Technical Report CS97-529, Department of Computer Science and Engineering, University of California at San Diego, March 1997. Revised March 2002.

A Note on Negligible Functions

Mihir Bellare*

Dept. of Computer Science & Engineering
University of California at San Diego
9500 Gilman Drive
La Jolla, California 92093, USA.

E-Mail: mihir@cs.ucsd.edu.
URL: http://www-cse.ucsd.edu/users/mihir.

March 1997

Abstract

In theoretical cryptography, one formalizes the notion of an adversary's success probability being "too small to matter" by asking that it be a negligible function of the security parameter. We argue that the issue that really arises is what it might mean for a *collection* of functions to be "negligible." We consider (and define) two such notions, and prove them equivalent. Roughly, this enables us to say that any cryptographic primitive has a *specific* associated "security level." In particular we say this for any one-way function. We also reconcile different definitions of negligible error arguments and computational proofs of knowledge that have appeared in the literature. Although the motivation is cryptographic, the main result is purely about negligible functions.

*Supported in part by NSF Grant CCR-0098123, a Packard Foundation Fellowship in Science and Engineering, and an IBM Faculty Partnership Development Award.

Contents

1 Introduction

A function $g\colon \mathbb{N} \to \mathbb{R}$ is called *negligible* if it approaches zero faster than the reciprocal of any polynomial. That is, for every $c \in \mathbb{N}$ there is an integer n_c such that $g(n) \le n^{-c}$ for all $n \ge n_c$. In theoretical cryptography, one formalizes the notion of an adversary's success probability being "too small to matter" by asking that it be a negligible function of the security parameter.

In this note we point out that there are two possible ways to formalize the notion of a cryptographic primitive being secure based on the negligibility of success probabilities of polynomial-time adversaries. Roughly, the difference is in whether the primitive has a single "security level," or a different one for each adversary. Both formalizations have been used in the literature. We ask how these formalizations relate to each other.

We show that the underlying technical question has nothing to do with cryptography. It can be captured by defining two notions of negligibility for a *collection* of functions and asking how they relate to each other. We define the notions and show that they are equivalent.

Roughly, this implies that to any cryptographic primitive we can associate a single function that is its "security level," rather than having a different security level for each adversary. In the case of negligible error arguments and computational proofs of knowledge with negligible knowledge error, this reconciles two different definitions that have appeared in the literature. To illustrate the issues, however, let us begin by looking at a more basic primitive, namely a one-way function.

1.1 The issue for one-way functions

Two notions. Let $f\colon \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable, length preserving function. An *inverter* for f is a probabilistic, polynomial time algorithm I. (We will discuss later the non-uniform case, where an inverter is a family of circuits of polynomial size.) To any inverter I we associate a function $\mathbf{Inv}_I$ called its success probability, defined for any value $n \in \mathbb{N}$ of the security parameter by $\mathbf{Inv}_I(n) = \Pr[\, f(I(f(x))) = f(x) \,]$, the probability being over a random choice of x from $\{0,1\}^n$, and over the coin tosses of I. The following is standard:

> We say f is *one-way* if for every inverter I the function $\mathbf{Inv}_I$ is negligible.

There is another way we might consider formalizing f being "one-way". To describe this, we first introduce the following terminology and notation. We say that $g_1\colon \mathbb{N} \to \mathbb{R}$ is *eventually less than* $g_2\colon \mathbb{N} \to \mathbb{R}$, written $g_1 \le_{\rm ev} g_2$, if there is an integer k such that $g_1(n) \le g_2(n)$ for all $n \ge k$. Now:

> We say f is *uniformly one-way* if there is a negligible function δ such that $\mathbf{Inv}_I \le_{\rm ev} \delta$ for every inverter I.

In other words, there is a negligible function δ that is a "witness" to the fact that the success probability of any inverter eventually becomes "small." More precisely, for each inverter I there is an integer k_I such that $\mathbf{Inv}_I(n) \le \delta(n)$ for all $n \ge k_I$. We call $s(\cdot) = 1/\delta(\cdot)$ the "security level."[1]

Discussion. Another way of viewing the above definitions is that the order of quantification is different:

$$\begin{array}{rcl} f \text{ is one-way} & : & \forall \text{ inverters } I \ \exists \text{ negligible } \delta_I \text{ such that } \mathbf{Inv}_I \le_{\rm ev} \delta_I \\ f \text{ is uniformly one-way} & : & \exists \text{ negligible } \delta \text{ such that } \forall \text{ inverters } I \text{ we have } \mathbf{Inv}_I \le_{\rm ev} \delta \end{array}$$

[1] In asking for a single "security level", this definition is in the style of the definition of Levin [7]. (See also Luby [8]). The latter however measure the quality of inverters via their time to success probability ratios. It seems the notion of uniform one-wayness is a simplified, special case of their notions in which one looks only at polynomial time adversaries and security of the form $1/\delta$ for a negligible function δ.

Yet another way to see the difference is by taking the contra-positives of the definitions, as we must do in proving theorems based on the assumption that f is one-way or uniformly one-way. Function f is *not one-way* if there is an inverter I and a constant c such that $\mathbf{Inv}_I(n) > n^{-c}$ for infinitely many n. That is, there is some inverter whose success probability is not negligible. On the other hand, f is *not uniformly one-way* if for *every* negligible δ there is an inverter I_δ such that $\mathbf{Inv}_{I_\delta}(n) > \delta(n)$ for infinitely many n. This does not directly say that there is one inverter achieving non-negligible success.

Is there a single security level? It is not hard to see that if f is uniformly one-way then it is one-way, but it is not clear whether or not the converse is true. Perhaps different inverters have different success probabilities, all negligible, but so that for any particular negligible δ, there is some inverter who does better than δ infinitely often.

Equivalence. The results in this paper imply that the above two definitions are equivalent, meaning f is one-way if and only if it is uniformly one-way. This means that given a one-way function f there exists a single negligible function δ such that the success probability of any inverter is eventually less than δ. So every one-way function does have a "specific" associated security level. In other words, the order of the quantifiers does not matter.

1.2 Negligibility of function collections

It turns out the technical question underlying the above has nothing to do with one-way functions, or even with cryptography. It is just about negligible functions. Let us first formulate the question, then relate it to the above.

Let $\boldsymbol{F} = \{ F_i : i \in \mathbb{N} \}$ be a collection of functions, all mapping $\mathbb{N}$ to $\mathbb{R}$. We will consider two definitions of "negligibility" for the collection $\boldsymbol{F}$. The first is simple: just ask that each function, taken individually, is negligible. Formally, we say $\boldsymbol{F}$ is *pointwise negligible* if F_i is negligible for each $i \in \mathbb{N}$. The second is to ask that the collection is "uniformly" negligible in that the different functions conform to some common limit point. Formally, $\boldsymbol{F}$ is *uniformly negligible* if there is a negligible function δ (called a limit point) such that $F_i \leq_{\mathrm{ev}} \delta$ for all $i \in \mathbb{N}$. That is, each F_i drops below δ for large enough n. (The terminology here is by some sort of rough analogy with the notions of pointwise and uniform convergence of collections of functions in real analysis.)

It is quite easy to see that if collection $\boldsymbol{F}$ is uniformly negligible then it is also pointwise negligible. But is the converse true? Theorem 3.2 shows that the answer is yes: the two notions of negligible collections are equivalent.

We stress that the collections considered here are countable. The result is not true for an uncountable collection.

1.3 Application to cryptographic notions

Application to one-way functions. Now, how does this relate to the issue for one-way functions? Let $\mathcal{I} = \langle I_i : i \in \mathbb{N} \rangle$ be an enumeration of all inverters. (Since an inverter is a probabilistic, polynomial-time algorithm, the number of inverters is countable. For the non-uniform case, where there are uncountably many inverters, see Section 1.4 and Section 4.) For each $i \in \mathbb{N}$ define the function F_i by $F_i(n) = \mathbf{Inv}_{I_i}(n)$, the latter being the success probability of I_i as defined in Section 1.1. Let $\boldsymbol{F} = \{ F_i : i \in \mathbb{N} \}$. Then observe that f is one-way if and only if the collection $\boldsymbol{F}$ is pointwise negligible, and f is uniformly one-way if and only if the collection $\boldsymbol{F}$ is uniformly negligible. Theorem 3.2 thus implies that f is one-way if and only if it is uniformly one-way.

More generally. Now that we see this, it is clear the same is true for pretty much any cryptographic primitive. The (asymptotic) definition of security for any primitive has the following form.

To any "adversary" A and any value of the security parameter n there will be associated a success probability $\mathbf{Succ}_A(n)$, under some experiment. (For now, an adversary is a uniform algorithm.) The primitive will be said to be "secure" if for each adversary A the function $\mathbf{Succ}_A$ is negligible. To put this in the framework we have been looking at, let $\boldsymbol{A} = \{ A_i : i \in \mathbb{N} \}$ be an enumeration of all adversaries in question. (Since an adversary is a uniform algorithm, the number of adversaries is countable.) Let $F_i(n) = \mathbf{Succ}_{A_i}(n)$. Then we see that the definition indicated above is asking that the collection of functions $\boldsymbol{F} = \{ F_i : i \in \mathbb{N} \}$ is pointwise negligible. It seems equally reasonable, however, to ask that the collection of functions be uniformly negligible. This says there exists a particular negligible function δ such that the success probability $\mathbf{Succ}_A(\cdot)$ of any adversary A is eventually less than δ. Here, a specific security level is associated to the primitive, to which all adversaries must eventually conform. This might seem different, but Theorem 3.2 says the two notions are equivalent. In particular it says that it is always possible to find such a specific security level for any primitive even if the definition does not explicitly ask for it.

ERROR PROBABILITIES IN PROTOCOLS. For a one-way function, it seems of some interest that there is a single "security level" associated to the function, but we do not see any particular advantage to actually formulating the definition in the new way. However, a setting where the second type of formulation seems more natural is in computationally sound proofs and proofs of knowledge.

We often talk of "the error probability" of a protocol such as a computationally sound interactive proof. This terminology indicates that we imagine there being associated to a given interactive proof system, defined by a given verifier, a single entity (function) called its error-probability. A definition of "negligible error arguments" based on this view is given in [2]. Earlier, however, other definitions had appeared which did not have this view of error probability in the case of negligible error [5, 9]: each prover had a different associated "error-probability," so that the term "the error-probability" of the protocol did not have a realization. Applying the above however we can show that the two formulations are equivalent. See Section 4.2. Similarly, we relate two notions of computational proofs of knowledge with negligible knowledge error suggested in [1]. See Section 4.3.

1.4 Non-uniform adversaries and uncountability

As we indicated above, we wish to associate a function to each adversary, this function being the adversary's success probability, and consider the "negligibility" of the ensuing collection of functions. When the class of adversaries includes only uniform algorithms, the number of adversaries, and hence of functions, is countable, and the result mentioned in Section 1.2 applies. However, the set of adversaries might be the set of all *non-uniform* polynomial time algorithms. This set is uncountable, and hence so is the collection of associated functions. In this case, we cannot directly apply our main result. However, we will see that it is still possible to apply the equivalence, and get the desired results, by considering the "best possible" non-uniform adversaries for each specific polynomial size-bound. This will "reduce" the uncountable case to the countable one.

The treatment in Section 2 is general, applying to either countable or uncountable collections. We prove in Section 3 the equivalence in the countable case, and also a characterization, for the uncountable case, that enables us to reduce the latter to the former.

2 Definitions and elementary facts

Let $\mathbb{N} = \{1, 2, 3 \ldots\}$ be the set of positive integers, and $\mathbb{R}$ the reals. Unless otherwise indicated, a function maps $\mathbb{N}$ to $\mathbb{R}$. We sometimes regard a function as a "point" in the space of all functions and refer to it this way. An "integer" means a positive integer, ie. an element of $\mathbb{N}$. We begin with a useful shorthand:

Definition 2.1 If f, g are functions we say that f is *eventually less than* g, written $f \leq_{\mathrm{ev}} g$, if there is an integer k such that $f(n) \leq g(n)$ for all $n \geq k$. ▮

It is useful to note that this relation is transitive:

Proposition 2.2 The relation $\leq_{\mathrm{ev}}$ is transitive: if $f_1 \leq_{\mathrm{ev}} f_2$ and $f_2 \leq_{\mathrm{ev}} f_3$ then $f_1 \leq_{\mathrm{ev}} f_3$. ▮

Recall that a function f is negligible if for every integer c there is an integer n_c such that $f(n) \leq n^{-c}$ for all $n \geq n_c$. With the above shorthand, another way to say it is:

Definition 2.3 A function f is *negligible* if $f \leq_{\mathrm{ev}} (\cdot)^{-c}$ for every integer c. ▮

Here $(\cdot)^{-c}$ stands for the function $n \mapsto n^{-c}$. It is useful to note the following:

Proposition 2.4 A function f is negligible if and only if there is a negligible function g such that $f \leq_{\mathrm{ev}} g$. ▮

Proof of Proposition 2.4: If f is negligible then the other condition is satisfied by setting $g = f$. Conversely assume there is a negligible g such that $f \leq_{\mathrm{ev}} g$. We want to show f is negligible. So let $c \in \mathbb{N}$. Then $f \leq_{\mathrm{ev}} g$ (by assumption) and $g \leq_{\mathrm{ev}} (\cdot)^c$ (because g is negligible) so by Proposition 2.2 we have $f \leq_{\mathrm{ev}} (\cdot)^c$. So f is negligible by Definition 2.3. ▮

A *collection of functions* is a set of functions whose cardinality could be countable or uncountable.

Definition 2.5 A collection of functions $\boldsymbol{F}$ is *pointwise negligible* if for every $F \in \boldsymbol{F}$ it is the case that F is a negligible function. ▮

This means that for each $F \in \boldsymbol{F}$ and each integer c there is some number $k(F, c)$, depending on both F and c, such that $F(n) \leq n^{-c}$ whenever $n \geq k(F, c)$. In other words, the different functions could go down at different rates, and although each is eventually below any inverse polynomial, the time at which this happens depends both on the function and the value of c defining the inverse polynomial. The notion we define next is stronger, in that it asks that there be a single negligible function δ that is a "witness" to the fact that the functions in the collection eventually become small. All functions must eventually drop below δ.

Definition 2.6 A collection of functions $\boldsymbol{F}$ is *uniformly negligible* if there is a negligible function δ such that $F \leq_{\mathrm{ev}} \delta$ for every $F \in \boldsymbol{F}$. ▮

In other words, the collection $\boldsymbol{F}$ is uniformly negligible if there is a negligible function δ such that for each $F \in \boldsymbol{F}$ there is an integer $k(F)$ such that $F(n) \leq \delta(n)$ for all $n \geq k(F)$. Notice that the point at which F drops below δ is allowed to depend on F and may vary from function to function in the collection.

Definition 2.7 Let $\boldsymbol{F}$ be a collection of functions and let δ be a function. We say that δ is a *limit point* of $\boldsymbol{F}$ if $F \leq_{\mathrm{ev}} \delta$ for each $F \in \boldsymbol{F}$. ▮

The following is obvious:

Proposition 2.8 A collection of functions $\boldsymbol{F}$ is uniformly negligible if and only if it has a negligible limit point. ▮

Note that limit points are not unique.

3 Relations between the two notions of negligible collections

It is easy to see that uniform negligibility implies pointwise negligibility. This is true regardless of whether the collection is countable or uncountable.

Proposition 3.1 If $\boldsymbol{F}$ is uniformly negligible then it is pointwise negligible.

Proof: By Proposition 2.8 there exists a negligible function δ that is a limit function for $\boldsymbol{F}$. Let $F \in \boldsymbol{F}$. We know that $F \leq_{\mathrm{ev}} \delta$ since δ is a limit point of $\boldsymbol{F}$, and we know that δ is negligible, so F is negligible by Proposition 2.4. So $\boldsymbol{F}$ is pointwise negligible as per Definition 2.5. ▮

The question we want to look at is whether the notions are equivalent. We first consider the case where the collection of functions is countable, and then the case where it is uncountable.

3.1 The case of a countable collection of functions

The important case is when the collection is countable. In that case we show that the two notions of negligibility are equivalent.

Theorem 3.2 Let $\boldsymbol{F} = \{ F_i \,:\, i \in \mathbb{N} \}$ be a countable collection of functions. Then $\boldsymbol{F}$ is pointwise negligible if and only if it is uniformly negligible. ▮

We know from Proposition 3.1 that if $\boldsymbol{F}$ is uniformly negligible then it is pointwise negligible. The other direction is more interesting. The assumption is that each F_i is a negligible function. We claim that $\boldsymbol{F}$ is uniformly negligible. To show this we will define a negligible limit point δ for $\boldsymbol{F}$. Before doing so, it may help to see why a tempting easier construction does not work.

Remark 3.3 Perhaps the first thought would be to set

$$\delta(n) \;=\; \max\{ F_1(n), F_2(n), \ldots, F_n(n) \} \,. \tag{1}$$

Certainly $F_i \leq_{\mathrm{ev}} \delta$ for each $i \in \mathbb{N}$. But it is not hard to see that δ need not be negligible. For example let ν be a negligible function and set $F_i(j) = 1$ if $j \leq i$ and $F_i(j) = \nu(j)$ if $j > i$. The collection of functions $\{ F_i \,:\, i \in \mathbb{N} \}$ is pointwise negligible, but the function δ of Equation (1) is the constant function 1 which is definitely not negligible. ▮

Proof of Theorem 3.2: Assume $\boldsymbol{F}$ is pointwise negligible. We will construct a negligible limit point δ for $\boldsymbol{F}$. The construction uses diagonalization. Let us first sketch the idea and then provide the details.

Imagine a table with rows indexed by the values $i = 1, 2, \ldots$; columns indexed by the values of $n = 1, 2, \ldots$; and entry (i, n) of the table containing $F_i(n)$. We know that for any c, the entries in each row eventually drop below n^{-c}. But where it happens differs from row to row. We define δ by a sort of diagonalization, in a sequence of "stages." In stage c we will consider only the first c functions in the list, namely $F_1, \ldots, F_c$. We will find a value $h(c)$, such that all these functions are less than $(\cdot)^{-c}$ for $n \geq h(c)$, by "moving out" as much as is necessary for all c functions to fall below our target. We view this as defining a sequence of rectangles, each finite, but so that the sequence eventually covers the entire table. (For a vague illustration, see Figure 1.) We will use h to define δ. Namely for each n we define $\delta(n)$ to maximize the functions in the rectangle with column number "closest" to n. Let us now give the construction and proof in more detail.

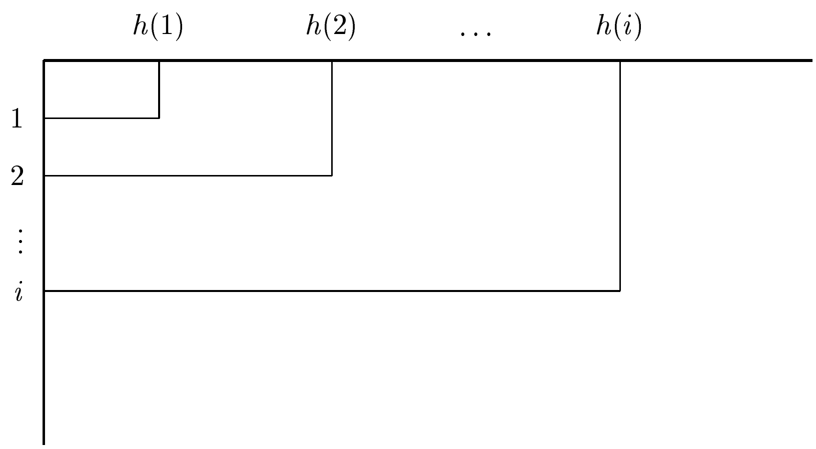

Figure 1: Entry (i,n) of this table is $F_i(n)$. To each row number i we associate a column number $h(i,i)$ such that all entries to the right of the corresponding rectangle (meaning stay above the bottom edge!) are bounded by n^{-i}.

For every $i, c \in \mathbb{N}$ we know that $F_i \leq_{\text{ev}} (\cdot)^{-c}$. Let $N(i,c) \in \mathbb{N}$ be such that $F_i(n) \leq n^{-c}$ for all $n \geq N(i,c)$. We now define a function h: $\{0\} \cup \mathbb{N} \to \mathbb{N}$ recursively as follows. Let $h(0) = 0$, and for $c \in \mathbb{N}$ let

$$h(c) \ = \ \max\{\, N(1,c), N(2,c), \ldots, N(c,c),\ 1 + h(c-1)\,\}\ . \tag{2}$$

That is, $h(c)$ is a point beyond which the first c functions drop below $(\cdot)^{-c}$. The following two claims are clear from the definition:

Claim 1: $F_1(n), \ldots, F_c(n) \leq n^{-c}$ for all $n \geq h(c)$ and all $c \in \mathbb{N}$. $\Box$

Claim 2: h is an increasing function, meaning $h(c) < h(c+1)$ for all $c \in \mathbb{N} \cup \{0\}$. $\Box$

For any $n \in \mathbb{N}$ we let

$$g(n) \ = \ \max\{\, j \in \mathbb{N} \ : \ h(j) \leq n \,\}\ . \tag{3}$$

That is, the first $g(n)$ functions drop below $(\cdot)^{-g(n)}$ for inputs that are at least n. Note the fact that h is increasing means that the set in the above maximization is finite, so the maximum is well defined. The following is clear from Equation (3):

Claim 3: g is a non-decreasing function, meaning $g(n) \leq g(n+1)$ for all $n \in \mathbb{N}$. $\Box$

Intuitively, we think of g as an inverse of function h. The precise relationship is provided by Claims 4 and 5 below. Claim 4 is clear from Equation (3):

Claim 4: $h(g(n)) \leq n$ for all $n \in \mathbb{N}$. $\Box$

Letting $n = h(c)$ in Equation (3) and using Claim 2, we also get:

Claim 5: $g(h(c)) = c$ for all $c \in \mathbb{N}$. $\Box$

Now for any $n \in \mathbb{N}$ we let

$$\delta(n) \ = \ \max\{\, F_i(n) \ : \ 1 \leq i \leq g(n) \,\}\ . \tag{4}$$

We have two final claims, to be proven below:

Claim 6: The function δ is a limit point of the collection $\boldsymbol{F} = \{ F_i : i \in \mathbb{N} \}$. □

Claim 7: The function δ is negligible.

Claims 6 and 7 together say that δ is a negligible limit point for the collection $\boldsymbol{F} = \{ F_i : i \in \mathbb{N} \}$, and hence $\boldsymbol{F}$ is uniformly negligible by Proposition 2.8, completing the proof. It remains to prove Claims 6 and 7.

To prove Claim 6, let $i \in \mathbb{N}$. As per Definition 2.7 we need to show there is an integer n_i such that $F_i(n) \leq \delta(n)$ for all $n \geq n_i$. We set $n_i = h(i)$ and claim this works. Indeed, suppose $n \geq h(i)$. Applying first Claim 3 and then Claim 5 we get

$$g(n) \geq g(h(i)) = i \, .$$

From Equation (4) it follows that $F_i(n) \leq \delta(n)$, as desired.

To prove Claim 7, we need to show that δ meets Definition 2.3. So let $c \in \mathbb{N}$. We need to show that there is an integer n_c such that $\delta(n) \leq n^{-c}$ for all $n \geq n_c$. We set $n_c = h(c)$ and claim this works. To see this, assume $n \geq h(c)$. The following is justified below:

$$\begin{aligned} \delta(n) &= \max \{ F_i(n) : 1 \leq i \leq g(n) \} \\ &\leq n^{-g(n)} \\ &\leq n^{-c} \, . \end{aligned}$$

The first line is from Equation (4). Claim 4 tells us that $n \geq h(g(n))$, and Claim 1 then gives us the second line above. Since we assumed $n \geq h(c)$, applying first Claim 3 and then Claim 5 we get

$$g(n) \geq g(h(c)) = c \, .$$

This implies $n^{-g(n)} \leq n^{-c}$ which was the last line above. ▮

Remark 3.4 The limit point δ constructed in Theorem 3.2 has properties beyond being a negligible limit point. In particular, it is a non-increasing function, meaning $\delta(n) \leq \delta(n+1)$ for all $n \in \mathbb{N}$. ▮

3.2 The case of an uncountable collection of functions

The collection of functions considered in Theorem 3.2 is countable. Proposition 3.1 says that even an uncountable collection of uniformly negligible functions is pointwise negligible. But the converse fails for some uncountable collections.

Proposition 3.5 There is an uncountable collection of functions $\boldsymbol{F}$ that is pointwise negligible but not uniformly negligible.

Proof: Let $\boldsymbol{F}$ be the set of all negligible functions mapping $\mathbb{N}$ to $\mathbb{R}$. Obviously $\boldsymbol{F}$ is pointwise negligible. But it is not uniformly negligible. To see this, let g be any negligible function. It cannot be a limit point of $\boldsymbol{F}$, because the function $f = 2g$ is negligible, hence in $\boldsymbol{F}$, but f is not eventually less than g. Thus no negligible function can be a limit point for $\boldsymbol{F}$, so that $\boldsymbol{F}$ has no negligible limit point. ▮

This does not mean that *all* uncountable collections of pointwise negligible functions fail to be uniformly negligible. The following is a simple characterization of collections where the equivalence holds.

Definition 3.6 Let $\boldsymbol{F}, \boldsymbol{M}$ be collections of functions. We say that $\boldsymbol{F}$ is *majored by* $\boldsymbol{M}$, or $\boldsymbol{M}$ *majors* $\boldsymbol{F}$, if for every $F \in \boldsymbol{F}$ there is an $M \in \boldsymbol{M}$ such that $F \leq_{\text{ev}} M$. ▮

The following characterization holds for any collection $\boldsymbol{F}$, but the interesting case is when $\boldsymbol{F}$ is uncountable. The key point below is that the collection that majors $\boldsymbol{F}$ is required to be countable.

Theorem 3.7 Let $\boldsymbol{F}$ be a collection of functions. Then $\boldsymbol{F}$ is uniformly negligible if and only if it is majored by some pointwise negligible, countable collection of functions.

Proof: First assume $\boldsymbol{F}$ is uniformly negligible. By Proposition 2.8 it has a negligible limit point δ. We set $\boldsymbol{M} = \{ m\delta \, : \, m \in \mathbb{N} \}$. This is a countable, pointwise negligible collection of functions, and it majors $\boldsymbol{F}$ because it contains δ. So $\boldsymbol{F}$ is indeed majored by some pointwise negligible countable collection of functions.

Conversely suppose $\boldsymbol{M}$ is a pointwise negligible, countable collection of functions that majors $\boldsymbol{F}$. Since $\boldsymbol{M}$ is countable, it is uniformly negligible by Theorem 3.2. By Proposition 2.8, $\boldsymbol{M}$ has a negligible limit point δ. Now if $F \in \boldsymbol{F}$ then by Definition 3.6 there is some $M \in \boldsymbol{M}$ such that $F \leq_{\text{ev}} M$. But $M \leq_{\text{ev}} \delta$ because $M \in \boldsymbol{M}$ and δ is a limit point for $\boldsymbol{M}$. Thus, $F \leq_{\text{ev}} \delta$ by Proposition 2.2 and δ is also a limit point for $\boldsymbol{F}$. So $\boldsymbol{F}$ is uniformly negligible. ▮

Recall that we want to make the functions in the collection correspond to success probabilities of adversaries. We have discussed in Section 1.4 how the countability or uncountability of the collection is a question of whether uniform or non-uniform adversaries are being considered. Although it is not possible to directly apply Theorem 3.2 in the latter case, we will see that it is possible to apply Theorem 3.7, and get the desired results.

4 Application to cryptographic definitions

We discussed in Section 1.3 how the above relates to cryptographic definitions. Let us look at this in more detail. We first summarize the implications for one-way functions and then move on to arguments and proofs of knowledge.

Below, a *uniform* adversary is a probabilistic, polynomial-time (PPT) algorithm. A *non-uniform* adversary $A = \langle A_i : i \in \mathbb{N} \rangle$ is a sequence of circuits of polynomial size (meaning, there is a polynomial p such that for all i the size of A_i is at most $p(i)$), and in this case the notation $A(x)$ denotes the output of circuit $A_{|x|}$ on input x. An adversary means either a uniform or a non-uniform adversary.

4.1 Application to one-way functions

Let $f\colon \{0,1\}^* \to \{0,1\}^*$ be a polynomial time computable, length preserving function. An adversary in this context is called an inverter. Associated to any inverter I (uniform or non-uniform) is its success probability function $\mathbf{Inv}_I\colon \mathbb{N} \to \mathbb{R}$, defined for all $n \in \mathbb{N}$ by $\mathbf{Inv}_I(n) = \Pr[\, f(I(f(x))) = f(x) \,]$, the probability being over the choice of x, and, in the uniform case, over the coins of I. We let $\mathcal{I}$ denote the set of all inverters. (In the uniform case, this is the countable set of all PPT algorithms, and in the non-uniform case the uncountable set of all sequences of circuits that have polynomial size.) We consider two definitions of one-wayness.

Definition 4.1 Let $f, \mathcal{I}$ be as above. We say that f is *one-way* if for every inverter $I \in \mathcal{I}$ the function $\mathbf{Inv}_I$ is negligible. We say that f is *uniformly one-way* if there is a negligible function δ such that $\mathbf{Inv}_I \leq_{\text{ev}} \delta$ for every inverter $I \in \mathcal{I}$. ▮

We claim the notions are equivalent. The following applies to both the uniform and the non-uniform cases:

Theorem 4.2 Let f be as above. Then f is one-way if and only if it is uniformly one-way. ▮

Proof: We let $\boldsymbol{F} = \{ \mathbf{Inv}_I : I \in \mathcal{I} \}$ denote the collection of success probability functions associated to the set of inverters under consideration. This collection is countable in the uniform case and uncountable in the non-uniform case. The key observation is that f is one-way if and only if $\boldsymbol{F}$ is pointwise negligible, and f is uniformly one-way if and only if $\boldsymbol{F}$ is uniformly negligible. To complete the proof it suffices to show that $\boldsymbol{F}$ is pointwise negligible if and only if it is uniformly negligible.

In the uniform case case $\boldsymbol{F}$ is countable, so the conclusion follows directly from Theorem 3.2. We now consider the non-uniform case.

Proposition 3.1 says that if $\boldsymbol{F}$ is uniformly negligible then it is pointwise negligible. To prove the converse, assume $\boldsymbol{F}$ is pointwise negligible. We will exhibit a countable, pointwise negligible collection $\boldsymbol{M}$ that majors $\boldsymbol{F}$. It follows from Theorem 3.7 that $\boldsymbol{F}$ is uniformly negligible, and our proof is complete. It remains to exhibit $\boldsymbol{M}$.

For any integers n, s there are finitely many n-input circuits of size at most s, and hence we can fix an n-input circuit $B_{n,s}$ of size at most s such that for all n-input circuits C of size at most s we have

$$\Pr[\, f(B_{n,s}(f(x))) = f(x) \,] \;\geq\; \Pr[\, f(C(f(x))) = f(x) \,] \;,$$

the probability above being over a random choice of x from $\{0,1\}^n$. Let $p_1, p_2, \ldots$ be an enumeration of all polynomials, and for any $i \in \mathbb{N}$ define the non-uniform inverter $I_i = \langle\, B_{n,p_i(n)} \,:\, n \in \mathbb{N} \,\rangle$. For any $n \in \mathbb{N}$ let $M_i(n) = \mathbf{Inv}_{I_i}(n)$ and let $\boldsymbol{M}$ be the collection of functions $\{\, M_i \,:\, i \in \mathbb{N} \,\}$. It is clear that $\boldsymbol{M}$ is countable, and $\boldsymbol{M}$ is pointwise negligible because it is a subset of the pointwise negligible collection $\boldsymbol{F}$. To complete the proof, we show that $\boldsymbol{M}$ majors $\boldsymbol{F}$. Consider any inverter $I \in \mathcal{I}$ and let p be a polynomial bounding its size. Let i be such that $p = p_i$. Then for each $n \in \mathbb{N}$ we have $\mathbf{Inv}_I(n) \leq \mathbf{Inv}_{I_i}(n)$. Thus $\mathbf{Inv}_I \leq_{\mathrm{ev}} \mathbf{Inv}_{I_i} = M_i$. ▮

4.2 Application to negligible error arguments

An argument, also called a computationally sound proof [3, 4], is a two-party protocol in which soundness is only required to hold with respect to polynomial-time cheating provers. (As usual one can consider either uniform or non-uniform cheating provers.) A couple of definitions of negligible error arguments have appeared in the literature. We show how they correspond to the two different views of negligibility of collections of functions and then show they are equivalent.

Let us begin with the definitions. We consider a two-party protocol in which a prover attempts to convince a probabilistic, polynomial time verifier V that their common input belongs to some underlying language L. An adversary in this context is called a cheating prover. Associated to any cheating prover P (uniform or non-uniform) is its error-probability function $\mathbf{Err}_P \colon \mathbb{N} \to \mathbb{R}$, defined for all $n \in \mathbb{N}$ by

$$\mathbf{Err}_P(n) \;=\; \max \{\, \mathbf{Acc}_P(x) \;:\; x \in \{0,1\}^n \text{ and } x \notin L \,\} \,.$$

Here $\mathbf{Acc}_P(x)$ denotes the probability, taken over the coins of V and P, that V accepts in a conversation with P on common input x. We adopt the convention $\mathbf{Err}_P(n) = 0$ when the set in the maximization above is empty. We let $\mathcal{P}$ denote the set of all cheating provers. (In the uniform case, this is the countable set of all PPT algorithms, and in the non-uniform case the uncountable set of all sequences of circuits that have polynomial size.)

In an interactive proof setting [6], where cheating provers are not computationally bounded, we would say the error probability is $\delta\colon \mathbb{N} \to \mathbb{R}$ if $\mathbf{Err}_P(n) \leq \delta(n)$ for all P and all $n \in \mathbb{N}$. One might try to define the error-probability similarly in the case of arguments by simply replacing "for all P" by "for all $P \in \mathcal{P}$". This however does not yield a suitable notion of error probability for an argument.[2] Taking this into account, two different definitions of computational soundness have been proposed in the literature. Computational soundness as defined below is from [5, 9] while uniform computational soundness is from [2]:

Definition 4.3 Let $V, L, \mathcal{P}$ be as above. We say that V is *computationally sound over L* if for every cheating prover $P \in \mathcal{P}$ the function $\mathbf{Err}_P$ is negligible. We say that V is *uniformly computationally sound over L* if there is a negligible function δ (called the *error-probability* of V) such that $\mathbf{Err}_P \leq_{\text{ev}} \delta$ for every $P \in \mathcal{P}$. ▌

In the notion of computational soundness, there is no "error-probability" associated to V. Instead, different cheating provers might have different error probabilities, as long as they are all negligible. Uniform computational soundness, in contrast, asks that there be an identifiable function δ, depending only on V, that is called the error-probability, and in that sense is closer in spirit to the definition in the case of interactive proofs. However, it turns out the two notions are equivalent.

Theorem 4.4 Let V, L be as above. Then V is computationally sound over L if and only if V is uniformly computationally sound over L. ▌

For the proof, we let $\boldsymbol{F} = \{\, \mathbf{Err}_P \ :\ P \in \mathcal{P} \,\}$ denote the collection of error-probability functions associated to the set of cheating provers under consideration. This collection is countable in the uniform case and uncountable in the non-uniform case. As before, the key observation is that V is computationally sound over L if and only if $\boldsymbol{F}$ is pointwise negligible, and V is uniformly computationally sound over L if and only if $\boldsymbol{F}$ is uniformly negligible. To complete the proof it suffices to show that $\boldsymbol{F}$ is pointwise negligible if and only if it is uniformly negligible. This can be done as in the proof of Theorem 4.2, directly by Theorem 3.2 for the uniform case, and via Theorem 3.7 for the non-uniform case. We omit the details to avoid repetition.

4.3 Application to proofs of knowledge

An NP-*relation* is a function $\rho\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ computable in time polynomial in the length of its first argument. For any $x \in \{0,1\}^*$ we let $\rho(x) = \{\, w \in \{0,1\}^* \ :\ \rho(x,w) = 1 \,\}$ denote the witness set of x. We let $\mathsf{Lang}(\rho) = \{\, x \in \{0,1\}^* \ :\ \rho(x) \neq \emptyset \,\}$ denote the language defined by ρ. Note that a language L is in NP iff there exists an NP-relation ρ such that $L = \mathsf{Lang}(\rho)$.

Let ρ be an NP-relation and let $L = \mathsf{Lang}(\rho)$. Let V be a probabilistic, polynomial-time verifier defining a two-party protocol. An adversary in this context is called a cheating prover and $\mathcal{P}$ denotes the set of all cheating provers. (As above, $\mathcal{P}$ is countable in the uniform case and uncountable in the non-uniform case.) As above, let $\mathbf{Acc}_P(x)$ denote the probability, taken over the coins of V and P, that V accepts in a conversation with prover P on common input x. Below P_x denotes prover P with common input fixed to x. The two notions in the definition below are both from [1].

[2] To see why, consider the following protocol for membership in the language $L = \emptyset$. On common input x the verifier picks a pair of random primes of length $n = |x|$ and multiplies them to get a modulus N which it sends to the prover. It accepts if the prover returns the factorization of N. Intuitively (and formally as per Definition 4.3), this protocol is computationally sound if factoring is hard. But for any negligible $\delta\colon \mathbb{N} \to \mathbb{R}$, there exists a polynomial-time P and an $n \in \mathbb{N}$ such that $\mathbf{Err}_P(n) > \delta(n)$.

Definition 4.5 Let $\rho, V, L, \mathcal{P}$ be as above. We say that V defines a *computationally sound proof of knowledge for* ρ if there is an expected polynomial time oracle algorithm E (called the *extractor*) such that for each cheating prover $P \in \mathcal{P}$ there is a negligible function κ_P such that

$$\Pr[\, E^{P_x}(x) \in \rho(x) \,] \;\geq\; \mathbf{Acc}_P(x) - \kappa_P(|x|)$$

for all $x \in \mathsf{Lang}(\rho)$. We say that V defines a *uniformly computationally sound proof of knowledge over* ρ if there is an expected polynomial-time oracle algorithm E (the extractor) and a negligible function κ (called the *knowledge error*) such that for every cheating prover $P \in \mathcal{P}$ there is a constant n_P such that

$$\Pr[\, E^{P_x}(x) \in \rho(x) \,] \;\geq\; \mathbf{Acc}_P(x) - \kappa(|x|)$$

for all $x \in \mathsf{Lang}(\rho)$ that have length at least n_P. ▮

The difference is that in a uniformly computationally sound proof of knowledge, there is an identifiable function called the knowledge-error, analogous to an error-probability in proofs of membership, while in a computationally sound proof of knowledge, there is no single such function, but instead the function depends on the cheating prover. However, the two notions are equivalent.

Theorem 4.6 Let ρ, V, L be as above. Then V defines a computationally sound proof of knowledge over ρ if and only if V defines a uniformly computationally sound proof of knowledge over ρ. ▮

In this case it may be a little less evident than before how the issue corresponds to negligibility of some collection of functions. For the proof, we first claim something stronger than the theorem statement, namely that not only are the notions equivalent, but the extractor is the same in both cases. So view the extractor E as now fixed. For each prover $P \in \mathcal{P}$ define the function

$$F_P(n) \;=\; \max\{\, \mathbf{Acc}_P(x) - \Pr[\, E^{P_x}(x) \in \rho(x) \,] \;:\; x \in \{0,1\}^n \text{ and } x \in L \,\}\,.$$

We adopt the convention $F_P(n) = 0$ when the set in the maximization above is empty. We consider the collection of functions $\boldsymbol{F} = \{\, F_P \;:\; P \in \mathcal{P} \,\}$. Now we observe that V defines a computationally sound proof of knowledge over ρ if and only $\boldsymbol{F}$ is pointwise negligible, and V defines a uniformly computationally sound proof of knowledge over ρ if and only if $\boldsymbol{F}$ is uniformly negligible. We then show that $\boldsymbol{F}$ is uniformly negligible if and only if it is pointwise negligible as before.

Acknowledgments

Thanks to Oded Goldreich for pointing out that Theorem 3.2 cannot be directly applied in the case of non-uniform adversaries because there are uncountably many of these, and suggesting how to overcome this difficulty. His solution is used in the proofs of the non-uniform cases. Thanks to Phillip Rogaway and Oded Goldreich for (independently) pointing to the connection with the notion of security of [7, 8]. Thanks to the anonymous referees of the Journal of Cryptology for their comments on previous versions of this manuscript.

References

[1] M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. *Advances in Cryptology – CRYPTO '92*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992, pp. 390–420.

[2] M. Bellare, M. Jakobsson and M. Yung. Round-optimal zero-knowledge arguments based on any one-way function. *Advances in Cryptology – EUROCRYPT '97*, Lecture Notes in Computer Science Vol. 1233, W. Fumy ed., Springer-Verlag, 1997, pp. 280–305.

[3] G. Brassard and C. Crépeau. Non-transitive Transfer of Confidence: A perfect Zero-knowledge Interactive protocol for SAT and Beyond. *Proceedings of the 27th Symposium on Foundations of Computer Science*, IEEE, 1986, pp. 188–195.

[4] G. Brassard, D. Chaum and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *J. Computer and System Sciences*, Vol. 37, 1988, pp. 156–189.

[5] O. Goldreich. Foundations of cryptography: Basic tools. Cambridge University Press, 2001.

[6] S. Goldwasser, S. Micali and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. on Computing*, Vol. 18, No. 1, February 1989, pp. 186–208.

[7] L. Levin. One-way functions and pseudorandom generators. *Combinatorica*, Vol. 7, No. 4, 1987, pp. 357–363.

[8] M. Luby. Pseudorandomness and cryptographic applications. Princeton Computer Science Notes, Princeton University Press, 1996.

[9] M. Naor, R. Ostrovsky, R. Venkatasan, M. Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. *Journal of Cryptology*, Vol. 11, No. 2, 1998, pp. 87–108.