notice that for this only requires semantic security to hold for a particular and simple function, the identity function, and a particular and simple distribution over the message space. This theorem is implicit in [15] for the asymmetric setting and their proof is easily adapted to the symmetric setting.

Theorem 6 [SEM-ATK ⇒ FTG-ATK] *For any scheme* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$,

$$\begin{aligned} \mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) &\leq \mathbf{Adv}^{\text{sem-cpa}}_{\mathcal{SE}}(k, t, q_e, \mu_e) \\ \mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d) &\leq \mathbf{Adv}^{\text{sem-cca}}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d). \end{aligned}$$

Proof: Assume that A_3 is an adversary attacking $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ in the FTG-ATK sense. We construct a new adversary A_4, using A_3, that attacks $\mathcal{SE}$ in the SEM-ATK sense. We use the standard reduction of [15], which is easily extended to take into account the presence of oracles.

Let $\mathcal{O}_4(\cdot)$ be A_4's encryption oracle and $\mathcal{O}^{-1}(\cdot)$ be its decryption oracle.

Algorithm $A_4^{\mathcal{O}_3(\cdot), \mathcal{O}^{-1}(\cdot)}(k, \mathsf{select})$

(1) Let $(x_0, x_1, s) \leftarrow A_3^{\mathcal{O}_4(\cdot), \mathcal{O}^{-1}(\cdot)}(k, \mathsf{find})$

(2) Return $((x_0, x_1), (s, (x_0, x_1)))$

That is, $\mathcal{M}$ is the pair (x_0, x_1), with a probability $1/2$ assigned to each of x_0 and x_1.

Algorithm $A_4^{\mathcal{O}_4(\cdot), \mathcal{O}^{-1}(\cdot)}(k, \mathsf{predict}, y, (s, (x_0, x_1)))$

(1) Let $d \leftarrow A_3^{\mathcal{O}_4(\cdot), \mathcal{O}^{-1}(\cdot)}(k, \mathsf{guess}, y, s)$

(2) Return (f, x_d), where f is the identity function.

For the advantage of A_4 we have,

$$\mathbf{Adv}^{\text{sem-atk}}_{\mathcal{SE},A_4}(k) \;=\; \Pr[\,\mathbf{Exp}^{\text{ftg-atk-1}}_{\mathcal{SE},A_3}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{ftg-atk-0}}_{\mathcal{SE},A_3}(k) = 1\,] \;=\; \mathbf{Adv}^{\text{ftg-atk}}_{\mathcal{SE},A_3}(k)$$

Uaing this, we get the claimed relation in the advantage functions. ▮

Combining this with Theorem 4 yields a reduction from security in the semantic sense to security in the left-or-right sense, but this reduction inherits the security loss of the reduction of Theorem 4. As before it turns out this loss is inherent: security in the left-or-right sense is a stronger notion. The example to see this is essentially the same as that in the proof of Proposition 5 but the setup becomes more complicated. We do not discuss it further here.

Theorem 7 [FTG-ATK ⇒ SEM-ATK] *For any scheme* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$,

$$\begin{aligned} \mathbf{Adv}^{\text{sem-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e) &\le 2 \cdot \mathbf{Adv}^{\text{ftg-cpa}}_{\mathcal{SE}}(k,t,q_e,\mu_e) \\ \mathbf{Adv}^{\text{sem-cca}}_{\mathcal{SE}}(k,t,q_e,\mu_e,q_d,\mu_d) &\le 2 \cdot \mathbf{Adv}^{\text{ftg-cca}}_{\mathcal{SE}}(k,t,q_e,\mu_e,q_d,\mu_d). \end{aligned}$$

Proof: Assume that A_4 is an adversary attacking $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ in the SEM-ATK sense. We construct a new adversary A_3, using A_4, that attacks $\mathcal{SE}$ in the FTG-ATK sense.

Let $\mathcal{O}_3(\cdot)$ be A_3's encryption oracle and $\mathcal{O}^{-1}(\cdot)$ be its decryption oracle.

Algorithm $A_3^{\mathcal{O}_3(\cdot),\mathcal{O}^{-1}(\cdot)}(k, \mathsf{find})$

(1) Let $(\mathcal{M}, s) \leftarrow A_4^{\mathcal{O}_3(\cdot),\mathcal{O}^{-1}(\cdot)}(k, \mathsf{select})$

(2) Let $x_0 \leftarrow \mathcal{M}$; $x_1 \leftarrow \mathcal{M}$

(3) Let $s' \leftarrow (\mathcal{M}, s, x_0, x_1)$

(4) Return (x_0, x_1, s')

Algorithm $A_3^{\mathcal{O}_3(\cdot),\mathcal{O}^{-1}(\cdot)}(k, \mathsf{guess}, y, (\mathcal{M}, s, x_0, x_1))$

(1) Let $(f, z) \leftarrow A_4^{\mathcal{O}_3(\cdot),\mathcal{O}^{-1}(\cdot)}(k, \mathsf{predict}, y, s)$

(2) If $z = f(x_1)$ then return 1; else return a random bit.

For the advantage of A_3 we have,

$$\mathbf{Adv}^{\text{ftg-atk}}_{\mathcal{SE},A_3}(k) \;=\; \Pr[\,\mathbf{Exp}^{\text{sem-atk-1}}_{\mathcal{SE},A_4}(k) = 1\,] - \Pr[\,\mathbf{Exp}^{\text{sem-atk-0}}_{\mathcal{SE},A_4}(k) = 1\,] \;=\; \mathbf{Adv}^{\text{sem-atk}}_{\mathcal{SE},A_4}(k)$$

Using this, we get the claimed relation in the advantage functions. ▮

In earlier work [15, 22, 11] no restriction was made on the complexity of f; it was even allowed to be uncomputable. Clearly semantic security against such very complex functions is not captured by Definition 4. There are alternate definitions and theorems (following techniques of [11]) that are useful in talking about complex functions f (but less useful when talking about simple functions). We do not pursue this more at the moment because, as we have indicated above, other notions of security are more suitable for practice.

Putting things together, showing an encryption scheme left-or-right secure or real-or-random secure implies tight reductions to all other notions Showing an encryption scheme find-then-guess secure or semantically secure does not. Thus, if the bounds are equal, it is better to demonstrate security with respect to one of the first two notions, since that immediately translates into equally-good bounds for the other notions.

Asymptotic security. The above theorems imply that all the definitions considered (under the same attack) are equivalent under polynomial time reductions, because, as the theorems indicate, all the translations involve only polynomial factors. We are just saying something stronger.

Asymmetric encryption. All of the above definitions and results carry over to the asymmetric setting. In that setting it is not necessary to give the adversary an encryption oracle for the purpose of facilitating a chosen plaintext attack (but the encryption oracle remains for left-or-right indistinguishability and real-or-random indistinguishability for the purpose of testing the adversary's effectiveness). For all four definitions it is important to provide the adversary with the public key. Then it remains true, even in the asymmetric setting that, from the point of view of concrete security, to prove a good bound on left-or-right indistinguishability, say, is "better" than providing an equally-good bound on find-then-guess security.

4 Finite PRFs and PRPs

The symmetric encryption schemes we study in this paper are based on finite pseudorandom functions [6], a concrete security version of the original notion of pseudorandom functions [14]. We thus recall some necessary definitions from [6].

A *family of functions* is a map $F\colon \mathrm{Keys}(F) \times \mathrm{Dom}(F) \to \mathrm{Ran}(F)$. Here $\mathrm{Keys}(F)$ is the key space of F; $\mathrm{Dom}(F)$ is the domain of F; and $\mathrm{Ran}(F)$ is the range of F. The two-input function F takes a key $K \in \mathrm{Keys}(F)$ and an input $x \in \mathrm{Dom}(F)$ to return a point $F(K,x) \in \mathrm{Ran}(F)$. If $\mathrm{Keys}(F) = \{0,1\}^k$ for an integer k then we refer to k as the key-length. If $\mathrm{Dom}(F) = \{0,1\}^l$ for some integer l then we refer to l as the input-length. If $\mathrm{Ran}(F) = \{0,1\}^L$ for some integer L then we refer to L as the output-length. In this paper, $\mathrm{Keys}(F)$, $\mathrm{Dom}(F)$, and $\mathrm{Ran}(F)$ will always be finite. For each key $K \in \mathrm{Keys}(F)$ we define the map $F_K\colon \mathrm{Dom}(F) \to \mathrm{Ran}(F)$ by $F_K(x) = F(K,x)$ for all $x \in \mathrm{Dom}(F)$. Thus, F specifies a collection of maps from $\mathrm{Dom}(F)$ to $\mathrm{Ran}(F)$, each map being associated with a key. That is why F is called a family of functions. We refer to F_K as an *instance* of F. We often speak of choosing a random key K uniformly from $\mathrm{Keys}(F)$. This operation is written $K \stackrel{R}{\leftarrow} \mathrm{Keys}(F)$. We write $f \stackrel{R}{\leftarrow} F$ for the operation $K \stackrel{R}{\leftarrow} \mathrm{Keys}(F)\ ;\ f \leftarrow F_K$. That is, $f \stackrel{R}{\leftarrow} F$ denotes the operation of selecting at random a function from the family F. When f is so selected it is called a *random instance* of F. We say that F is a family of permutations if $\mathrm{Dom}(F) = \mathrm{Ran}(F)$, and for each key $K \in \mathrm{Keys}(F)$ it is the case that F_K is a permutation (ie. a bijection) on $\mathrm{Dom}(F)$.

In order to define PRFs and PRPs we first need to fix two function families. One is $\mathrm{Rand}^{l \to L}$, the family of all functions from $\{0,1\}^l$ to $\{0,1\}^L$, and the other is Perm^l, the family of all permutations on $\{0,1\}^l$.

Definition 5 [PRF and PRP families, [6]] Let F be a function family with input-length l and output-length L and P be a permutation family with length l. Let $b \in \{0,1\}$. Let $D_{\mathrm{fn}}, D_{\mathrm{pn}}$ be distinguishers that have access to the oracle $\mathcal{O}_b(\cdot)$. Now, we consider the following experiments:

Experiment $\mathbf{Exp}^{\mathrm{prf}\text{-}b}_{F,D_{\mathrm{fn}}}$	Experiment $\mathbf{Exp}^{\mathrm{prp}\text{-}b}_{P,D_{\mathrm{pn}}}$
$\mathcal{O}_0 \stackrel{R}{\leftarrow} \mathrm{Rand}^{l \to L};\ \mathcal{O}_1 \stackrel{R}{\leftarrow} F$	$\mathcal{O}_0 \stackrel{R}{\leftarrow} \mathsf{Perm}^l;\ \mathcal{O}_1 \stackrel{R}{\leftarrow} P$
$d \leftarrow D_{\mathrm{fn}}^{\mathcal{O}_b(\cdot)}$	$d \leftarrow D_{\mathrm{pn}}^{\mathcal{O}_b(\cdot)}$
Return d	**Return** d

We define the *advantages* of the distinguishers via

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{F,D_{\mathrm{fn}}} &= \Pr[\,\mathbf{Exp}^{\mathrm{prf}\text{-}1}_{F,D_{\mathrm{fn}}} = 1\,] - \Pr[\,\mathbf{Exp}^{\mathrm{prf}\text{-}0}_{F,D_{\mathrm{fn}}} = 1\,] \\ \mathbf{Adv}^{\mathrm{prp}}_{P,D_{\mathrm{pn}}} &= \Pr[\,\mathbf{Exp}^{\mathrm{prp}\text{-}1}_{P,D_{\mathrm{pn}}} = 1\,] - \Pr[\,\mathbf{Exp}^{\mathrm{prp}\text{-}0}_{P,D_{\mathrm{pn}}} = 1\,]\,.\end{aligned}$$

We define the *advantage functions of the function family* as follows. For any integers t, q,

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{F}(t,q) &= \max_{D_{\mathrm{fn}}}\{\mathbf{Adv}^{\mathrm{prf}}_{F,D_{\mathrm{fn}}}\} \\ \mathbf{Adv}^{\mathrm{prp}}_{P}(t,q) &= \max_{D_{\mathrm{pn}}}\{\mathbf{Adv}^{\mathrm{prp}}_{P,D_{\mathrm{pn}}}\}\end{aligned}$$

where the maximum is over all $D_{\mathrm{fn}}, D_{\mathrm{pn}}$ with time complexity t, each making at most q queries to the oracle. ▮

Notice that since we are talking about finite families F, P, there is no fixed or formal notion of a "secure" PRF or PRP family. Every family has some associated insecurity as a PRF or PRP family. We use the terminology "F is a secure PRF" only in informal discussions, to indicate that $\mathbf{Adv}^{\mathrm{prf}}_{F}(t,q)$ is "low" for "reasonable" values of t, q. Notice also that unlike Luby and Rackoff [21], we measure the quality of a PRP family with respect to the family of random permutations, not random functions. This is motivated by the fact that PRPs, as we define them, are better models for block ciphers, than PRFs. (Of course, the distinction is only in the concrete security, but that is indeed our concern.) Nonetheless, the following relation between the two notions is often enough:

Proposition 8 [PRPs are PRFs] *For any permutation family P with length l,*

$$\mathbf{Adv}^{\mathrm{prf}}_{P}(t,q) \;\le\; \mathbf{Adv}^{\mathrm{prp}}_{P}(t,q) + q^2 2^{-l-1}\,.\ ▮$$

A block cipher is a (finite) family of permutations. For example, DES is a family of permutations with Keys(DES) $= \{0,1\}^{56}$ and Dom(DES) $=$ Ran(DES) $= \{0,1\}^{64}$. The estimated cryptanalytic strength of the block ciphers gives us values of t, q for which the block cipher may be viewed as a PRP family. Using Proposition 8 we get the bounds by which it can be viewed as a PRF family.

5 Analysis of the XOR and CTR Schemes

Fix a function family F with input-length l, output-length L, and key-length k. We let K denote the key shared between the two parties who run the encryption scheme. It will be used to specify the function $f = F_K$. In fact, all the schemes depend only only on this function, in the sense that they can be implemented given an oracle for the function.

The CTR scheme is stateful (counter based and deterministic). The XOR scheme is a stateless (randomized) variant of CTR.

SPECIFICATIONS. The scheme $\mathrm{XOR}[F] = (\mathcal{E}\text{-XOR}, \mathcal{D}\text{-XOR}, \mathcal{K}\text{-XOR})$ works as follows. The key generation algorithm $\mathcal{K}$-XOR just outputs a random k-bit key K for the underlying PRF family F, thereby specifying a function $f = F_K$ of l-bits to L-bits. The message x to be encrypted is regarded as a sequence of L-bit blocks (padding is done first, if necessary), $x = x_1 \cdots x_n$. We define $\mathcal{E}\text{-XOR}_K(x) = \mathcal{E}\text{-XOR}^{F_K}(x)$ and $\mathcal{D}\text{-XOR}_K(z) = \mathcal{D}\text{-XOR}^{F_K}(z)$, where:

function $\mathcal{E}\text{-XOR}^f(x)$	**function** $\mathcal{D}\text{-XOR}^f(z)$
$\quad r \leftarrow \{0,1\}^l$	$\quad$ Parse z as $r \,\|\, y_1 \cdots y_n$
$\quad$ **for** $i = 1, \ldots, n$ **do** $y_i = f(r+i) \oplus x_i$	$\quad$ **for** $i = 1, \ldots, n$ **do** $x_i = f(r+i) \oplus y_i$
$\quad$ **return** $r \,\|\, y_1 y_2 \cdots y_n$	$\quad$ **return** $x = x_1 \cdots x_n$

We call r the *nonce*. Addition, above, is modulo 2^l, and the result is encoded as an l-bit string in the usual way.

This scheme's stateful variant is $\mathrm{CTR}[F] = (\mathcal{E}\text{-CTR}, \mathcal{D}\text{-CTR}, \mathcal{K}\text{-CTR})$. Here the role of r is played by a l-bit counter, denoted ctr, that is initially -1 and increases after each encryption by the number of encrypted blocks. Note only the sender maintains the counter and it is output as part of the ciphertext. A restriction placed on the scheme is that the total number of encrypted blocks be at most 2^l.

The key generation algorithm $\mathcal{K}$-CTR is the same as before, meaning just outputs a random key K for the PRF family. With the same formatting conventions as above, we define $\mathcal{E}\text{-CTR}_K(x, ctr) = \mathcal{E}\text{-CTR}^{F_K}(x, ctr)$ and $\mathcal{D}\text{-CTR}_K(z) = \mathcal{D}\text{-CTR}^{F_K}(z)$, where:

function $\mathcal{E}\text{-CTR}^f(x, ctr)$	**function** $\mathcal{D}\text{-CTR}^f(z)$
for $i = 1, \ldots, n$ **do** $y_i = f(ctr + i) \oplus x_i$	Parse z as $ctr \,\|\, y_1 \cdots y_n$
$ctr \leftarrow ctr + n$	**for** $i = 1, \ldots, n$ **do** $x_i = f(ctr + i) \oplus y_i$
return $(ctr,\ ctr \,\|\, y_1 y_2 \cdots y_n)$	**return** $x = x_1 \cdots x_n$

FEATURES OF THE SCHEMES. Notice that decryption does not require the ability to invert $f = F_K$. Thus F_K need not be a permutation.

The XOR and CTR schemes have some computational advantages over the more common modes of operation. Namely, the F_K computations on different blocks can be done in parallel since the computation on a block is independent of the other blocks. This parallelizability advantage can be realized through either hardware or software support. Decryption does not have to be done in order if each block is tagged with its index. These schemes also support off-line processing, in the sense that the F_K computations can be done during idle times before the messages they are to be used with become available.

SECURITY OF XOR. We give bounds on the advantage function for the $\mathrm{XOR}[F]$ scheme, assuming F is a finite PRF family. We drop the security parameter k in our notation since there are no asymptotics present in this case. This convention is followed for all other schemes we study in this work too.

We first derive a lower bound on the success of an adversary trying to break the $\mathrm{XOR}[F]$ scheme in the LOR-CPA sense. In the common cryptographic terminology, this means, simply, that we are providing an attack. The attack we specify is on the "ideal" scheme, $\mathrm{XOR}[\mathrm{Rand}^{l \to L}]$.

Proposition 9 [Lower bound on insecurity of XOR using a random function] *Suppose $R = \mathrm{Rand}^{l \to L}$. Then, for any q_e, μ_e, such that $\mu_e q_e / L \le 2^l$,*

$$\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{XOR}[R]}(\cdot, t, q_e, \mu_e) \quad \ge \quad 0.316 \cdot \frac{\mu_e \cdot (q_e - 1)}{L \cdot 2^l}. \ \blacksquare$$

This is a "birthday" attack. It may be easier to gauge if we let $\bar{n} = \mu_e/(Lq_e)$ be the average number of blocks per query, so that $\mu_e = Lq_e \cdot \bar{n}$. Then we see that the advantage function is $\Omega(q_e^2/2^l) \cdot \bar{n}$, a typical birthday behavior exhibiting a quadratic dependence on the number of queries.

Since we prove a lower bound in the random function model, we do not discuss the time complexity. However it is clear from the strategy that the time complexity would be just a little overhead besides the time for the oracle calls. This is true for all lower bounds and we do not mention it again. Proposition 9 indicates that even when the underlying block cipher F is very good (it cannot get better than truly random) the XOR scheme leaks some information as more and more data is encrypted. Next, we show that the above is essentially the best attack: one cannot get a better advantage, up to a constant factor.

Lemma 10 [Upper bound on insecurity of XOR **using a random function]** *Suppose* $R = \mathrm{Rand}^{l \to L}$. *Then, for any* t, q_e, μ_e,

$$\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{XOR}[R]}(\cdot, t, q_e, \mu_e) \quad \leq \quad \frac{\mu_e \cdot (q_e - 1)}{L \cdot 2^l}. \blacksquare$$

Of course, an indication of security in the ideal model is not an indication of security when we use a block cipher. The "real-world" case however is easily derived from the above:

Theorem 11 [Security of XOR **using a pseudorandom function]** *Suppose* F *is a PRF family with input-length* l *and output-length* L. *Then, for any* t, q_e *and* $\mu_e = q'L$,

$$\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{XOR}[F]}(\cdot, t, q_e, \mu_e) \quad \leq \quad 2 \cdot \mathbf{Adv}^{\text{prf}}_{F}(t, q') + \frac{\mu_e \cdot (q_e - 1)}{L \cdot 2^l}. \blacksquare$$

Security of CTR. The stateful version of the scheme has better security. The adversary has *no* advantage in the ideal case:

Lemma 12 [Security of CTR **using a random function]** *Suppose* $R = \mathrm{Rand}^{l \to L}$. *Then, for any* t, q_e *and* $\mu_e \leq L2^l$,

$$\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{CTR}[R]}(\cdot, t, q_e, \mu_e) \quad = \quad 0. \blacksquare$$

This translates into the following "real-world" security:

Theorem 13 [Security of CTR **using a pseudorandom function]** *Suppose* F *is a PRF family with input-length* l *and output-length* L. *Then, for any* t, q_e *and* $\mu_e = \min(q'L, L2^l)$,

$$\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{CTR}[F]}(\cdot, t, q_e, \mu_e) \quad \leq \quad 2 \cdot \mathbf{Adv}^{\text{prf}}_{F}(t, q'). \blacksquare$$

Proofs. The following will be useful in various estimates:

Fact 14 For any real number x with $0 \leq x \leq 1$ we have $(1 - e^{-1})x \leq 1 - e^{-x} \leq x$

We use throughout the following notation. If x is a string of length a multiple of L we view it as a sequence of L bit blocks. We let $n = |x|_L$ denote the number of blocks and $x[i]$ denote the i-th block, so that $x = x[1] \ldots x[n]$. For an integer m let $[m] = \{1, \ldots, m\}$. In the proofs, we let q denote q_e and μ denote μ_e.

Proof of Proposition 9: The proof of this is by construction of an adversary that achieves the given security parameters. Recall that an adversary in the LOR-CPA sense makes encryption oracle queries consisting of pairs of messages, trying to tell whether the left or right half of the pair is being encrypted. Our adversary A looks for a collision in the inputs to the random function f underlying the scheme.

Algorithm $A^{\mathcal{O}(\cdot,\cdot)}(k)$

(1) Let $n = \mu/(Lq)$. (This will be the number of blocks in all queried messages.)

(2) Choose messages $N_1, \ldots, N_q$, all n blocks long, such that $N_i[k] \neq N_j[k']$ for all $i, j = 1, \ldots, q$ and $k, k' = 1, \ldots, n$ satisfying $(i, k) \neq (j, k')$. (For example, set $N_i[k]$ to the L-bit binary encoding of the integer $n(i-1) + k$ for $i = 1, \ldots, q$ and $k = 1, \ldots, n$.)

(3) For $i = 1, \ldots, q$ do: $(r_i, y_i[1] \ldots y_i[n]) \leftarrow \mathcal{O}(0^{nl}, N_i)$. We call r_i the i'th nonce.

(4) If there is some $i \neq j$ that $|r_i - r_j| < n$ (treat r_i, r_j as integers here!) then determine the values $k, k' \in \{1, \ldots, n\}$ such that $r_i + k = r_j + k'$. Output 1 if $y_i[k] = y_j[k']$ and 2 otherwise.

(5) If there is no $i \neq j$ that $|r_i - r_j| < n$, output a coin flip.

Let $\mathsf{OverlapNonce}$ be the event that for some $i \neq j$ we have $|r_i - r_j| < n$. Whenever this event occurs we say that there has been an *overlap of nonces*. We claim that the advantage of A is just the probability of $\mathsf{OverlapNonce}$. To see this, first observe that the probability of this event is the same in both games as it involves only the random nonce values. Let p be this probability. Let $\Pr_b[\, A = 1\,]$ be the probability that A declares that it is playing game 0 when it is playing game $b \in \{0, 1\}$. We have

$$\mathbf{Adv}^{\text{lor-cpa}}_{\text{XOR}[R],A}(\cdot) \;=\; \Pr_1[\, A = 1\,] - \Pr_0[\, A = 1\,] \;=\; \left(p \cdot 1 + (1-p) \cdot \frac{1}{2}\right) - \left(p \cdot 0 + (1-p) \cdot \frac{1}{2}\right) \;=\; p$$

Now we want to lower bound p. Let D_i be the event that there has not been an overlap of nonces up to and including the i'th query. We observe that for D_{i+1} to be true, the nonce of the $(i+1)$'th query must not overlap with any of the i nonces of the previous queries. In terms of values that the $(i+1)$'th nonce can assume, we note that there are at least in values that would cause an overlap of nonces. (In general there could be as many as $i(n-1)$ more such values, but we may ignore them for now since our interest is a lower bound on p.) We therefore have

$$\Pr[\, \mathsf{D}_{i+1} \mid \mathsf{D}_i \,] \;\leq\; \frac{2^l - in}{2^l} \;=\; 1 - \frac{in}{2^l}\,.$$

The probability of no overlap of nonces at the end of the q'th query can now be computed as follows

$$\Pr[\, \mathsf{D}_q \,] \;=\; \textstyle\prod_{i=1}^{q-1} \Pr[\, \mathsf{D}_{i+1} \mid \mathsf{D}_i \,] \;\leq\; \prod_{i=1}^{q-1} \left(1 - \dfrac{in}{2^l}\right) \;\leq\; \prod_{i=1}^{q-1} e^{-in/2^l} \;=\; e^{-nq(q-1)/2^{l+1}}\,.$$

The last inequality follows from Fact 14. Continuing,

$$p \;=\; \Pr[\, \mathsf{OverlapNonce} \,] \;=\; 1 - \Pr[\, \mathsf{D}_q \,] \;\geq\; 1 - e^{-nq(q-1)/2^{l+1}} \;=\; 1 - e^{-(1/2)\cdot\mu(q-1)/(L2^l)}\,.$$

We have assumed $\mu q/L \leq 2^l$. This means $x \stackrel{\text{def}}{=} \mu(q-1)/(L2^l) \leq 1$ and we can apply the inequality $1 - e^{-x} \geq (1 - e^{-1})x$ of Fact 14 to get

$$p \;\geq\; \left(1 - \frac{1}{e}\right) \cdot \frac{1}{2} \cdot \frac{\mu(q-1)}{L2^l}\,,$$

which proves the Proposition. ▮

Proof of Lemma 10: Let $(M_1, N_1), \ldots, (M_q, N_q)$ be the oracle queries of the adversary A, each consisting, by definition, of a pair of equal length messages. These queries are random variables that depend on the coin tosses of A and responses of the oracle to previous queries. Let $r_i \in \{0,1\}^l$ be the nonce associated to (M_i, N_i) as chosen at random by the oracle, for $i = 1, \ldots, q$. Let n_i be the number of blocks in the i'th query. In answering the i'th query, the oracle applies the underlying function f to the n_i strings $r_i + 1, \ldots, r_i + n_i \in \{0,1\}^l$. We call these strings the *i'th sequence*, and $r_i + k$ is the k-th point in this sequence, $k = 1, \ldots, n_i$.

Let D be the following event, defined for either game: $r_i + k \neq r_j + k'$ whenever $(i,k) \neq (j,k')$, for all $i, j = 1, \ldots, q$ and $k = 1, \ldots, n_i$ and $k' = 1, \ldots, n_j$. That is D is the event that no collision occurs in the inputs to the random function (or equivalently, that there are no overlapping sequences) among all of the queries. We also define $\Pr_0[\,\cdot\,]$ to be the probability of an event in game 0 and $\Pr_1[\,\cdot\,]$ of the event in game 1.

Claim 1. $\Pr_0\left[\,\overline{\mathsf{D}}\,\right] = \Pr_1\left[\,\overline{\mathsf{D}}\,\right]$

Proof: The event D for either game depends only on the nonce chosen for each query. The nonces themselves are chosen randomly and are thus independent of the game being played (or of the messages given to the oracle). $\square$

Claim 2. $\Pr_0[\,A=1 \mid \mathsf{D}\,] = \Pr_1[\,A=1 \mid \mathsf{D}\,]$

Proof: Given the event D, we have that, in either game, the function f is evaluated at a new point each time it is invoked, and thus the output is randomly and uniformly distributed over $\{0,1\}^L$, independently of anything else. Thus each cipher block is a message block XORed with a random value. A consequence of this is that each cipher block has a distribution that is independent of any previous cipher blocks and of the messages. $\square$

We now upper bound the advantage of A as follows:

$$\begin{aligned}
\mathbf{Adv}^{\text{lor-cpa}}_{\text{XOR}[R],A}(\cdot) &= \Pr_1[\,A=1\,] - \Pr_0[\,A=1\,] \\
&= \Pr_1\left[\,A=1 \mid \overline{\mathsf{D}}\,\right]\cdot\Pr_1\left[\overline{\mathsf{D}}\right] + \Pr_1[\,A=1 \mid \mathsf{D}\,]\cdot\Pr_1[\mathsf{D}] - \\
&\quad\ \Pr_0\left[\,A=1 \mid \overline{\mathsf{D}}\,\right]\cdot\Pr_0\left[\overline{\mathsf{D}}\right] - \Pr_0[\,A=1 \mid \mathsf{D}\,]\cdot\Pr_0[\mathsf{D}]
\end{aligned}$$

Using *Claim* 1 and *Claim* 2, we have,

$$\mathbf{Adv}^{\text{lor-cpa}}_{\text{XOR}[R],A}(\cdot) = \left(\Pr_1\left[\,A=1 \mid \overline{\mathsf{D}}\,\right] - \Pr_0\left[\,A=1 \mid \overline{\mathsf{D}}\,\right]\right)\cdot\Pr_1\left[\overline{\mathsf{D}}\right] \le \Pr_1\left[\overline{\mathsf{D}}\right]$$

Given Claim 1 we drop the subscript in talking about the probability of D and write the above just as $\Pr[\overline{\mathsf{D}}]$. Now we want to upper bound $\Pr[\overline{\mathsf{D}}]$. We observe that the chance of collision at the time of the choice of the i'th nonce is maximized if all the $i-1$ previous queries resulted in $i-1$ sequences of inputs to f that were no less than $n_i - 1$ blocks apart. We have a collision if the i'th sequence begins in a block that is $n_i - 1$ blocks before any other previous sequence j or in a block position occupied by that sequence j. Now let the probability of the i'th sequence colliding with any of the previous sequences be p_i. We then have, for $i > 1$

$$p_i \le \frac{\sum_{j=1}^{i-1}(n_j + n_i - 1)}{2^l} = \frac{(i-1)(n_i-1) + \sum_{j=1}^{i-1} n_j}{2^l}.$$

Thus

$$\Pr[\overline{\mathsf{D}}] \le \sum_{i=1}^{q} p_i \le \sum_{i=1}^{q} \frac{\left((i-1)(n_i-1) + \sum_{j=1}^{i-1} n_j\right)}{2^l} = \frac{\frac{\mu}{L}(q-1) - \frac{q(q-1)}{2}}{2^l} \le \frac{\mu(q-1)}{L\cdot 2^l}.$$

Putting everything together we have $\mathbf{Adv}^{\text{lor-cpa}}_{\text{XOR}[R],A}(\cdot) \le \frac{\mu(q-1)}{L\cdot 2^l}$. ▮

Proof of Theorem 11: Intuitively, Lemma 10 says the XOR$[R]$ is secure. If XOR$[F]$ were not secure, this would mean F is not good as a PRF function family. Formally we prove the theorem by a contradiction argument. Assume that A is an adversary attacking XOR$[F]$ in the LOR-CPA sense and having an advantage greater than $\mathbf{Adv}^{\text{lor-cpa}}_{\text{XOR}[F]}(k,t,q_e,\mu_e)$. We build a distinguisher D, using A, that has an advantage better than $\mathbf{Adv}^{\text{prf}}_{F}(t,q')$, for some reasonable values for q', contradicting the assumed security of F as a pseudorandom function family. Our distinguisher simply runs A and tries to see whether A breaks the encryption scheme. If so, it bets that f is drawn from F, else it bets that f is drawn from R. In order to run A it simulates its oracle $\mathcal{O}(\cdot,\cdot)$ via queries to its own oracle f by using the latter as the function underlying the encryption scheme. In more detail:

Algorithm $D^f(k)$

(1) $b \leftarrow \{0,1\}$. (This represents a choice to play either left or right oracle for A.)

(2) Run A, responding to its oracle queries as follows. When A makes an oracle query (M_1, M_2), let $z \leftarrow \mathcal{E}\text{-XOR}^f(M_b)$, and return z to A as the answer to the oracle query. (It is important here that D can implement the encryption function given an oracle for f.)

(3) Eventually A stops and outputs a guess d to indicate whether it thought its oracle was the left oracle or the right oracle. If $d = b$ then output 1, else output 0.

In responding to oracle query (M_1, M_2), distinguisher D makes n oracle queries to f, where $n = |M_1|/L = |M_2|/L$ is the number of blocks in the messages. So the total number of oracle queries made by D is at most μ/L, which by assumption is q'.

To compute $\mathbf{Adv}^{\mathrm{prf}}_{F,D}$ we first need some notation. For $G \in \{F, R\}$, let $\mathsf{Correct}(G)$ be the probability that A correctly identifies its oracle when the function underlying the encryption scheme is $f \leftarrow G$, One can check that $\mathsf{Correct}(G) = (1/2) \cdot [1 + \mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{XOR}(G),A}(\cdot)]$. Now note

$$\mathbf{Adv}^{\mathrm{prf}}_{F,D} \;=\; \mathsf{Correct}(F) - \mathsf{Correct}(R) \;=\; (1/2) \cdot \left[\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{XOR}[F],A}(\cdot) - \mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{XOR}[R],A}(\cdot)\right] .$$

Lemma 10 gives us a bound for $\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{XOR}[R],A}(\cdot)$. Using this, we see that to avoid a contradiction, we must bound the advantage function as stated in the theorem statement. ▌

Proof of Lemma 12: The proof is similar to that of Lemma 10. The difference is in that $\Pr_0\left[\overline{\mathsf{D}}\right] = \Pr_1\left[\overline{\mathsf{D}}\right] = 0$ for $\mu/L \leq 2^l$. This is because the counter will not repeat until 2^l blocks have been encrypted. ▌

Proof of Theorem 13: The proof is similar to that of Theorem 11 and is omitted. ▌

6 Analysis of the CBC Scheme

For the CBC scheme we require that $l = L$ (the input-length and output-length of F are the same) and that each F_K be a permutation such that given K we can compute not only F_K but also F_K^{-1}.

Specification. The scheme $\mathrm{CBC}[F] = (\mathcal{E}\text{-CBC}, \mathcal{D}\text{-CBC}, \mathcal{K}\text{-CBC})$ has the same key generation algorithm as the previous schemes, meaning the key for encryption is the key K specifying $f = F_K$. The message x to be encrypted is regarded as a sequence of l bit blocks, $x = x_1 \ldots x_n$. We define $\mathcal{E}\text{-CBC}_K(x) = \mathcal{E}\text{-CBC}^{F_K}(x)$ and $\mathcal{D}\text{-CBC}_K(z) = \mathcal{D}\text{-CBC}^{F_K}(z)$, where:

function $\mathcal{E}\text{-CBC}^f(x)$	**function** $\mathcal{D}\text{-CBC}^f(z)$
$\quad y_0 \leftarrow \{0,1\}^l$	$\quad$ Parse z as $y_0 \,\|\, y_1 \cdots y_n$
$\quad$ **for** $i = 1, \ldots, n$ **do** $y_i = f(y_{i-1} \oplus x_i)$	$\quad$ **for** $i = 1, \ldots, n$ **do** $x_i = f^{-1}(y_i) \oplus y_{i-1}$
$\quad$ **return** $y_0 \,\|\, y_1 y_2 \cdots y_n$	$\quad$ **return** $x = x_1 \ldots x_n$

The value y_0 is called *initial vector*, or *nonce*. See discussion below for the counter variant.

Features of the scheme. We have already mentioned the computational advantages of the XOR and CTR schemes over the CBC scheme. The CBC scheme, however, has superior error-propogation and erro-recovery properties to these other schemes. CBC is self-synchronizing, in that the corruption or even loss of a few ciphertext blocks prevents the correct decryption of only a few of the plaintext blocks, without requiring explicit re-synchronization.

Security of CBC. The $\mathrm{CBC}[F]$ scheme should be analyzed assuming F is a PRP family, not a PRF family, because the scheme must indeed be used with permutations, not functions. However, the analysis is significantly simpler using functions, rather than permutations. Hence, our approach will be the following. For the upper bound (on the insecurity of CBC[F]), we first analyze CBC[$\text{Rand}^{l\to l}$] (ie. the scheme using random functions). Then, similar to Theorem 11, we derive the security of CBC[F], assuming F is a PRF family. Finally, using Proposition 8, we translate this to the security when F is viewed as a PRP family. For the lower bound, however, this approach does not work. Hence we derive this directly.

Proposition 15 [Lower bound on insecurity of CBC **using a random permutation]** *Suppose* $R = \mathsf{Perm}^l$*. Then, for* $\mu_e \le l \cdot 2^{\frac{l}{2}}$ *and* $q_e = \mu_e/l$*,*

$$\mathbf{Adv}^{\text{lor-cpa}}_{\text{CBC}[R]}(\cdot, t, q_e, \mu_e) \quad \geq \quad 0.316 \cdot \left(\frac{\mu_e^2}{l^2} - \frac{\mu_e}{l}\right) \cdot \frac{1}{2^l} \ . \ \blacksquare$$

Next, we show that this is the best possible attack up to a constant factor.

Lemma 16 [Upper bound on insecurity of CBC **using a random function]** *Suppose* $R = \text{Rand}^{l\to l}$*. Then, for any* t, q_e, μ_e*,*

$$\mathbf{Adv}^{\text{lor-cpa}}_{\text{CBC}[R]}(\cdot, t, q_e, \mu_e) \quad \leq \quad \left(\frac{\mu_e^2}{l^2} - \frac{\mu_e}{l}\right) \cdot \frac{1}{2^l} \ . \ \blacksquare$$

The "real-world" security follows:

Theorem 17 [Security of CBC **using a pseudorandom permutation]** *Suppose* F *is a PRP family with length* l*. Then, for any* t, q_e *and* $\mu_e = ql$*,*

$$\mathbf{Adv}^{\text{lor-cpa}}_{\text{CBC}[F]}(\cdot, t, q_e, \mu_e) \quad \leq \quad 2 \cdot \mathbf{Adv}^{\text{prp}}_{F}(t, q) + q^2 2^{-l-1} + \left(\frac{\mu_e^2}{l^2} - \frac{\mu_e}{l}\right) \cdot 2^{-l} \ . \ \blacksquare$$

CBC WITH COUNTERS. It is tempting to make a counter variant of CBC and hope that the security is increased (or at least preserved). Indeed it is suggested in various books that the initialization vector may be a counter. But this does not work; knowing the next value of the counter, the adversary can choose a message query that forces a collision in the inputs to f, thus breaking the scheme (under any of the definitions).

To make a proper counter version of CBC, one can let the initialization vector be $y_0 = f(ctr)$ and increment ctr by one following every encryption. The scheme is capable of encrypting at most 2^l messages. An analog to Theorem 17 is then possible. The result is easiest (following as a corollary to Theorem 17 if the key used to determine y_0 is separate from the key used for the rest of the CBC encryption.

PROOFS. We begin with the attack on CBC[Perm^l].

Proof of Proposition 15: The idea is that it suffices to find collisions in the initial vectors (nonces). The details follow.

The adversary sets $q = \mu/l$. It then sets $M_i = 0^l$ for $i = 1, \ldots, q$ and chooses $N_1, \ldots, N_q$ to be distinct, non-zero l-bit strings. It makes q queries, consisting of the pairs of messages $(M_1, M_1'), \ldots, (M_q, M_q')$. Let $C_i[0]C_i[1]$ denote the response to the i-th query. If $C_1[0], \ldots, C_q[0]$ are all distinct the adversary flips a coin to determine its output. Else, let $i \neq j$ be such that $C_i[0] = C_j[0]$. The adversary outputs 1 if $C_i[1] = C_j[1]$ and 2 otherwise. It is easy to see that the advantage is exactly the chance that there is a collision in the initial vectors. We use the lower bound from Fact 14 to bound this advantage.

Note that in the attack above, given μ_e, we allow the adversary to choose a convenient q_e. This turns out to be $q_e = \mu_e/l$. It is possible to prove something stronger, namely that an attack could be mounted for any given value of q_e. The proof of this, again is by construction of an adversary that achieves the given security parameters. Our adversary A looks for a collision in the inputs to the random function f underlying the scheme.

Algorithm $A^{\mathcal{O}(\cdot,\cdot)}$

(1) Let $n = \mu/(lq)$. (This will be the number of blocks in all queried messages.) Let $T = [q] \times [n]$.

(2) Choose messages $M_1, \ldots, M_q$, all n blocks long, such that $M_i[k] \neq M_j[k']$ for all distinct $(i,k),(j,k') \in T$. (For example, set $M_i[k]$ to the l-bit binary encoding of the integer $n(i-1)+k$ for all $(i,k) \in T$.) Also set $M'_i[k] = 0^l$ and $M'_i = M'_i[1] \ldots M'_i[n]$ for all $(i,k) \in T$.

(3) For $i = 1, \ldots, q$ do: $(C_i[0], C_i[1] \ldots C_i[n]) \leftarrow \mathcal{O}(M_i, M'_i)$. We call $C_i[0]$ the i'th initial vector.

(4) If D is true then output a coin flip and halt. Else (meaning D is false) go on with the rest of the algorithm below.

(5) Let $(j,k) \in T$ be the least pair for which $\mathsf{D}_{j,k}$ is false. (Meaning if $\mathsf{D}_{j',k'}$ is false for some other pair $(j',k') \in T$ then $(j,k) \prec (j',k')$.)

(6) If there exists $(j',k') \prec (j,k)$ such that $C_j[k-1] \oplus M_j[k] = C_{j'}[k'-1] \oplus M_{j'}[k']$ then set $b_0 = 1$ and test if $C_j[k] = C_{j'}[k']$. If the test passes then set $a_0 = 1$ else set $a_0 = 0$. Otherwise set $b_0 = 0$.

(7) If there exists $(j',k') \prec (j,k)$ such that $C_j[k-1] = C_{j'}[k'-1]$, then set $b_1 = 1$ and test if $C_j[k] = C_{j'}[k']$. If the test passes then set $a_1 = 1$ else set $a_1 = 0$. Otherwise set $b_1 = 0$.

(8) If $b_1 = 1$ then: if $a_1 = 1$ then output 1, else output 0.

(9) Else (meaning $b_1 = 0$) it must be that $b_0 = 1$. Then if $a_0 = 1$ then output 0, else output 1.

We omit details of the analysis of this attack, noting that same bound derived for the attack allowing the choice of a convenient q_e holds in this case too. ▮

We next give a lemma that will be useful in proving Lemma 16.

Consider an arbitrary adversary A, attacking CBC$[R]$ (where $R = \text{Rand}^{l \to l}$) in the LOR-CPA sense. It makes up to q queries to its oracle $\mathcal{O}(\cdot,\cdot)$, totaling at most μ bits. Let $(M_1, M'_1), \ldots, (M_q, M'_q)$ be the oracle queries of the adversary A, each consisting, by definition, of a pair of equal length messages. These queries are random variables that depend on the coin tosses of A and responses of the oracle to previous queries. Let $n_i = |M_i|_l = |M'_i|_l$ be the number of blocks in a message in the i-th query, $i = 1, \ldots, q$. Let $C_i = C_i[0] \ldots C_i[n_i]$ be the random variable which is the response of the oracle to query (M_i, M'_i), for $i = 1, \ldots, q$.

Some notation will be useful. Let $T = \{ (j,k) \ : \ j \in [q] \text{ and } k \in [n_j] \}$ and $T' = \{ (j,k) \ : \ j \in [q] \text{ and } k = 0, \ldots, n_j \}$ and $T'' = \{ (j,k) \ : \ j \in [q] \text{ and } k = 0, \ldots, n_j + 1 \}$. We put an order $\prec$ on T'' defined as follows:

$$(j,k) \prec (j',k') \quad \text{if} \quad \left(\textstyle\sum_{i=1}^{j-1}(n_i+2)\right) + k \ < \ \left(\textstyle\sum_{i=1}^{j'-1}(n_i+2)\right) + k' \ ,$$

for any $(j,k),(j',k') \in T''$. We write $(j,k) \preceq (j',k')$ if either $(j,k) \prec (j',k')$ or $(j,k) = (j',k')$. Of course, the order inherits to any subset of T'' and we will most often use it on T or T'.

We let $\Pr_b[\,\cdot\,]$ denote the probability distribution in Game $b \in \{0,1\}$, where Game b is the one where $\mathcal{O}(\cdot,\cdot) = \mathcal{E}\text{-CBC}^f(\mathcal{LR}(\cdot,\cdot,b))$, with $f \leftarrow R$.) We know that $C_j[k] = C_j[k-1] \oplus M_j[k]$ in Game 0 and $C_j[k] = C_j[k-1] \oplus M'_j[k]$ in Game 1, for all $j \in [q]$ and $k \in [n_j]$. The following defines an event, for either game, that says there are no collisions in the inputs to f, in either game, upto the indicated point.

Definition 6 [Event Distinct] In the above setting, with adversary A fixed, define the event $\mathsf{D}_{i,u}$ (called *distinct*), for $i \in [q]$ and $u \in [n_i]$, to be true if

$$C_j[k-1] \oplus M_j[k] \neq C_{j'}[k'-1] \oplus M_{j'}[k'] \text{ and } C_j[k-1] \oplus M'_j[k] \neq C_{j'}[k'-1] \oplus M'_{j'}[k']$$
$$\text{for all } (j,k),(j',k') \in T \text{ satisfying } (j',k') \prec (j,k) \preceq (i,u).$$

Let $\mathsf{D} \equiv \mathsf{D}_{q,n_q}$. Also let $\mathsf{D}_{1,0}$ be an event that is always true and $\mathsf{D}_{i,0} \equiv \mathsf{D}_{i-1,n_{i-1}}$ for $i \geq 2$. Finally let $\mathsf{D}_{i,n_i+1} \equiv \mathsf{D}_{i,n_i}$ for $i \in [q]$.

It turns out the probability of D tells us pretty much all we want to know about the advantage of the adversary.

Lemma 18 [Main CBC lemma] *Let A be an adversary for* $\mathrm{CBC}[R]$ *in the setting above. Then*

(1) $\Pr_0\left[\overline{\mathsf{D}}\right] = \Pr_1\left[\overline{\mathsf{D}}\right]$.

Furthermore, letting p be the (common) value of this probability, we have

(2) $\frac{1}{2}\left(1-\frac{1}{e}\right)\cdot\left(\frac{\mu^2}{l^2}-\frac{\mu}{l}\right)\cdot\frac{1}{2^l} \;\leq\; p \;\leq\; \left(\frac{\mu^2}{l^2}-\frac{\mu}{l}\right)\cdot\frac{1}{2^l}$, *and*

(3) $\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{CBC}[R],A}(\cdot) = \left(\Pr_1\left[\,A=1 \mid \overline{\mathsf{D}}\,\right] - \Pr_0\left[\,A=1 \mid \overline{\mathsf{D}}\,\right]\right)\cdot p.$

We first prove our results given the Main CBC lemma and then return to the proof of the lemma.

Proof of Lemma 16: From Lemma 18 (3) we have

$$\mathbf{Adv}^{\text{lor-cpa}}_{\mathrm{CBC}[R],A}(\cdot) \;=\; \left(\Pr_1\left[\,A=1 \mid \overline{\mathsf{D}}\,\right] - \Pr_0\left[\,A=1 \mid \overline{\mathsf{D}}\,\right]\right)\cdot p \;\leq\; p\,.$$

Now apply the upper bound of Lemma 18 (2). ▮

Proof of Theorem 17: The proof is similar to the one given for Theorem 11. The addition here is that once we get the security assuming F to be a PRF family, we must use Proposition 8 to transate this to security assuming F to be a PRP family. ▮

Proof of Lemma 18: For $i \in [q]$ and $u \in \{0,\ldots,n_i\}$ let $C_{i,u} = (C_j[k] \,:\, (j,k) \in T' \text{ and } (j,k) \preceq (i,u))$ be the sequence of all ciphertext blocks upto and including $C_i[u]$.

Let $c_j[k]$ be an l-bit string for $j \in [q]$ and $k \in \{0,\ldots,n_j\}$. For $i \in [q]$ and $u \in \{0,\ldots,n_i\}$ let $c_{i,u} = (c_j[k] \,:\, (j,k) \in T' \text{ and } (j,k) \preceq (i,u))$ be the sequence of all strings "below" and including $c_i[u]$.

For $(i,u) \in T$ we define the set $\mathsf{Proh}_{i,u}(c_{q,n_q})$, for the fixed set of cipher blocks c_{q,n_q}, to consist of all of the following l bit strings:

(1) $c_j[k-1] \oplus M_j[k] \oplus M_i[u]$ for all $(j,k) \in T$ such that $(j,k) \prec (i,u)$

(2) $c_j[k-1] \oplus M'_j[k] \oplus M'_i[u]$ for all $(j,k) \in T$ such that $(j,k) \prec (i,u)$

That is $\mathsf{Proh}_{i,u}(c_{q,n_q})$ is the set of values that $C_i[u-1]$ may take which cause $\overline{\mathsf{D}_{i,u}}$ given that we had $C_j[k] = c_j[k]$ for all $(j,k) \prec (i,u-1)$.

We observe from the definition of $\mathsf{Proh}_{i,u}(c_{q,n_q})$ that

$$(n_1 + \cdots + n_{i-1} + u - 1) \;\leq\; \left|\mathsf{Proh}_{i,u}(c_{q,n_q})\right| \;\leq\; 2\cdot(n_1 + \cdots + n_{i-1} + u - 1)\,. \qquad (1)$$

We note that we have calculated bounds on the cardinality of $\mathsf{Proh}_{i,u}(c_{q,n_q})$. In general the size of the set could be something in between.

Remember that the difference between the games is that in Game 0 we have $C_j[k] = C_j[k-1] \oplus M_j[k]$ and in Game 1 we have $C_j[k] = C_j[k-1] \oplus M'_j[k]$, for all $j \in [q]$ and $k \in [n_j]$. Our first claim is that the probability distributions conditioned on D are nonetheless equal.

Claim 1: Let c_{q,n_q} be a fixed sequence of ciphertext blocks as above. Then

$$\Pr_0 [\, C_{i,u-1} = c_{i,u-1} \mid \mathsf{D}_{i,u} \,] \;=\; \Pr_1 [\, C_{i,u-1} = c_{i,u-1} \mid \mathsf{D}_{i,u} \,] \tag{2}$$

for all $i \in [q]$ and $u \in [n_i + 1]$.

Proof: By induction. The base case is $(i, u) = (1, 1)$. Here $C_{1,0} = C_1[0]$ is uniformly distributed since it is the randomly chosen initial vector, so the claim holds.

Now suppose $(1, 1) \prec (i, u)$. The inductive hypothesis is that

$$\Pr_0 [\, C_{j,k-1} = c_{j,k-1} \mid \mathsf{D}_{j,k} \,] \;=\; \Pr_1 [\, C_{j,k-1} = c_{j,k-1} \mid \mathsf{D}_{j,k} \,]$$

for all $(j, k) \prec (i, u)$ with $j \in [q]$ and $k \in [n_j]$.

Let $\Pr'_b [\,\cdot\,] = \Pr_b [\,\cdot \mid \mathsf{D}_{i,u}\,]$, for $b = 0, 1$. We consider two cases.

First suppose $u \geq 2$, so that $u \in \{2, \ldots, n_i + 1\}$. Then

$$\Pr'_b [\, C_{i,u-1} = c_{i,u-1} \,] \;=\; \Pr'_b [\, C_i[u-1] = c_i[u-1] \mid C_{i,u-2} = c_{i,u-2} \,] \cdot \Pr'_b [\, C_{i,u-2} = c_{i,u-2} \,] \,. \tag{3}$$

We take the two terms one by one and show each is independent of b. (The arguments justifying the claims are slightly different in the cases $u \leq n_i$ and $u = n_i + 1$, but the claims are true in both cases.) Begin with the second. We are conditioning on $\mathsf{D}_{i,u}$. It would make no difference, for this term, to condition on $\mathsf{D}_{i,u-1}$ since the quantities in the probability expression do not involve $C_{i,u-1}$ or $c_{i,u-1}$. That is,

$$\Pr'_b [\, C_{i,u-2} = c_{i,u-2} \,] \;=\; \Pr_b [\, C_{i,u-2} = c_{i,u-2} \mid \mathsf{D}_{i,u-1} \,] \,.$$

Now by the induction hypothesis this term is independent of b.

For the first term of the right hand side of Equation (3), observe

$$\Pr'_b [\, C_i[u-1] = c_i[u-1] \mid C_{i,u-2} = c_{i,u-2} \,] \;=\; \begin{cases} 0 & \text{if } c_i[u-2] \in \mathsf{Proh}_{i,u-1}(c_{q,n_q}) \\ 2^{-l} & \text{otherwise.} \end{cases} \tag{4}$$

We see Equation (4) like this. The first case (the probability of 0) is true because we have conditioned on $\mathsf{D}_{i,u}$ which exactly prohibits the event in question. For the second case, note $C_i[u-1] = f(C_i[u-2] \oplus M_i[u-1])$ in Game 0 and $C_i[u-1] = f(C_i[u-2] \oplus M'_i[u-1])$ in Game 1. However, *both* $C_i[u-2] \oplus M_i[u-1]$ and $C_i[u-2] \oplus M'_i[u-1]$ are points on which f has not been invoked before, regardless of which game is being played, if we know that $c_i[u-2]$ is not in the prohibited set. Thus the probability in question is as claimed and in particular independent of b. We have thus completed the proof that the quantity in Equation (3) is independent of b.

Now we have to deal with the case $u = 1$, namely show

$$\Pr_0 [\, C_{i,0} = c_{i,0} \mid \mathsf{D}_{i,1} \,] \;=\; \Pr_1 [\, C_{i,0} = c_{i,0} \mid \mathsf{D}_{i,1} \,] \,. \tag{5}$$

We can assume $i \geq 2$ since the case $(i, u) = (1, 1)$ was covered in the base case of the induction. We have

$$\Pr'_b [\, C_{i,0} = c_{i,0} \,] \;=\; \Pr'_b [\, C_i[0] = c_i[0] \mid C_{i-1,n_{i-1}} = c_{i-1,n_{i-1}} \,] \cdot \Pr'_b [\, C_{i-1,n_{i-1}} = c_{i-1,n_{i-1}} \,] \,. \tag{6}$$

The first term is 2^{-l} since $C_i[0]$ is the random initial vector. For the second term, we could condition on $D_{i-1,n_{i-1}+1}$ rather than $D_{i,1}$ without changing the outcome. Then we can apply the induction hypothesis to see that the term in question is independent of b. □

Claim 2. $\Pr_0 [\, A = 1 \mid \mathsf{D} \,] = \Pr_1 [\, A = 1 \mid \mathsf{D} \,]$.

Proof: This follows from Claim 1. □

The following is the first claim in the statement of Lemma 18.

Claim 3. $\Pr_0\left[\overline{\mathsf{D}}\right] = \Pr_1\left[\overline{\mathsf{D}}\right]$.

Proof: We will show by induction that for each $(i,u) \in T$ we have

$$\Pr_1\left[\overline{\mathsf{D}_{i,u}}\right] \;=\; \Pr_2\left[\overline{\mathsf{D}_{i,u}}\right] .$$

Clearly when $(i,u) = (1,1)$ both probabilities are one, so suppose $(1,1) \prec (i,u) \in T$. Assume inductively that $\Pr_0\left[\overline{\mathsf{D}_{j,k}}\right] = \Pr_1\left[\overline{\mathsf{D}_{j,k}}\right]$ for all $(j,k) \prec (i,u)$. For any $b = 0,1$,

$$\Pr_b\left[\overline{\mathsf{D}_{i,u}}\right] \;=\; \Pr_b\left[\overline{\mathsf{D}_{i,u}} \;\middle|\; \overline{\mathsf{D}_{i,u-1}}\right] \cdot \Pr_b\left[\overline{\mathsf{D}_{i,u-1}}\right] + \Pr_b\left[\overline{\mathsf{D}_{i,u}} \;\middle|\; \mathsf{D}_{i,u-1}\right] \cdot \Pr_b\left[\mathsf{D}_{i,u-1}\right] .$$

In the first term of the sum, the first term is 1 and the second term is by induction independent of b. In the second term of the sum, the second term is by induction independent of b. It remains to show that

$$\Pr_0\left[\overline{\mathsf{D}_{i,u}} \;\middle|\; \mathsf{D}_{i,u-1}\right] \;=\; \Pr_1\left[\overline{\mathsf{D}_{i,u}} \;\middle|\; \mathsf{D}_{i,u-1}\right] . \tag{7}$$

We break the proof of Equation (7) into two cases.

First suppose $u \geq 2$. Write

$$\Pr_b\left[\overline{\mathsf{D}_{i,u}} \;\middle|\; \mathsf{D}_{i,u-1}\right] \;=\;$$
$$\sum_{c_{i,u-2}} \Pr_b\left[\overline{\mathsf{D}_{i,u}} \;\middle|\; \mathsf{D}_{i,u-1} \wedge C_{i,u-2} = c_{i,u-2}\right] \cdot \Pr_b\left[\, C_{i,u-2} = c_{i,u-2} \;\middle|\; \mathsf{D}_{i,u-1}\,\right] .$$

We claim that each term in the sum is independent of b. To see this fix c_{q,n_q} and consider the term

$$\Pr_b\left[\overline{\mathsf{D}_{i,u}} \;\middle|\; \mathsf{D}_{i,u-1} \wedge C_{i,u-2} = c_{i,u-2}\right] \cdot \Pr_b\left[\, C_{i,u-2} = c_{i,u-2} \;\middle|\; \mathsf{D}_{i,u-1}\,\right] . \tag{8}$$

The second term of Equation (8) is independent of b by Claim 1. For the first term we claim:

$$\Pr_b\left[\overline{\mathsf{D}_{i,u}} \;\middle|\; \mathsf{D}_{i,u-1} \wedge C_{i,u-2} = c_{i,u-2}\right] \;=\; \frac{|\mathsf{Proh}_{i,u}(c_{q,n_q})|}{2^l} . \tag{9}$$

To see Equation (9), note $\overline{\mathsf{D}_{i,u}}$ occurs when $C_i[u-1]$ falls in the prohibited set. We know that $C_i[u-1] = f(C_i[u-2] \oplus M_i[u-1])$ in Game 0 and $C_i[u-1] = f(C_i[u-2] \oplus M_i'[u-1])$ in Game 1. Given that $\mathsf{D}_{i,u-1}$ is true, in either game, f has not previously been invoked on either $C_i[u-2] \oplus M_i[u-1]$ or $C_i[u-2] \oplus M_i'[u-1]$ and thus $C_i[u-1]$ is uniformly distributed. Thus its chance of landing in the prohibited set is as claimed. Finally, note that $\mathsf{Proh}_{i,u}(c_{q,n_q})$ involves only ciphertexts in $c_{i,u-2}$. This means its size is fixed and in particular independent of the Game. We have thus completed the proof that the quantity in Equation (8) is independent of b.

It remains to show Equation (7) for the case $u = 1$. We proceed similarly with mainly just a change in notation. We can assume $i \geq 2$ since the case $(i,u) = (1,1)$ was covered in the base case of the induction. Write

$$\Pr_b\left[\overline{\mathsf{D}_{i,1}} \;\middle|\; \mathsf{D}_{i,0}\right] \;=\;$$
$$\sum_{c_{i-1,n_{i-1}}} \Pr_b\left[\overline{\mathsf{D}_{i,1}} \;\middle|\; \mathsf{D}_{i,0} \wedge C_{i-1,n_{i-1}} = c_{i-1,n_{i-1}}\right] \cdot \Pr_b\left[\, C_{i-1,n_{i-1}} = c_{i-1,n_{i-1}} \;\middle|\; \mathsf{D}_{i,0}\,\right] .$$

Again, take the above sum term by term. Fix c_{q,n_q}, thereby fixing one term of the sum. In this term (itself a product of two terms) first consider the second term. We could equally well condition on $D_{i-1,n_{i-1}+1}$ without changing the probability. Then, we see the quantity i is independent of b by Claim 1. For the first term, argue analogously to the above in terms of the prohibited set. Note that $C_i[u-1]$ is random (being the initial vector) and the prohibited set, and thus its size, depends only on quantities that we have fixed via the conditioning. Thus this term is also independent of b. This completes the proof of Claim 3. □

We now let $p \stackrel{\text{def}}{=} \Pr_0\left[\overline{\mathsf{D}}\right] = \Pr_1\left[\overline{\mathsf{D}}\right]$. The following is the upper bound of the second claim in the statement of Lemma 18.

Claim 4. $p \leq \left(\frac{\mu^2}{l^2} - \frac{\mu}{l}\right) \cdot \frac{1}{2^l}$.

Proof: Standard conditioning and bounding says that

$$\Pr_0\left[\overline{\mathsf{D}}\right] \leq \sum_{i=1}^{q}\sum_{u=1}^{n_i} \Pr_0\left[\overline{\mathsf{D}_{i,u}} \mid \mathsf{D}_{i,u-1}\right] .$$

A collision occurs when $C_i[u-1]$ falls in $\mathsf{Proh}_{i,u}(\cdot)$. Now we can apply Equation (1) to upper bound the above by

$$\begin{aligned}\sum_{i=1}^{q}\sum_{u=1}^{n_i} \frac{2(n_1+\ldots+n_{i-1}+u-1)}{2^l} &= \frac{2}{2^l}\sum_{i=1}^{q}\left(n_i(n_1+\ldots+n_{i-1}) + \frac{(n_i-1)n_i}{2}\right) \\ &= \frac{1}{2^l}\left[\frac{\mu^2}{l^2} - \frac{\mu}{l}\right] .\end{aligned}$$

This completes the proof of Claim 4. □

The following is the lower bound of the second claim in the statement of Lemma 18.

Claim 5: $p \geq \frac{1}{2}\left(1-\frac{1}{e}\right)\cdot\frac{1}{2^l}\left(\frac{\mu^2}{l^2} - \frac{\mu}{l}\right)$.

Proof: We upper bound the complementary event using Equation (1):

$$\begin{aligned}\Pr_0[\mathsf{D}] &= \prod_{i=1}^{q}\prod_{u=1}^{n_i} \Pr_0[\mathsf{D}_{i,u} \mid \mathsf{D}_{i,u-1}] \\ &\leq \prod_{i=1}^{q}\prod_{u=1}^{n_i} \frac{2^l - (n_1+\ldots+n_{i-1}+u-1)}{2^l} \\ &= \prod_{i=1}^{q}\prod_{u=1}^{n_i}\left(1 - \frac{n_1+\ldots+n_{i-1}+u-1}{2^l}\right) .\end{aligned}$$

Using the inequality $1-x \leq e^{-x}$ of Fact 14 we can upper bound the above by e^{-M} where

$$M = \sum_{i=1}^{q}\sum_{u=1}^{n_i} \frac{n_1+\ldots+n_{i-1}+u-1}{2^l} = \frac{1}{2}\frac{1}{2^l}\left[\frac{\mu^2}{l^2} - \frac{\mu}{l}\right] .$$

But $p \geq 1-e^{-M}$. Now apply the inequality $1-e^{-M} \geq (1-e^{-1})M$ of Fact 14 to get

$$p \geq \frac{1}{2}\left(1-\frac{1}{e}\right)\cdot\frac{1}{2^l}\left[\frac{\mu^2}{l^2} - \frac{\mu}{l}\right] .$$

This completes the proof of Claim 5. □

The following is the third claim in the statement of Lemma 18.

Claim 6: $\mathbf{Adv}^{\text{lor-cpa}}_{\text{CBC}[R],A}(\cdot) = \left(\Pr_1\left[\, A=1 \mid \overline{\mathsf{D}} \,\right] - \Pr_0\left[\, A=1 \mid \overline{\mathsf{D}} \,\right]\right) \cdot p.$

Proof: By conditioning we have

$$\begin{aligned}
\mathbf{Adv}^{\text{lor-cpa}}_{\text{CBC}[R],A}(\cdot) &= \Pr_1[\, A=1 \,] - \Pr_0[\, A=1 \,] \\
&= \Pr_1\left[\, A=1 \mid \overline{\mathsf{D}_{q,n_q}} \,\right]\Pr_1\left[\overline{\mathsf{D}_{q,n_q}}\right] + \Pr_1[\, A=1 \mid \mathsf{D}_{q,n_q} \,]\Pr_1[\mathsf{D}_{q,n_q}] \\
&\quad -\Pr_0\left[\, A=1 \mid \overline{\mathsf{D}_{q,n_q}} \,\right]\Pr_0\left[\overline{\mathsf{D}_{q,n_q}}\right] - \Pr_0[\, A=1 \mid \mathsf{D}_{q,n_q} \,]\Pr_0[\mathsf{D}_{q,n_q}] \,.
\end{aligned}$$

The proof of Claim 6 is concluded by applying Claims 2 and 3. □

This concludes the proof of Lemma 18. ▮

Acknowledgments

We thank Ran Canetti, who gave some helpful comments on an earlier draft, and Jim Gray, who suggested the variant of Definition 2 which appears here.

References

[1] W. Alexi, B. Chor, O. Goldreich, C. Schnorr, "RSA and Rabin functions: Certain parts are as hard as the whole," *SIAM Journal on Computing* Vol. 17, No. 2, 1988, pp. 194–209.

[2] ANSI X3.106, "American National Standard for Information Systems – Data Encryption Algorithm – Modes of Operation," American National Standards Institute, 1983.

[3] M. Bellare, R. Canetti and H. Krawczyk "Psuedorandom functions revisited: The cascade construction and its concrete security," *Proceedings of the 37th Symposium on Foundations of Computer Science*, IEEE, 1996.

[4] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology - Crypto '98*, LNCS Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[5] M. Bellare, R. Guérin and P. Rogaway, "XOR MACs: New methods for message authentication using finite pseudorandom functions," *Advances in Cryptology - Crypto '95*, LNCS Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.

[6] M. Bellare, J. Kilian and P. Rogaway, "The security of the cipher block chaining message authentication code," *Advances in Cryptology - Crypto '94*, LNCS Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.

[7] M. Bellare and P. Rogaway, "Optimal asymmetric encryption – How to encrypt with RSA," *Advances in Cryptology - Eurocrypt '95*, LNCS Vol. 921, L. Guillou and J. Quisquater ed., Springer-Verlag, 1995.

[8] M. Bellare and P. Rogaway, "The exact security of digital signatures: How to sign with RSA and Rabin," *Advances in Cryptology - Eurocrypt '96*, LNCS Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[9] M. Blum and S. Goldwasser, "An efficient probabilistic public-key encryption scheme which hides all partial information," *Advances in Cryptology - Crypto '84*, LNCS Vol. 196, R. Blakely ed., Springer-Verlag, 1984.

[10] D. Dolev, C. Dwork and M. Naor, "Non-malleable cryptography," *SIAM J. of Computing*, to appear. Preliminary version in *Proceedings of the 23rd Annual Symposium on the Theory of Computing*, ACM, 1991.

[11] O. Goldreich "A uniform complexity treatment of encryption and zero-knowledge," *Journal of Cryptology,* Vol. 6, 1993, pp. 21-53.

[12] O. Goldreich, R. Impagliazzo, L. Levin, R. Venkatesan and D. Zuckerman, "Security preserving amplification of hardness," *Proceedings of the 31st Symposium on Foundations of Computer Science*, IEEE, 1990.

[13] O. Goldreich and L. Levin, "A hard-core predicate for all one-way functions," *Proceedings of the 21st Annual Symposium on the Theory of Computing*, ACM, 1989.

[14] O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions," *Journal of the ACM,* Vol. 33, No. 4, 1986, pp. 210–217.

[15] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. of Computer and System Sciences*, Vol. 28, April 1984, pp. 270–299.

[16] A. Herzberg and M. Luby, "Public randomness in cryptography," *Advances in Cryptology - Crypto '92*, LNCS Vol. 740, E. Brickell ed., Springer-Verlag, 1992.

[17] J. Håstad, R. Impagliazzo, L. Levin and M. Luby, "Construction of a pseudo-random generator from any one-way function," *ICSI Technical Report*, No. 91-068, submitted to SICOMP.

[18] ISO 8372, "Information processing – Modes of operation for a 64-bit block cipher algorithm," International Organization for Standardization, Geneva, Switzerland, 1987.

[19] J. Katz and M. Yung, "Complete characterization of security notions for probabilistic private-key encryption," *Proceedings of the 32nd Annual Symposium on the Theory of Computing*, ACM, 2000.

[20] M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton University Press, 1996.

[21] M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," *SIAM J. Computation,* Vol. 17, No. 2, April 1988.

[22] S. Micali, C. Rackoff and R. Sloan, "The notion of security for probabilistic cryptosystems," *SIAM J. of Computing*, April 1988.

[23] National Bureau of Standards, NBS FIPS PUB 81, "DES modes of operation," U.S Department of Commerce, 1980.

[24] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," *Proceedings of the 22nd Annual Symposium on the Theory of Computing*, ACM, 1990.

[25] C. Rackoff and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen-ciphertext attack," *Advances in Cryptology - Crypto '91*, LNCS Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.

[26] A. C. Yao, "Theory and applications of trapdoor functions," *Proceedings of the 23rd Symposium on Foundations of Computer Science*, IEEE, 1982.