Preliminary version.

# Concrete Security Characterizations of PRFs and PRPs: Reductions and Applications

Anand Desai*　　　Sara Miner*

May 2000

**Abstract**

We investigate, in a concrete security setting, several alternate characterizations of pseudorandom functions (PRFs) and pseudorandom permutations (PRPs). By analyzing the concrete complexity of the reductions between the standard notions and the alternate ones, we show that the latter, while equivalent under polynomial-time reductions, are weaker in the concrete security sense. With these alternate notions, we argue that it is possible to get better concrete security bounds for certain PRF/PRP-based schemes. As an example, we show how using an alternate characterization of a PRF could result in tighter security bounds for a certain class of message authentication codes. We also apply these techniques to give a simple concrete security analysis of the counter mode of encryption. In addition, our results provide some insight into how injectivity impacts pseudorandomness.

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

# Contents

# 1 Introduction

Pseudorandom functions (PRFs) and pseudorandom permutations (PRPs) are extremely useful and widely used tools in cryptographic protocol design, particularly in the setting of private-key cryptography. In this paper, we study several different notions that may be used to characterize these objects. Specifically, we study these notions in a concrete security framework, and we show how different characterizations may be used to derive better security bounds for some commonly used private-key cryptographic protocols.

## 1.1 Descriptions of Notions

The notion of a PRF family was proposed by Goldreich, Goldwasser and Micali [7]. In such a family, each function is specified by a short, random key, and can be easily computed given the random key. Yet it has the property that it is computationally infeasible (for someone who does not know the key) to tell apart a function sampled from the PRF family and one from a random function family, given adaptive access to the function as a black-box. This is the standard notion of a PRF, and (to distinguish it from alternate notions) we refer to it in this paper as the PRF notion. Extending the notion above to permutation families, Luby and Rackoff introduced the notion of a PRP family [10], and we refer to it here as the PRP notion.

ALTERNATE CHARACTERIZATIONS. In addition to the standard notion, PRFs may be characterized in several different ways. We are particularly interested in one way suggested in the very paper that introduced the standard notion [7]. This alternate notion can be described informally through the following interactive protocol: a distinguisher who is given adaptive oracle access to the function obtains the output of the function on some points of its choice through oracle queries. It then outputs a point that has not been queried yet and gets back, based on a hidden coin flip, either the output of the function on that point or a uniformly distributed point in the range of the function. It should be computationally infeasible for the distinguisher to guess which of the two possibilities it was presented. We call this notion indistinguishable-uniform functions or IUF, to distinguish it from the standard notion PRF. A similar notion may be defined for permutation families, and we call this IUP for indistinguishable-uniform permutations.

As a second alternative, we consider another indistinguishability-based characterization that is normally associated with the security of encryption schemes. Again in this notion, the distinguisher is given adaptive oracle access to the function and uses this to obtain the value of the function on some points of its choice. It then outputs *two* new points and, based on a hidden coin flip, is presented with the output of the function on one of them. We require that a computationally restricted distinguisher have negligible success in telling apart the two cases. In this paper, we refer to this notion as IPF, for indistinguishable-point functions. Not surprisingly, however, this notion does not actually capture the idea of pseudorandomness for functions. However, we include it here for completeness because, when we consider the analogous notion for permutations, which we call IPP (indistinguishable-point permutations), we find that pseudorandomness is implied.

## 1.2 Concrete Security and Reductions Among the Notions

Making a break from the traditional approach of presenting PRF families in an asymptotic way, Bellare, Kilian and Rogaway started the practice of explicitly specifying the resources determining the security and paying particular attention to the quality of the security reductions [5]. This approach forms the basis of what is called concrete security analysis and has been used in many subsequent works [4, 2, 3]. One of the benefits of this approach is to enable the comparison and

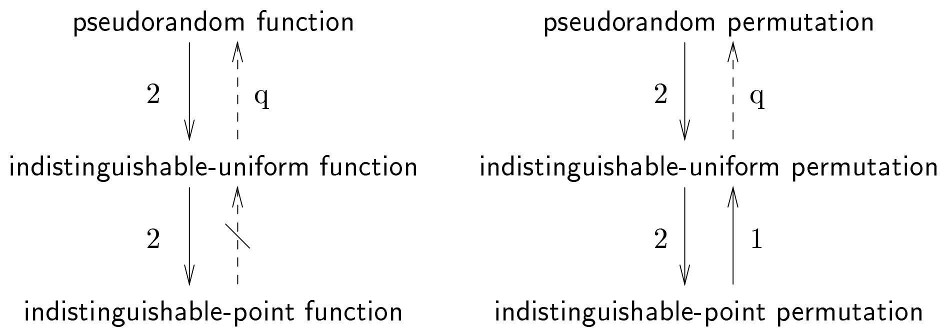


Figure 1: *Relating the notions. A solid arrow from notion A to notion B means that there is a security-preserving reduction from A to B. A broken arrow indicates a reduction that is not security-preserving. The arrows are labeled by the loss-factor of the reduction. A hatched arrow means that there is no polynomial-time reduction.*

classification as weaker or stronger of polynomially-equivalent notions in cryptography. Paying attention to the concrete complexity of reductions between notions is important in practice as inefficient reductions translate to a penalty either in security assurance or running time.

Reductions Among the Notions. The equivalence, under polynomial-time reductions, between the notions of PRF and IUF has been established in [7]. (In fact, the concrete security bounds we derive in our reductions between these notions are implicit in theirs.) We establish that our reductions are tight, in that one cannot hope to do any better. Additionally, we relate the notions of PRP and IUP. The relationship between these two permutation notions is the same as that between the corresponding notions for functions.

We also show that IUP and IPP are equivalent, up to a small constant factor in the reduction. However, as mentioned above, a different picture emerges when we look at the corresponding notions of function families. It turns out that IPF and IUF (or PRF) are not equivalent, even in just an asymptotic sense. We show that IPF is a strictly weaker notion, in that there are function families that are secure in the IPF sense, but completely insecure in the IUF sense. A summary of the reductions is given in Figure 1.

## 1.3 Motivation: Tighter Security Analyses

Showing that the alternate notions we consider here are weaker than the corresponding standard notions in the concrete security setting could be seen as an argument *against* using any of these alternate notions. Yet we recommend their use in certain circumstances (to complement, rather than replace the standard ones).

In a concrete security analysis of a protocol based on some primitive, the security of the protocol is related to that of the underlying primitive in a precise and quantitative way. If we have the concrete security of a protocol in terms of that of the underlying primitive under some notion, then it is easy to translate this to the security in terms of that of the underlying primitive under a weaker notion. We simply use the appropriate security reduction between the notions for this. We would see a drop in the translated security, reflecting the gap in the reduction between the notions. Clearly there isn't much to be gained by this alone. However we will see that it is sometimes possible to directly reduce the security of the protocol to that of the underlying primitive under a weaker notion *without* the expected drop in security. Such a situation exists when the weaker notion somehow "meshes" better with the notion of security for the protocol.

We make the above discussion more concrete with several examples: (deterministic) message authentication codes and symmetric encryption schemes. The security of message authentication codes (MACs) is captured by the notion of unpredictable functions [9, 1, 11]. In the context of MACs, this means that the adversary is allowed to see the output of the MAC on some messages of its choice and then must output a "new" message (that is one different from those whose MACs it had obtained in the earlier phase), along with a valid MAC on that message. It is well-known that any PRF is unpredictable (i.e. a secure MAC) [7]. Moreover, the reduction from unpredictable functions to PRF is almost tight [5]. It turns out that the bounds obtained by a direct reduction from unpredictable functions to IUF would be *exactly* the same. This represents a tightening of the analysis as we would expect security of a PRF in the IUF sense to be smaller than the security in the standard PRF sense. (More accurately we should say that the security in the IUF sense will never be more than a constant factor 2 greater than the security in the PRF sense and will typically be a quantitative factor less.)

Now let us examine in what sense IUF "meshes" better with the notion of unpredictable functions. The quantitative drop in security in the reduction from PRF to IUF can be traced to the fact that under IUF the distinguisher must "decide" given *one* challenge whereas under PRF every response to a query potentially constitutes a "challenge". Like IUF, the notion of unpredictable functions also has a single distinguished challenge. In the reduction to PRF, however, we cannot really take any advantage of the source of the strength of this notion, and hence the bounds derived are not as tight as what could be achieved otherwise.

Another example of a notion having a distinguished challenge phase is the standard indistinguishability of encryptions notion of security for encryption schemes [8, 3]. Here again, using the notion of IUF rather than PRF, we can hope to tighten analysis of block-cipher-based encryption schemes. We do this for the *counter mode* of encryption.

### 1.4 Related Work

We have already mentioned the foundational work on PRFs and PRPs [7, 10] and the concrete security analysis of these objects initiated in [5, 4]. Our approach in this work follows that of Bellare et al who compared and classified notions of security for symmetric encryption schemes according to the concrete complexity of reductions [3]. A concrete security analysis of various symmetric encryption schemes, including the counter mode, is given in that paper. Naor and Reingold have explored the relationship between unpredictable functions and PRFs under different attack models [11].

## 2 Definitions and Notation

A *function family* is a keyed multi-set $F$ of functions where all the functions have the same domain and range. To pick a function $f$ from family $F$ means to pick a key $a$, uniformly from the key space $\mathsf{Keys}(F)$ of $F$, and let $f = F_a$. A family $F$ has input length $l$ and output length $L$ if each $f \in F$ maps $\{0,1\}^l$ to $\{0,1\}^L$.

We let $R_{l,L}$ denote the function family consisting of all functions with input length $l$ and output length $L$. Similarly, we let $P_l$ denote the set of all permutations on $l$-bit strings.

A finite function family $F$ is pseudorandom if the input-output behavior of $F_a$ is indistinguishable from the behavior of a random function of the same domain and range. This is formalized via the notion of statistical tests [7]. Our concrete security formalizations are those of [5].

Here we informally describe the two alternate notions for functions considered in this paper. The corresponding alternate characterizations for permutations (IUP and IPP) are analogous to these, and therefore we do not state them informally here. Finally, at the end of this section, we formally define experiments and advantage functions for all six notions.

Indistinguishable-Uniform Functions. This is an adaptation of a notion given in [7]. The idea is that a distinguisher should not be able to distinguish the output of the PRF from a uniformly distributed value in the range of the function. The formalization considers two different experiments. In both experiments we start by choosing a random key $a \leftarrow \mathsf{Keys}(F)$, specifying a function $F_a$. In the first phase, the distinguisher is given an oracle for $F_a$ and allowed to query this oracle on points of its choice. It then outputs a point $x$ that has not been queried yet and some state information $s$ that it may want preserve for help in the second phase. In one experiment it receives in response the value $F_a(x)$. In the other experiment it receives a uniformly distributed value in the range of $F$. The PRF family is "good" if no "reasonable" distinguisher can obtain "significant" advantage in distinguishing the two experiments.

Indistinguishable-Point Functions. This is an adaptation of the notion of security for encryption captured by Definition 4.4 in Section Section 4.2. Here again we imagine a distinguisher $A$ that runs in two phases. In the find phase, given adaptive access to an oracle for the function, it comes up with a pair of points $x_0, x_1$ that it has not queried yet and some state information $s$. In the guess phase, given the output of the function $y$ on one of these points and $s$, it must identify which of the two points goes with $y$.

It is interesting that the notion IPP does capture pseudorandomness for permutation families, while IPF does not do so for function families. For most other primitives, we find that an indistinguishable-point-based characterization is weaker than an indistinguishable-uniform-based characterization. This is true for encryption schemes and turns out to be true for function families, as well. Observe that, for encryption schemes, we are usually concerned with this weaker characterization, because it captures the desired security requirements.

Formal Definitions. For each of the six notions we consider in this paper, we give definitions using the experiments defined in Figure 2. First, we consider the function family notions: PRF, IUF, and IPF.

**Definition 2.1** For each notion $\mathsf{N} \in \{\mathsf{PRF}, \mathsf{IUF}, \mathsf{IPF}\}$, let $F\colon \mathsf{Keys}(F) \times \{0,1\}^l \to \{0,1\}^L$ be a function family. Let $A$ be an algorithm that takes an oracle for a function $f : \{0,1\}^l \mapsto \{0,1\}^L$, and outputs a bit. Now, we consider the corresponding experiment $\mathsf{Exp}_F^{\mathsf{N}}(A, b)$, given in Figure 2. We define the *advantage* of $A$ and the *advantage function* of $F$ as follows. For any $t, q \geq 0$,

$$\begin{aligned}\mathsf{Adv}_F^{\mathsf{N}}(A) &= \Pr\left[\,\mathsf{Exp}_F^{\mathsf{N}}(A,0) = 0\,\right] - \Pr\left[\,\mathsf{Exp}_F^{\mathsf{N}}(A,1) = 0\,\right] \\ \mathsf{Adv}_F^{\mathsf{N}}(t,q) &= \max_A \{\mathsf{Adv}_F^{\mathsf{N}}(A)\}\end{aligned}$$

where the maximum is over all $A$ with time complexity $t$ and making at most $q$ queries. ▮

Here the "time-complexity" is the worst case total code of $A$, in some fixed RAM model of computation. This convention is used for all definitions in this paper.
Now, we turn our attention to the analogous definitions for the corresponding permutation family notions: PRP, IUP, and IPP.

**Definition 2.2** For each notion $\mathsf{N} \in \{\mathsf{PRP}, \mathsf{IUP}, \mathsf{IPP}\}$, let $F\colon \mathsf{Keys}(F) \times \{0,1\}^l \to \{0,1\}^l$ be a permutation family. Let $A$ be an algorithm that takes an oracle for a permutation $f : \{0,1\}^l \mapsto$

| | |
|---|---|
| PRF: $\mathsf{Exp}_F^{\mathrm{prf}}(A,b)$<br>$a \leftarrow \mathsf{Keys}(F)$<br>$\mathcal{O}_0 \leftarrow F_a; \mathcal{O}_1 \leftarrow R_{l,L}$<br>$d \leftarrow A^{\mathcal{O}_b}$<br>**return** $d$ | PRP: $\mathsf{Exp}_F^{\mathrm{prp}}(A,b)$<br>$a \leftarrow \mathsf{Keys}(F)$<br>$\mathcal{O}_0 \leftarrow F_a; \mathcal{O}_1 \leftarrow P_l$<br>$d \leftarrow A^{\mathcal{O}_b}$<br>**return** $d$ |
| IUF: $\mathsf{Exp}_F^{\mathrm{iuf}}(A,b)$<br>$a \leftarrow \mathsf{Keys}(F)$<br>$(x,s) \leftarrow A^{F_a}(\mathit{find})$<br>$y_0 \leftarrow F_a(x); y_1 \stackrel{R}{\leftarrow} \{0,1\}^L$<br>$d \leftarrow A(\mathit{guess}, y_b, s)$<br>**return** $d$ | IUP: $\mathsf{Exp}_F^{\mathrm{iup}}(A,b)$<br>$a \leftarrow \mathsf{Keys}(F)$<br>$(x,s) \leftarrow A^{F_a}(\mathit{find})$<br>$y_0 \leftarrow F_a(x); y_1 \stackrel{R}{\leftarrow} \{0,1\}^l$<br>$d \leftarrow A(\mathit{guess}, y_b, s)$<br>**return** $d$ |
| IPF: $\mathsf{Exp}_F^{\mathrm{ipf}}(A,b)$<br>$a \leftarrow \mathsf{Keys}(F)$<br>$(x_0,x,s) \leftarrow A^{F_a}(\mathit{find})$<br>$y \leftarrow F_a(x_b)$<br>$d \leftarrow A(\mathit{guess}, y, s)$<br>**return** $d$ | IPP: $\mathsf{Exp}_F^{\mathrm{ipp}}(A,b)$<br>$a \leftarrow \mathsf{Keys}(F)$<br>$(x_0,x,s) \leftarrow A^{F_a}(\mathit{find})$<br>$y \leftarrow F_a(x_b)$<br>$d \leftarrow A(\mathit{guess}, y, s)$<br>**return** $d$ |

Figure 2: Experiments used to define each of the notions considered in this paper.

$\{0,1\}^l$, and outputs a bit. Now, we consider the experiment $\mathsf{Exp}_F^{\mathsf{N}}(A,b)$. We define the advantage of $A$ and the advantage function of $F$ as follows. For any $t,q \geq 0$,

$$\begin{aligned} \mathsf{Adv}_F^{\mathsf{N}}(A) &= \Pr\left[\,\mathsf{Exp}_F^{\mathsf{N}}(A,0)=0\,\right] - \Pr\left[\,\mathsf{Exp}_F^{\mathsf{N}}(A,1)=0\,\right] \\ \mathsf{Adv}_F^{\mathsf{N}}(t,q) &= \max_A \{\mathsf{Adv}_F^{\mathsf{N}}(A)\} \end{aligned}$$

where the maximum is over all $A$ with time complexity $t$ and making at most $q$ queries. ▮

# 3 Reductions Among the Notions

In this section, we formalize the relations between the characterizations given in the diagram in Figure 1. 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. This convention is followed in all reductions given in this paper. Proofs of the results in this section are given in Appendix A.

## 3.1 Function Notions

The first theorem says that if a function family has certain security in the standard PRF sense, then it has essentially the same security in the IUF sense.

**Theorem 3.1** [$\mathsf{PRF} \Rightarrow \mathsf{IUF}$] For any function family $F$ and any $t_2, q_2 > 0$,

$$\mathsf{Adv}_F^{\mathrm{iuf}}(t_2, q_2) \quad \leq \quad 2 \cdot \mathsf{Adv}_F^{\mathrm{prf}}(t_1, q_1)$$

where $t_1 = t_2 + O(l + L)$ and $q_1 = q_2$.

Our next theorem says that if a function family has certain security in the IUF sense, then it is also secure in the PRF sense, but the security is quantitatively lower.

**Theorem 3.2** [$\mathsf{IUF} \rightarrow \mathsf{PRF}$] For any function family $F$ and $t_1, q_1 > 0$,

$$\mathsf{Adv}_F^{\mathrm{prf}}(t_1, q_1) \quad \leq \quad q_1 \cdot \mathsf{Adv}_F^{\mathrm{iuf}}(t_2, q_2)$$

where $t_2 = t_1 + O(l + L)$ and $q_2 = q_1$.

The following proposition establishes that the drop in security in the previous theorem was not any weakness of our reduction but is in fact intrinsic to the notions. In our proof, we give a concrete scheme that has higher security in the PRF sense, with the gap being of the same order as in the theorem above.

**Proposition 3.3** [$\mathsf{IUF} \not\Rightarrow \mathsf{PRF}$] There exists a function family $F$ such that

$$\mathsf{Adv}_F^{\mathrm{prf}}(t, q) \quad \geq \quad \frac{q}{2} \cdot \mathsf{Adv}_F^{\mathrm{iuf}}(t, q)$$

for any $t \geq 0$ and $q \leq 2^{L-1}$.

The following theorems demonstrate that the notion IPF is strictly weaker than the other two notions we have considered, and hence does not capture pseudorandomness.

**Theorem 3.4** [$\mathsf{IUF} \Rightarrow \mathsf{IPF}$] For any function family $F$ and any $t_3, q_3 > 0$,

$$\mathsf{Adv}_F^{\mathrm{ipf}}(t_3, q_3) \quad \leq \quad 2 \cdot \mathsf{Adv}_F^{\mathrm{iuf}}(t_2, q_2)$$

where $t_2 = t_3 + O(l)$ and $q_2 = q_3$.

**Proposition 3.5** [$\mathsf{IPF} \not\Rightarrow \mathsf{IUF}$] There exists a function family $F$ such that both

$$\begin{aligned} \mathsf{Adv}_F^{\mathrm{iuf}}(t, q) \quad &\geq \quad 1 - 2^{-qL} \\ \mathsf{Adv}_F^{\mathrm{ipf}}(t, q) \quad &= \quad 0 \end{aligned}$$

for any $t, q \geq 0$.

## 3.2 Permutation Notions

Next, we give the reductions between PRP and IUP. The situation is very similar to the relations that emerged between PRF and IUF, the analogous notions for functions. In fact, our next three claims show that the concrete security bounds we had derived between the notions for function families also hold between the corresponding notions for permutation families.

**Theorem 3.6** [PRP $\Rightarrow$ IUP] For any permutation family $F$ and any $t_2, q_2 > 0$,

$$\mathsf{Adv}_F^{\mathrm{iup}}(t_2, q_2) \quad \leq \quad 2 \cdot \mathsf{Adv}_F^{\mathrm{prp}}(t_1, q_1)$$

where $t_1 = t_2 + O(l)$ and $q_1 = q_2$.

**Theorem 3.7** [IUP $\rightarrow$ PRP] For any permutation family $F$ and $t_1, q_1 > 0$,

$$\mathsf{Adv}_F^{\mathrm{prp}}(t_1, q_1) \quad \leq \quad q_1 \cdot \mathsf{Adv}_F^{\mathrm{iup}}(t_2, q_2)$$

where $t_2 = t_1 + O(l)$ and $q_2 = q_1$.

**Proposition 3.8** [IUP $\not\Rightarrow$ PRP] There exists a permutation family $F$ such that

$$\mathsf{Adv}_F^{\mathrm{prp}}(t, q) \quad \geq \quad \frac{q}{2} \cdot \mathsf{Adv}_F^{\mathrm{iup}}(t, q)$$

for any $t \geq 0$ and $q \leq 2^{L-1}$.

Here, we establish that IUP and IPP, the two non-standard notions of pseudrandomness for permutations, are of essentially equivalent strength. Note that this is a departure from the relationship that exists between the analogous function family characterizations.

**Theorem 3.9** [IUP $\Rightarrow$ IPP] For any permutation family $F$ and any $t_3, q_3 > 0$,

$$\mathsf{Adv}_F^{\mathrm{ipp}}(t_3, q_3) \quad \leq \quad 2 \cdot \mathsf{Adv}_F^{\mathrm{iup}}(t_2, q_2)$$

where $t_2 = t_3 + O(l)$ and $q_2 = q_3$.

**Theorem 3.10** [IPP $\Rightarrow$ IUP] For any permutation family $F$ and any $t_2, q_2 > 0$,

$$\mathsf{Adv}_F^{\mathrm{iup}}(t_2, q_2) \quad \leq \quad \mathsf{Adv}_F^{\mathrm{ipp}}(t_3, q_3)$$

where $t_3 = t_2 + O(l)$ and $q_3 = q_2$.

# 4 Applications

Here, we give some motivation for the use of the IUF characterization of PRF families. As discussed in Section 1, use of this notion gives tighter security bounds for certain cryptographic protocols. We give two such examples in this section.

## 4.1 The case of message authentication codes

A message authentication code (MAC) enables two parties who share a secret key to authenticate their transmissions. The security property required of MACs is that they must resist existential forgery under chosen-message attacks [9, 5]. For deterministic MACs, this notion matches that of unpredictable functions [1, 11].

Formally, the notion is captured by allowing a distinguisher $A$ to query a MAC oracle, $F_a$, where $F$ is a function family and $a$ is a random MAC key. $A$ must then output a point $x$ that has not been queried yet, along with its prediction $y$ for the value of $F_a(x)$.

**Definition 4.1** [Message authentication security: UPF] Let $F$: $\mathsf{Keys}(F) \times \{0,1\}^l \to \{0,1\}^L$ be a MAC. Let $A$ be an algorithm that takes an oracle for a function $f : \{0,1\}^l \mapsto \{0,1\}^L$. Now, we consider the following experiment:

Experiment $\mathsf{Exp}_F^{\mathrm{upf}}(A)$
  $a \leftarrow \mathsf{Keys}(F)$; $(x, y) \leftarrow A^{F_a}$ [where $x$ is a point that $A$ has not queried]
  If $y = F_a(x)$ then $d \leftarrow 0$ else $d \leftarrow 1$; `Return` $d$.

We define the advantage of $A$ and the advantage function of $F$ as follows. For any $t, q \geq 0$,

$$\begin{aligned} \mathsf{Adv}_F^{\mathrm{upf}}(A) &= \Pr\left[\,\mathsf{Exp}_F^{\mathrm{upf}}(A) = 0\,\right] \\ \mathsf{Adv}_F^{\mathrm{upf}}(t, q) &= \max_A \{\mathsf{Adv}_F^{\mathrm{upf}}(A)\} \end{aligned}$$

where the maximum is over all $A$ with time complexity $t$ and making at most $q$ queries. ▮

PRF families are more well-studied than unpredictable function families and, moreover, are widely available. Hence the observation that a PRF family constitutes a secure MAC [7] has proved very useful in practice. The following exact security reduction appears in [5].

**Proposition 4.2** [PRF $\Rightarrow$ UPF] For any function family $F$ and $t, q > 0$,

$$\mathsf{Adv}_F^{\mathrm{upf}}(t, q) \leq \mathsf{Adv}_F^{\mathrm{prf}}(t', q) + 2^{-L}$$

where $t' = t + O(l + L)$.

The reduction is almost tight. Consider now translating the above, to get security as a MAC in terms of the security as a PRF family in the IUF sense. Using Theorem 3.2 will lead to a drop in security by a factor $q$. However, by applying a direct reduction, we avoid this expected loss.

**Proposition 4.3** [IUF $\Rightarrow$ UPF] For any function family $F$ and $t, q > 0$,

$$\mathsf{Adv}_F^{\mathrm{upf}}(t, q) \leq \mathsf{Adv}_F^{\mathrm{iuf}}(t', q) + 2^{-L}$$

where $t' = t + O(l + L)$.

**Proof:** The reduction is standard. Let $A$ be a forger attacking the MAC $F$, making at most $q$ oracle queries and running in time at most $t$, in the experiment $\mathsf{Exp}_F^{\mathrm{upf}}(A)$. We construct a distinguisher $A'$, making at most $q$ queries and running in time at most $t'$, using the forger $A$ as a subroutine.

Let $\mathcal{O}_f$ be $A'$'s oracle. $A'^{\mathcal{O}_f}$ will run $A$ using $\mathcal{O}_f$ to provide an appropriate simulation of $A$'s oracle, as indicated below.

Algorithm $A'^{\mathcal{O}_f}$

(1) Run $A$, answering any query $u$ with $\mathcal{O}_f(u)$.
(2) Let $(x, y) \leftarrow A$.
(3) Output $(x, y)$ and receive $y'$ as the challenge.
(4) If $y' = y$ then output 0, else output 1.

We assume for simplicity that $A$ makes exactly $q$ queries in $\mathsf{Exp}_F^{\rm upf}(A)$. It is easy to check that the time and query complexity are as claimed. Next, we compute the advantage of $A'$.

$$\begin{aligned}\mathsf{Adv}_F^{\rm iuf}(A') &= \Pr\left[\mathsf{Exp}_F^{\rm iuf}(A',0)=0\right] - \Pr\left[\mathsf{Exp}_F^{\rm iuf}(A',1)=0\right] \\ &= \Pr\left[\mathsf{Exp}_F^{\rm upf}(A)=0\right] - 2^{-L} = \mathsf{Adv}_F^{\rm upf}(A) - 2^{-L}\end{aligned}$$

Given that $A$ was any arbitrary forger, the relation in the advantage functions follow. ▮

Proposition 4.3 represents a tightening of the security bounds appearing in Proposition 4.2 since for most $F$ and any $t, q \geq 0$, we can expect $\mathsf{Adv}_F^{\rm iuf}(t,q) \approx \frac{1}{q} \cdot \mathsf{Adv}_F^{\rm prf}(t,q)$. As mentioned earlier, this can be traced to the fact that in the PRF notion every one of the $q$ queries of the distinguisher results in a challenge, whereas in the IUF notion, the distinguisher receives just one challenge.

## 4.2 The case of symmetric encryption schemes

In the following discussion we use the standard syntax and notion of security for encryption schemes given in [3]. The notion of security is an adaptation of one given in [8]. In the indistinguishability of encryptions under chosen-plaintext attacks the adversary $A$ is imagined to run in two phases. In the find phase, given adaptive access to an encryption oracle, $A$ comes up with a pair of messages $x_0, x_1$ along with some state information $s$ to help in the second phase. In the guess phase, given the encryption $y$ of one of the messages and $s$, it must identify which of the two messages goes with $y$.

**Definition 4.4** [Symmetric encryption scheme security: IND-CPA] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and $A$ be an algorithm that takes an encryption oracle and outputs a bit. Now, we consider the following experiment:

Experiment $\mathsf{Exp}_\Pi^{\text{ind-cpa}}(A,b)$
$a \leftarrow \mathcal{K}$; $(x_0, x_1, s) \leftarrow A^{\mathcal{E}_a}(\mathsf{find})$; $y \leftarrow \mathcal{E}_a(x_b)$; $d \leftarrow A^{\mathcal{E}_a}(\mathsf{guess}, y, s)$; `Return` $d$.

We define the advantage of $A$ and the advantage function of $\Pi$ as follows. For any $t, q, \mu \geq 0$,

$$\begin{aligned}\mathsf{Adv}_\Pi^{\text{ind-cpa}}(A) &= \Pr\left[\mathsf{Exp}_\Pi^{\text{ind-cpa}}(A,0)=0\right] - \Pr\left[\mathsf{Exp}_\Pi^{\text{ind-cpa}}(A,1)=0\right] \\ \mathsf{Adv}_\Pi^{\text{ind-cpa}}(t,q,\mu) &= \max_A \{\mathsf{Adv}_\Pi^{\text{ind-cpa}}(A)\}\end{aligned}$$

where the maximum is over all $A$ with time complexity $t$ and making at most $q$ queries, these totalling at most $\mu$ bits. ▮

We analyze the counter mode of encryption based on a finite PRF. In practice, the finite PRF may be instantiated by a block cipher. The counter mode $\text{CTR}(F) = (\mathcal{E}\text{-CTR}, \mathcal{D}\text{-CTR}, \mathcal{K}\text{-CTR})$ works as follows. The key generation algorithm $\mathcal{K}$-CTR outputs a random key $a$ for the underlying PRF family $F$, thereby specifying a function $f = F_a$ of $l$-bits to $L$-bits. The sender maintains a $l$ bit counter $ctr$ that is initially $-1$ and is incremented after each encryption by the number of blocks encrypted. 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 $\mathcal{E}\text{-CTR}_a(x, ctr) = \mathcal{E}\text{-CTR}^{F_a}(x, ctr)$ and $\mathcal{D}\text{-CTR}_a(z) = \mathcal{D}\text{-CTR}^{F_a}(z)$, where:

```
function ℰ-CTR^f(x, ctr)
    for i = 1,...,n do y_i = f(ctr + i) ⊕ x_i
    ctr ← ctr + n
    return (ctr, ctr‖y_1 y_2 ··· y_n)

function 𝒟-CTR^f(z)
    Parse z as ctr‖y_1 ··· y_n
    for i = 1,...,n do x_i = f(ctr + i) ⊕ y_i
    return x = x_1 ··· x_n
```

We show that CTR($F$) is secure in the IND-CPA sense if $F$ is secure in the IUF sense. As with our previous example, the reduction achieves the same concrete security bounds as those possible using the standard notion of PRF families.

**Theorem 4.5 [Security of** CTR **using indistinguishable-uniform functions]** For any function family $F$ and $t', q', q > 0$,

$$\mathsf{Adv}^{\text{ind-cpa}}_{\text{CTR}(F)}(t, q, \mu) \quad \leq \quad 2 \cdot \mathsf{Adv}^{\text{iuf}}_{F}(t', q')$$

where $\mu = \min(q'L, L2^l)$ and $t = t' - O(\frac{\mu}{L}(l + L))$.

**Proof:** We want to show that if CTR($F$) is not secure in the IND-CPA sense, then it must be the case that $F$ is not secure in the IUF sense. Let $A$ be an adversary attacking the CTR($F$), running in time at most $t$ and making at most $q$ oracle queries, these totalling at most $\mu$ bits, in the experiment $\mathsf{Exp}^{\text{ind-cpa}}_{\text{CTR}(F)}(A)$. We construct a distinguisher $A'$, making at most $q'$ queries and running in time at most $t'$, using the adversary $A$ as a subroutine.

Let $\mathcal{O}_f$ be $A'$'s oracle. $A'^{\mathcal{O}_f}$ will run $A$ using $\mathcal{O}_f$ to provide an appropriate simulation of $A$'s encryption oracle. We assume, for the sake of simplicity of the exposition, that the two messages $A$ outputs at the end of its first phase are exactly $L$ bits in length (i.e. of the size of one block). In the following, $\mu_G < \mu$, is the amount of ciphertext $A$ needs to see in its $\mathsf{guess}$ phase.

Algorithm $A'^{\mathcal{O}_f}$

(1) Initialize counter: $ctr \leftarrow -1$.
(2) Run $A(\mathsf{find})$, answering any query $u$ with $\mathcal{E}\text{-CTR}^{\mathcal{O}_f}(u)$.
(3) Let $(x_0, x_1, s) \leftarrow A(\mathsf{find})$.
(4) Let the current value of the counter be $ctr_0$.
(5) Compute $\mathcal{F} = \{\mathcal{O}_f(ctr_0 + i) : 1 \leq i \leq \frac{\mu_G}{L}\}$.
(6) Let $s' = (s, x_0, x_1, ctr_0, \mathcal{F})$.
(7) Output $(ctr_0, s')$ and receive $y$ as the challenge.
(8) Let $d \leftarrow \{0, 1\}$.
(9) Run $A(\mathsf{guess}, y \oplus x_d, s)$, answering any query $u$, using $\mathcal{F}$, with $\mathcal{E}\text{-CTR}(u)$.
(10) Let $d' \leftarrow A(\mathsf{guess}, y \oplus x_d, s)$.
(11) If $d = d'$ then output 0, else output 1.

In the reduction above, $A'$ maintains the counter $ctr$, appropriately incrementing it, as required. It is important here that $A'$ can implement $\mathcal{E}\text{-CTR}^f(\cdot, ctr)$ given an oracle for $f$. At the end of the $\mathsf{find}$ phase queries of $A$, it picks the current value of counter $ctr_0$ to be the output of its own $\mathsf{find}$ phase, along with the state information as indicated. A slight problem that comes up here is that $A'$ does not have access to $\mathcal{O}_f$ in its $\mathsf{guess}$ phase but it will still need to provide a simulation of the encryption oracle during $A$'s guess phase queries, We get around this by having $A'$ pre-compute the value of $\mathcal{O}_f$ on as many points, starting from $ctr_0 + 1$, as necessary, to answer all of $A$'s $\mathsf{guess}$ phase encryption oracle queries. These pre-computed values are in the set $\mathcal{F}$ which is passed to $A'$'s $\mathsf{guess}$ phase via state information $s$. Note that it is important that $A'$ did not query $\mathcal{O}_f$ with $ctr_0$, since otherwise it could not output $ctr_0$ as the point on which it gets its challenge. The counter mode guarantees that, as long as fewer than $\frac{\mu}{L}$ queries are made (i.e the counter does not loop around), the function will always be invoked on a new point.

The total number of oracle queries made by $A'$ is at most $\frac{\mu}{L}$, which by assumption is $q'$. Given this, one can check that the running time of $A'$ is as claimed. The advantage of $A'$ is given by,

$$\begin{aligned} \mathsf{Adv}^{\mathrm{iuf}}_{F}(A') &= \Pr\left[\, \mathsf{Exp}^{\mathrm{iuf}}_{F}(A',0)=0 \,\right] - \Pr\left[\, \mathsf{Exp}^{\mathrm{iuf}}_{F}(A',1)=0 \,\right] \\ &= \Pr\left[\, \mathsf{Exp}^{\mathrm{ind\text{-}cpa}}_{\mathrm{CTR}(F)}(A,0)=0 \,\right] + \Pr\left[\, \mathsf{Exp}^{\mathrm{ind\text{-}cpa}}_{\mathrm{CTR}(F)}(A,1)=1 \,\right] - \frac{1}{2} \\ &= \frac{1}{2}(1+\mathsf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathrm{CTR}(F)}(A)) - \frac{1}{2} \;=\; \frac{1}{2}\cdot \mathsf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathrm{CTR}(F)}(A) \end{aligned}$$

Given that $A$ was an arbitrary adversary, the relation between the advantage functions follows. ▌

# 5 Discussion

We stress that, in practice, the benefits of a tighter security analysis, as we have here, are real. For example, using the standard notion of a PRF, the security of a protocol may appear to be marginal, prompting the use of a larger security parameter. However, using a tighter characterization, such as IUF, the security might have been found to be adequate.

In criticism to our approach to getting tighter bounds for MACs and symmetric encryption schemes, one may suggest that we are looking at the wrong notions of security for these protocols. Indeed, there are alternate notions to the ones we assumed for which our gains would disappear. However, the notions of security we consider for both MACs and symmetric encryption are standard and a case can be made that these are, in fact, the more natural ones.

Future Directions. Unlike the case with the counter mode of encryption, in our first example we view the entire MAC as being the primitive, when in fact it too may be built on a PRF (for example, the CBC-MAC based on a block cipher). While it seems unlikely that we can achieve a tighter security analysis for the CBC-MAC scheme analyzing it this way, it may be possible for other MACs. Then there are other protocols, besides those for message authentication and symmetric encryption, to which our techniques could be applied. For example, it may be possible to improve the security bounds of variable-length input pseudorandom functions (VI-PRFs) [2] and variable-input-length ciphers [6].

Using similar techniques as above, we can also get tighter bounds for PRP-based protocols. In a sense, this is more interesting given that PRP families provide a more natural model for block ciphers [5]. Viewing a block cipher as a PRP family rather than a PRF family itself can also give tighter security bounds. However, we were motivated by the fact that analysis of block-cipher-based schemes is usually done modeling the block cipher as a PRF. This is because the analysis using PRFs is often significantly simpler.

Along those lines, we remark that it seems somewhat significant that there is a difference between function families and permutation families with respect to the indistinguishability of points characterization. This in fact is the only such distinction of which we are aware when using asymptotic measures. It may be interesting to investigate further the impact of injectivity upon pseudorandomness.

# Acknowledgements

We thank Mihir Bellare for his help and advice.

# References


[1] M. Bellare, R. Canetti and H. Krawczyk, "Keying hash functions for message authentication," *Advances in Cryptology – Crypto 96 Proceedings*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.

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

[3] M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation," Proceedings *of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.

[4] M. Bellare, R. Guérin and P. Rogaway, "XOR MACs: New methods for message authentication using finite pseudorandom functions," *Advances in Cryptology – Crypto 95 Proceedings*, Lecture Notes in Computer Science Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.

[5] M. Bellare, J. Kilian and P. Rogaway, "The security of the cipher block chaining message authentication code," *Advances in Cryptology – Crypto 94 Proceedings*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.

[6] M. Bellare and P. Rogaway, "On the construction of variable-input-length ciphers," *Proceedings of the 6th Workshop on Fast Software Encryption*, Ed. L.Knudsen, 1999.

[7] O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions. *Journal of the ACM*, Vol. 33, N0. 4, 1986, pp. 210-217.

[8] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Science*, Vol. 28, 1984, pp. 270–299.

[9] S. Goldwasser, S. Micali and R. Rivest, "A digital signature signature scheme secure against adaptive chosen-message attacks," *SIAM J. of Computing*, 17(2): 281-308, April 1988.

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

[11] M. Naor and O. Reingold, "From unpredictability to indistinguishability: A simple construction of PRFs from MACs," *Advances in Cryptology – Crypto 98 Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.


# A Remaining Proofs

In this section, we provide the proofs of results from Section 3.

**Proof of Theorem 3.1:** [PRF $\Rightarrow$ IUF] We use a standard contradiction argument to prove this. Assume that there exists a distinguisher $A_2$ for $F$ in the IUF sense. We construct a distinguisher $A_1$ for $F$ in the PRF sense, using $A_2$ as a subroutine.

Let $\mathcal{O}_f$ be $A_1$'s oracle. $A_1^{\mathcal{O}_f}$ will run $A_2$ using $\mathcal{O}_f$ to provide an appropriate simulation of $A_2$'s oracle, as indicated below.

Algorithm $A_1^{\mathcal{O}_f}$

(1) For $i = 1, \cdots, q_1$: on query $x_i$ from $A_2$, respond with $\mathcal{O}_f(x_i)$.
(2) Let $(x, s) \leftarrow A_2(\mathsf{find})$.
(3) Let $y_0 \leftarrow \mathcal{O}_f(x)$ and $y_1 \leftarrow \{0,1\}^L$.
(4) Let $b \leftarrow \{0,1\}$.
(5) Let $b' \leftarrow A_2(\mathsf{guess}, y_b, s)$.
(6) If $b' = b$ then output 0, else output 1.

Clearly, $q_1 = q_2$ and $t_1 = t_2 + O(l + L)$. Now, we will compute the advantage of $A_1$.

$$\begin{aligned} \mathsf{Adv}^{\mathrm{prf}}_F(A_1) &= \Pr\left[\mathsf{Exp}^{\mathrm{prf}}_F(A_1, 0) = 0\right] - \Pr\left[\mathsf{Exp}^{\mathrm{prf}}_F(A_1, 1) = 0\right] \\ &= \Pr\left[\mathsf{Exp}^{\mathrm{iuf}}_F(A_2, b) = b\right] - \frac{1}{2} = \frac{1}{2}\mathsf{Adv}^{\mathrm{iuf}}_F(A_2) \end{aligned}$$

Given that $A_2$ was any arbitrary distinguisher, the relation in the advantage functions follow. ▮

**Proof of Theorem 3.2:** [IUF → PRF] Assume that there exists a distinguisher $A_1$ for $F$ in the PRF sense. We construct a distinguisher $A_2$ for $F$ in the IUF sense, using $A_1$ as a subroutine.

Algorithm $A_2^{\mathcal{O}_f}$

(1) Select $p \leftarrow \{1, \cdots, q_1\}$.
(2) For $j = 1, 2, \cdots, p-1$: on query $x_j$, respond with $y_j \leftarrow \mathcal{O}_f(x_j)$.
(3) Let query $x_p$ be output as the challenge point; receive in response $y$.
(4) Return $y$ as the response to $A_1$'s query $x_p$.
(5) For $j = (p+1), (p+2), \cdots, q_1$: on query $x_j$, respond with $y_j \leftarrow \{0,1\}^L$.
(6) Output $A_1^{\mathcal{O}_f}$.

We want to analyze the advantage of $A_2$ in terms of the advantage of $A_1$. We use a standard hybrid argument for this. Consider the sequence of games $G_i$ (for $0 \le i < q_1$), defined below.

Experiment $G_i^{A_1}$
$a \leftarrow \mathsf{Keys}(F); \mathcal{O}_f \leftarrow F_a$.
For $j = 1, \cdots, i$: on $A_1$'s query $x_j$, answer $y_j \leftarrow \mathcal{O}_f(x_j)$.
For $j = i+1, \cdots, q-1$: on $A_1$'s query $x_j$, answer $y_j \leftarrow \{0,1\}^L$.
Return $A_1^{\mathcal{O}}$

Let $S_i$ be the probability that $G_i$ returns 0. Notice that if $b = 0$ then the response to $A_1$'s $(i+1)$th query is $\mathcal{O}_f(x_{i+1})$ and we have that $A_1$ is actually playing game $G_{i+1}$. On the other hand, if $b = 1$, then $A_1$ is actually playing game $G_i$, since the response to the $(i+1)$th query is actually a random string. Finally, since $p$ is chosen randomly from $\{1, \cdots, q_1\}$, we have:

$$\begin{aligned} \mathsf{Adv}^{\mathrm{iuf}}_F(A_2) &= \Pr\left[\mathsf{Exp}^{\mathrm{iuf}}_F(A_2, 0) = 0\right] - \Pr\left[\mathsf{Exp}^{\mathrm{iuf}}_F(A_2, 1) = 0\right] \\ &= \frac{1}{q_1}\sum_{i=1}^{q_1} S_i - \frac{1}{q_1}\sum_{i=0}^{q_1} S_{i-1} = \frac{1}{q_1}\sum_{i=1}^{q_1}(S_{i+1} - S_i) = \frac{1}{q_1}(S_{q_1} - S_0) \\ &= \frac{1}{q_1}(\Pr\left[\mathsf{Exp}^{\mathrm{prf}}_F(A_1, 0) = 0\right] - \Pr\left[\mathsf{Exp}^{\mathrm{prf}}_F(A_1, 1) = 0\right]) = \frac{1}{q_1}\mathsf{Adv}^{\mathrm{prf}}_F(A_1) \end{aligned}$$

It is clear from the description that $q_1 = q_2$ and $t_1 = t_2 - O(l + L)$. This completes the proof. ▮

**Proof of Proposition 3.3:** [$\mathsf{IUF} \not\Rightarrow \mathsf{PRF}$] Let $F'$ be the random function family. We know that for any $t, q \geq 0$, the advantage function $\mathsf{Adv}^{\mathrm{iuf}}_{F'}(t, q) = 0$. We use $F'$ to define a second function family $F$ that will have the claimed properties. Let $\mathsf{Keys}(F) = \{\mathsf{Keys}(F') \times \{1, \cdots, q\}\}$. For any value $i$, we let $i' = 0 \cdots 0 || i$, so that $i'$ has length $L$ bits. $F_{(a,i)}(x)$ is defined as $0^L$ if $x = i'$ and $F'_a(x)$ otherwise. Now consider the following distinguisher $A$ for $F$ in the $\mathsf{PRF}$ sense.

Algorithm $A^{\mathcal{O}}$

(1) For $i = 1, \cdots, q$ do: $y_i \leftarrow \mathcal{O}(i')$ where $i'$ is padded as earlier.
(2) If $y_i = O^L$, for some $1 \leq i \leq q$, then output 0 else output 1.

We lower bound the advantage of distinguisher $A$ in the $\mathsf{PRF}$ sense as follows.

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{prf}}_{F}(A) &= \Pr\left[\mathsf{Exp}^{\mathrm{prf}}_{F}(A, 0) = 0\right] - \Pr\left[\mathsf{Exp}^{\mathrm{prf}}_{F}(A, 1) = 0\right] \\
&\geq 1 - (1 - \Pr\left[(f(0) \neq 0^L \wedge f(1) \neq 0^L \wedge \cdots \wedge f(q) \neq 0^L) | f \leftarrow R_{l,L}\right] \\
&= 1 - (1 - (1 - \frac{1}{2^L})^q) \;=\; (1 - \frac{1}{2^L})^q
\end{aligned}$$

Thus, $\mathsf{Adv}^{\mathrm{prf}}_{F}(t, q) \geq (1 - \frac{1}{2^L})^q$. For $q \leq \frac{1}{2}2^L$ we have $\mathsf{Adv}^{\mathrm{prf}}_{F}(t, q) \geq 0.5$.

Next, we assume we have some distinguisher $B'$ for $F$ in the $\mathsf{IUF}$ sense, and we build an algorithm $B$ for $F'$ in the $\mathsf{IUF}$ sense, with oracle $\mathcal{O} = F'_a$ which simulates the function $F_{a,i}$ for $A'$.

Algorithm $B^{\mathcal{O}}$

(1) Select $i \leftarrow \{1, \cdots, q\}$. Compute $i'$ as described above.
(2) Run $B'(\mathsf{find})$, answering a query $u$ as: if $u = i'$, reply $0^L$, else reply $\mathcal{O}(u)$.
(3) Let $(x, s) \leftarrow B'(\mathsf{find})$.
(4) Output $(x, s)$ and receive as challenge $y$.
(5) Let $d \leftarrow B'(\mathsf{guess}, y, s)$.
(6) Output $d$.

One can check that the probability that $x = i'$ is at most $\frac{1}{q}$. It follows that, for any $t, q \geq 0$, $\mathsf{Adv}^{\mathrm{iuf}}_{F'}(t', q) \geq \mathsf{Adv}^{\mathrm{iuf}}_{F}(t, q) - \frac{1}{q}$ where $t' = t + O(l + L)$. By our choice of $F'$ we get $\mathsf{Adv}^{\mathrm{iuf}}_{F}(t, q) \leq \frac{1}{q}$, for any $t, q \geq 0$. Combining this with the lower bound derived earlier, we get the claimed result. ▮

**Proof of Theorem 3.4:** [$\mathsf{IUF} \Rightarrow \mathsf{IPF}$] Let $A_3$ be a distinguisher in the $\mathsf{IPF}$ sense. We construct a distinguisher $A_2$ in the $\mathsf{IUF}$ sense using $A_3$.

Algorithm $A_2^{\mathcal{O}_f}$

(1) For $i = 1, \cdots, q_3$: on query $x_i$ from $A_3$, respond with $\mathcal{O}_f(x_i)$.
(2) Let $(x_0, x_1, s) \leftarrow A_3(\mathsf{find})$.
(3) Let $b \leftarrow \{0, 1\}$.
(4) Output $x_b$ as challenge and receive $y$ as the challenge response.
(5) $d \leftarrow A_3(\mathsf{guess}, y, s)$.
(6) If $d = b$, output 0, else output 1.

$$
\begin{aligned}
\mathsf{Adv}_F^{\mathrm{iuf}}(A_2) &= \Pr\left[\,\mathsf{Exp}_F^{\mathrm{iuf}}(A_2,0)=0\,\right] - \Pr\left[\,\mathsf{Exp}_F^{\mathrm{iuf}}(A_2,1)=0\,\right] \\
&= \Pr\left[\,\mathsf{Exp}_F^{\mathrm{iuf}}(A_2,b)=b\,\right] - \frac{1}{2} \;=\; \frac{1}{2}\mathsf{Adv}_F^{\mathrm{ipf}}(A)
\end{aligned}
$$

Clearly, $q_2 = q_3$ and $t_2 = t_3 + O(l)$. ▮

**Proof of Proposition 3.5:** [IPF $\not\Rightarrow$ IUF] We justify the proposition by giving a function family that is secure in the IPF sense but completely insecure in the IUF sense. Consider $F_a(x) = 0^L$ for all values of $a$ and $x$. Clearly, a distinguisher in the IPF sense could not distinguish between $F_a(x_0)$ and $F_a(x_1)$ with probability greater than $\frac{1}{2}$ for any pair $(x_0, x_1)$. However, it is easy to see that with overwhelming probability, a distinguisher in the IUF sense would be able to do so. ▮

**Proof of Theorem 3.6:** [PRP $\Rightarrow$ IUP] The details of this proof are omitted since it is similar to the proof given for Theorem 3.1. ▮

**Proof of Theorem 3.7:** [IUP $\rightarrow$ PRP] The details of this proof are omitted since it is similar to the proof given for Theorem 3.2. ▮

**Proof of Proposition 3.8:** [IUP $\not\Rightarrow$ PRP] The proof of this follows that given for Theorem 3.3 for the most part. The gap-exhibiting function $F$ is defined slightly differently here. Using the same notation as in that proof, the function $F_{(a,i)}(x)$ here is defined as being $0^l$ if $x = i'$ and $F'_a(i')$ if $F'_a(x) = 0^l$ and $F'_a(x)$ otherwise. The analysis is similar. ▮

**Proof of Theorem 3.9:** [IUP $\Rightarrow$ IPP] The details of this proof are omitted since it is similar to the proof given for Theorem 3.4. ▮

**Proof of Theorem 3.10:** [IPP $\Rightarrow$ IUP] Let $A_2$ be a distinguisher in the IUP sense. We construct a distinguisher $A_3$ in the IPP sense using $A_2$.

Algorithm $A_3^{\mathcal{O}_f}$

(1) For $i = 1, \cdots, q_2$: on query $x_i$ from $A_2$, respond with $\mathcal{O}_f(x_i)$.
(2) Let $(x_0, s) \leftarrow A_2(\mathsf{find})$.
(3) Select $x_1 \leftarrow \{0,1\}^l \setminus \{\mathcal{O}_f(x_i) : 1 \le i \le q_2\}$.
(4) Output $(x_0, x_1, s)$ and receive $y$ as the challenge.
(5) $d \leftarrow A_2(\mathsf{guess}, y, s)$.
(6) Output $d$.

$$
\begin{aligned}
\mathsf{Adv}_F^{\mathrm{ipp}}(A_3) &= \Pr\left[\,\mathsf{Exp}_F^{\mathrm{ipp}}(A_3,0)=0\,\right] - \Pr\left[\,\mathsf{Exp}_F^{\mathrm{ipp}}(A_3,1)=0\,\right] \\
&= \Pr\left[\,\mathsf{Exp}_F^{\mathrm{iup}}(A_2,0)=0\,\right] - \Pr\left[\,\mathsf{Exp}_F^{\mathrm{iup}}(A_2,1)=0\,\right] \;=\; \mathsf{Adv}_F^{\mathrm{iup}}(A)
\end{aligned}
$$

Clearly, $q_3 = q_2$ and $t_3 = t_2 + O(l)$. This completes the proof. ▮