One-way Functions are Essential for Complexity Based Cryptography

(Extended Abstract)

Russell Impagliazzo*
Department of Mathematics
U.C. Berkeley
Russelli@ernie.berkeley.edu

Michael Luby†
International Computer Science Institute
Berkeley, California
Luby@icsi.berkeley.edu

1 Introduction

In much of modern cryptography, for many cryptographic tasks, the security of protocols that have been designed is based on the intractability of a problem such as factorization of randomly chosen large numbers. To date, the problems assumed to be intractable all have the same form; they are based on a one-way function, i.e. one that is easy to compute but hard to invert. In fact, intuitively, any protocol that is secure needs to be based on some notion of "one-wayness", although perhaps not with respect to the same definition of "one-way" function for each protocol. In this paper, we formalize this intuition by showing that many of the standard cryptographic tasks are equivalent to the usual definition of a one-way function. In particular, we show that for some of the standard cryptographic tasks any secure protocol for the task can be converted into a one-way function in the usual sense, and thus the security of any proposed protocol for these tasks is implicitly based on a function being "one-way". Thus, the usual definition of a one-way function is robust; any one-way function with respect to another definition from which a secure cryptographic protocol can be based can be used to construct a one-way function in the usual sense.

Because there are too many cryptographic tasks in existence to discuss in a single paper, we choose to cover a spectrum of the more central cryptographic applications: private-key encryption, identification/authentication, bit commitment and coin-flipping by telephone. The proof techniques presented here can be easily adopted to prove analogous results for a variety of cryptographic tasks not discussed in this paper.

Our results allows us for all of the tasks to design a protocol that is in a "normal form"; i.e., for each task there is a universal format so that any given secure protocol for the task can be used to design a secure protocol in the universal format. For example, we show that any secure protocol for private-key encryption, (even one with many communication rounds that is probabalistic and has some chance of error in decryption) yields a pseudo-random generator. Combining this with [Goldreich, Goldwasser, Micali 84] and [Luby, Rackoff 86], we have a way of converting any private-key encryption scheme into a "pseudo-random block cipher" (a provably secure DES-like function). This resolves an open problem stated in [Rackoff 88].

Some of the results presented here were announced in [Impagliazzo, Luby 89]. Many of the proofs presented here rely on [Impagliazzo, Levin, Luby 89] and [Hastad 89], which strengthens the results in [Impagliazzo, Levin Luby 89] to the uniform model.

Notation : Let x and y be bit strings. Then, $|x|$ is the length of x, $x \circ y$ is the concatenation of x and y, x_i is the i^{th} bit of x and $x_{\leftarrow i}$ is the first i bits of x. If α is a number, then $|\alpha|$ is the absolute value of α.

2 Background

For brevity in this extended abstract, the class of all polynomial bounded functions is the only *resource class* considered with respect to adversaries. Thus, a *feasible adversary* is an algorithm that runs in polynomial time in the uniform model, and is a family of polynomial size circuits in the non-uniform model. All the results in this paper hold for both the uniform and non-uniform model of security, and each definition, lemma, proposition and theorem has two

*Research partially supported by NSF grant CCR 88-13632
†On leave of absence from the University of Toronto, research partially supported by NSERC of Canada operating grant A8092

CH2806-8/89/0000/0230/$01.00 © 1989 IEEE

versions, one in each model. A function $p : N \rightarrow N$ is *negligible* if for all constants $c > 0$ and for almost all n, $p(n) \leq 1/n^c$.

2.1 One-way functions

Intuitively, a function f is *one-way* if it is easy to compute but hard to invert, i.e. given x the value of $f(x)$ can be computed in polynomial-time but every feasible algorithm that receives as input $f(x)$ (when x is a randomly chosen string of length n) can output a y such that $f(y) = f(x)$ with only negligible probability.

Notation (functions and probability ensembles) : A *function* f with input length n and output length $l(n)$ specifies for each $n \in N$ a function $f_n : \{0,1\}^n \rightarrow \{0,1\}^{l(n)}$ that is computable in polynomial time. For simplicity, we write $f(x)$ in place of $f_n(x)$. D is a *probability ensemble* if $D = \{D_n : n \in N\}$, where D_n is a probability distribution on $\{0,1\}^n$. We use the notation $x \in_D \{0,1\}^n$ to mean that x is randomly chosen from $\{0,1\}^n$ according to D_n. The Shannon entropy [Shannon 48] of D_n is given by $Ent(D_n) = -\sum_{x \in \{0,1\}^n} D[\{x\}] \cdot \log(D[\{x\}])$. The *uniform probability ensemble* U assigns to each positive integer n the uniform probability distribution U_n on bit strings of length n. When the context is clear, we simply say that x is chosen randomly when $x \in_U \{0,1\}^n$. Let D_n and E_n be probability distributions on $\{0,1\}^n$. D_n and E_n are *statistically indistinguishable within* δ if for every $X \subseteq \{0,1\}^n$, $|\Pr[x \in X] - \Pr[y \in X]| \leq \delta$ when $x \in_D \{0,1\}^n$ and $y \in_E \{0,1\}^n$.

Definition (one-way function) : We say that f is *somewhat one-way* if, for some constant $c > 0$, for every feasible algorithm M, the inverting probability $\Pr[f(x) = f(M(f(x)))]$ when $x \in_U \{0,1\}^n$ is at most $1 - 1/n^c$. We say that f is *one-way* if, for every feasible algorithm M, the inverting probability $\Pr[f(x) = f(M(f(x)))]$ when $x \in_U \{0,1\}^n$ is negligible.

The following is implicitly used in [Yao 82].

Proposition (somewhat one-way $\rightarrow$ one-way) : From any f that is somewhat one-way it is possible to construct a g that is one-way.

2.2 Pseudo-random generators

The following definition is from [Blum, Micali 82], [Yao 82].

Definition (prg) : f is a *generator* if $l(n) > n$ for all $n \in N$. The *distinguishing probability* $p(n)$ of algorithm M for f is $|\Pr[M(f(x)) = 1] - \Pr[M(y) = 1]|$ when $x \in_U \{0,1\}^n$ and $y \in_U \{0,1\}^{l(n)}$. We say that f is *pseudo-random* if every feasible algorithm has negligible distinguishing probability for f.

The following proposition for the non-uniform model of security is from [Impagliazzo, Levin, Luby 89]. The strengthened version of the proposition for the uniform model of security is from [Hastad 89].

Proposition (one-way $\rightarrow$ prg) : If there is a one-way function then there is a pseudo-random generator.

The following definition appears in [Goldwasser, Micali 82], [Yao 82] and [Goldwasser, Micali, Rackoff 85].

Definition (comp. indistinguishable) : Let D and E be probability ensembles. The *distinguishing probability function* $p(n)$ of algorithm M for D and E is $|\Pr[M(x) = 1] - \Pr[M(x') = 1]|$ when $x \in_D \{0,1\}^n$ and $x' \in_E \{0,1\}^n$. D is *computationally indistinguishable* from E if every feasible algorithm M has negligible distinguishing probability.

The following definition and proposition is from [Impagliazzo, Levin, Luby 89].

Definition (false entropy) : We say function f has *false entropy* if there is a constant $c > 0$ such that the probability ensemble D defined by $D_n = f(x)$ where $x \in_U \{0,1\}^n$ is computationally indistinguishable from a polynomial time sampleable probability ensemble E, where $Ent(E_n) \geq Ent(D_n) + 1/n^c$.

Proposition (false entropy $\rightarrow$ prg) : If there is a polynomial time computable function f that has false entropy then there is a pseudo-random generator.

3 Distributionally One-Way

One of the main stepping stones in our proofs is a distributionally one-way function (defined below). The notion of a distributionally one-way function was inspired by [Rudich 88] and was defined under the name "informationally one-way functions" in [Impagliazzo, Luby 89]. Before giving the formal definition, we motivate why distributionally one-way is a natural generalization of the concept of one-way. Consider a one-way function f which is not computed directly but instead via a probabilistic polynomial time algorithm A that can err, i.e. for most r, $A(x,r) = f(x)$. For example, to compute $f(x)$, A might use a BPP subroutine; where if the subroutine makes an error, the value A outputs is arbitrary. Let (x,r) be a random pair and let $y = f(x)$. Even though no adversary can find, given y, an x' such that $f(x') = y$, there might be an adversary that can find a pair (x',r') such that $A(x',r')$ is mistakenly equal to y (instead of to $f(x')$). Thus, the new function A might not be one-way in the standard sense. Although, technically,

this adversary does find an inverse of y with respect to A, the inverses found all have a special form, easily distinguishable from a random inverse. The following definition of distributionally one-way insists that an adversary must produce random preimages of y with respect to A approximately uniformly. Thus, even though the A described above is not one-way, it is distributionally one-way.

Definition (distributionally one-way) : We say that f is *distributionally one-way* if, for some constant $c > 0$, for every feasible (probabilistic) algorithm M, the distribution defined by $x \circ f(x)$ and the distribution defined by $M(f(x)) \circ f(x)$ are statistically distinguishable by at least n^{-c} when $x \in_U \{0,1\}^n$.

If f is a distributionally one-way function then it is computationally infeasible to randomly generate preimages of $f(x)$. With this definition, we fix a parameter c of distinguishability: an algorithm that randomly generates preimages of $f(x)$ in a manner statistically close but still n^{-c} distinguishable from the uniform distribution is considered to be unsuccessful. This is a weak form of a one-way function, but the following lemma shows that such a one-way function can be used to construct a one-way function in the usual sense.

Lemma 1 : If there is a distributionally one-way function then there is a one-way function.

Proof Sketch : Let f be a distributionally one-way function with associated constant c. Let $H_{n,m}$ be a family of universal hash functions from n bit strings to m bit strings [Carter, Wegman]. Let $g(h \circ i \circ x)$ be $h \circ i \circ f(x) \circ (h(x)_{\leftarrow i+2c\log n})$ (where $i \in \{1, \cdots, n\}$ and $h \in H_{n,n+2c\log n}$). In the full paper, we prove that g is a somewhat one-way function. The proof of this claim uses techniques and lemmas from [Impagliazzo, Levin, Luby 89]. By Proposition (somewhat one-way $\rightarrow$ one-way), this suffices to construct a one-way function. □

In our results, we prove that a secure protocol yields one of the following: (1) a distributionally one-way function; (2) a one-way function; (3) a false entropy generator; (4) a pseudo-random generator. By the results of [Impagliazzo, Levin, Luby 89] and [Hastad 89], and by Lemma 1, all of these are equivalent.

4 Applications

Definition (protocol) : A *protocol* consists of a message length function, $m(n)$, that is polynomial in n, a number of rounds function, $r(n)$, and two polynomial time deterministic Turing machines A and B, known as the participants "Alice" and "Bob". On input x for Alice and y for Bob, the conversation between Alice and Bob is defined by $C_{AB}(x,y) = m_1, m_2, \ldots, m_{r(n)}$, where $m_i = A(x, m_1, \ldots, m_{i-1})$ if i is even, and $m_i = B(y, m_1, \ldots, m_{i-1})$ if i is odd. Bob has a private output $OUT_B^{AB}(x,y)$ which is defined to be the output of B when the input to A is x and the input to B is y. Note that $OUT_B^{AB}(x,y)$ is completely determined by y and $C_{AB}(x,y)$. $OUT_A^{AB}(x,y)$ is defined similarly to be the output of A for the same inputs. A well known fact about protocols is:

Proposition 2 : If, for a particular protocol, $C_{AB}(x,y) = C_{AB}(x',y')$ then $C_{AB}(x,y) = C_{AB}(x,y') = C_{AB}(x',y) = C_{AB}(x',y')$. In other words, the set of input pairs x, y that yield the same conversation is $(x,y) \in X \times Y$, where X is a set of inputs for Alice and Y is a set of inputs for Bob.

4.1 Identification

Intuitively, an identification protocol allows Alice to convince Bob that she is the same person he was talking to at an earlier time.

Definition (identification protocol) : This consists of two protocols, P^1, called the "introduction" protocol, and P^2, called the "recognition" protocol. In the introduction protocol, Alice (A^1) has random input x and Bob (B^1) has random input y, and this produces the conversation $C_{A^1B^1}(x,y)$. In the recognition protocol, the input for Alice (A^2) is x, $C^1_{A^1B^1}(x,y)$ and x', where x' is a new randomly chosen input. Similarly, the input for Bob (B^2) is y, $C_{A^1B^1}(x,y)$ and y'. At the conclusion of the recognition protocol, Bob either outputs "ACCEPT" or "REJECT". A *secure identification protocol* has the following two properties:

- The probability that

  $$OUT_{B^2}^{A^2B^2}((x, x', C_{A^1B^1}(x,y)), (y, y', C_{A^1B^1}(x,y)))$$

  is "ACCEPT" is at least .9, i.e. Bob should recognize Alice with high probability if Alice participated in the original introduction protocol.

- For all but finitely many lengths n, the probability that

  $$OUT_{B^2}^{LB^2}((0, x', C_{A^1B^1}(x,y)), (y, y', C_{A^1B^1}(x,y)))$$

  is "ACCEPT" is at most .1 for any feasible algorithm L that has access to the conversation from the introduction protocol but does not have the value of x that Alice has. (The random input x' to L is allowed to be of any length that L requires, i.e. x' may be much longer for L than it

is for A). Intuitively, any listener L who heard the introduction conversation between Alice and Bob and who tries to impersonate Alice in the recognition protocol is caught with high probability by Bob.

Theorem : Any secure identification protocol can be used to construct a distributionally one-way function.
Proof Sketch : Let $f(x \circ y) = C_{A^1B^1}(x,y)$. We prove that if the identification protocol is secure then f is a distributionally one-way function by contradiction. Let $X \times Y$ be the set of pairs of inputs to A^1 and B^1, respectively, such that for all $(a,b) \in X \times Y$, $C_{A^1B^1}(a,b) = C_{A^1B^1}(x,y)$. Let M be the feasible algorithm that on input $f(x \circ y)$ produces random (a,b) such that the distribution is statistically indistinguishable within n^{-c} of the uniform distribution on $X \times Y$. The idea is to use M to help listener L simulate Alice during the recognition protocol without knowing x. L takes as input x' and $C_{A^1B^1}(x,y)$. L simulates M on input $C_{A^1B^1}(x,y)$, and this produces (a,b). During the recognition protocol, L simulates exactly what A would do when $x = a$. By Proposition 2 and because M produces close to the uniform distribution on $X \times Y$, the distribution on conversations from the recognition protocol produced by B interacting with L is statistically indistinguishable within n^{-c} from the distribution on conversations from the recognition protocol produced by B interacting with A. Thus, the difference between the probabilities of Bob's accepting in these two cases is not at least .8, as required for a secure identification protocol. □

Identification, as defined here, is a "least common denominator" of many types of cryptographic tasks. Secure protocols for many other cryptographic tasks can be used to construct a secure identification protocol. One such example is a secret key agreement protocol between two parties, where the parties involved agree on a common secret (with reasonable probability) which an eavesdropper cannot feasibly deduce from the conversation. An identification scheme based on secret sharing can be designed as follows. In the introduction protocol, Alice and Bob agree on a common secret. To recognize Alice, Bob asks Alice to send her this secret. This special case, that a secret key agreement protocol implies a one-way function, was independently proved in [Bellare, Cowen, Goldwasser 89] (using Theorem 4.5 of [Impagliazzo, Luby 89], which is Lemma 1 in this paper).

4.2 Bit Commitment

Intuitively a bit commitment protocol makes Alice "commit" herself to a bit without allowing her to change her mind later in the protocol. Alice does not want Bob to know the value of her bit until a later time, and thus Alice sends encrypted information to Bob about the bit from which Bob is unable to determine its value with probability significantly greater than 1/2. On the other hand, when Alice finally does release the value of her bit by releasing information about the encryption, Bob would like to be sure that Alice can't change her mind about the original value to which she committed herself.

Definition (bit commitment protocol) : This consists of two protocols, P^1, called the "commit" protocol, and P^2, called the "release" protocol. In the commit protocol, Alice (A^1) has random input x and a random bit b, and Bob (B^1) has a random input y. Bob either outputs "ACCEPT" or "REJECT" at the conclusion of this protocol, and if and only if Bob outputs "ACCEPT" the release protocol is executed. In the release protocol, Alice (A^2) has input x, b, a random x', and the conversation from the commit protocol; Bob (B^2) has y, a random y', and the commit conversation. At the end of the release protocol, Bob either outputs "REJECT" or a bit b'. If Alice and Bob obey the protocol, Bob should accept with high probability in the commit protocol, and should output $b' = b$ with high probability in the release protocol. Informally, the protocol is secure if: immediately after the commit protocol, no feasible Bob can predict b with probability more than 1/2 + a negligible amount; and, every feasible Alice has only one choice of bit b' she can make Bob output. Formally,

1. $OUT_{B^1}^{A^1B^1}((x,b),y)$ is "ACCEPT" with high probability, for x, y chosen at random.

2. $$OUT_{B^2}^{A^2B^2}((x,b,C_{A^1B^1}((x,b),y),x'),$$
$$(y,C_{A^1B^1}((x,b),y),y')) = b$$
with high probability, for x, y, x', y', b chosen at random.

3. Let B' be any feasible algorithm (for the commitment protocol), and let L be any feasible algorithm that tries to guess b after the commitment protocol and let A^1 be Alice's (real) algorithm for the commitment protocol. Then, $\Pr[OUT_L(y^*, y, C_{A^1B'}((x,b),y)) = b] \leq 1/2+$ a negligible amount, where y^* is the random tape input for L. (Note: y, the random input to cheating algorithm B', may be longer than the y input to B^1 would be.)

4. Let A'^1 and A'^2 be any feasible algorithms (for the two stages of the protocol, impersonating Alice), and let B^1 and B^2 be Bob's (real) algorithm for the two stages of the protocol. Let x, x', y, y' be the random tapes for A'^1, A'^2, B^1, B^2, respectively (x and x' may be longer than the x and x' used by the real A^1 and A^2). Then the probability that:

   (a) $OUT_{B^1}^{A'^1 B^1}((x,b),y) =$ "ACCEPT".

   (b)

$$OUT_{B^2}^{A'^2 B^2}((x,b,C_{A'^1 B^1}((x,b),y),x'),$$
$$(y,C_{A'^1 B^1}((x,b),y),y')) \neq b'$$

   for some fixed value $b' \in \{0,1\}$.

   is negligible. (In other words, at the end of the commit protocol at most one value in $\{0,1\}$ is a possibility to be output by Bob at the end of the release protocol.)

The above definition of bit commitment includes as special cases bit commitment versus a strong receiver [Brassard, Crepeau 86], [Chaum 86] and versus a strong committer [Goldreich, Micali, Wigderson 86]. If we remove the word "feasible" in the third clause above, we call the protocol "secure vs. a strong receiver", and if we remove the word "feasible" from the fourth clause, we call the protocol "secure vs a strong committer." Combining the results here with [Impagliazzo, Levin, Luby 89], [Hastad 89] and [Naor 89], we get that the existence of any bit commitment protocol implies a bit commitment protocol versus a strong committer. [Goldwasser, Micali, Rackoff 85] define the notion of a zero-knowledge proof system, and [Goldreich, Micali, Wigderson 86] show how any bit commitment protocol that is secure versus a strong committer can be used to construct a zero-knowledge proof system for any problem in *NP*. Putting this all together, any bit commitment scheme can be used to construct a zero-knowledge proof system for any problem in *NP*. The analogous statement for a strong receiver, which would imply that any bit commitment could be used for minimum disclosure proofs [Brassard, Crepeau 86], [Chaum 86], is an open question.

Theorem : Any secure bit commitment protocol can be used to construct a distributionally one-way function.

Proof Sketch : In the full paper, we show that a bit-commitment scheme can be used for identification. In the introduction protocol, Alice commits to several randomly chosen bits. In the recognition protocol, she releases these bits. □

4.3 Private Key Encryption

Private key encryption is a system by which two people who have previously met and privately selected a random n bit key k, can send messages of length longer than n securely over an insecure channel. In other words, a computationally limited listener who does not know k should not be able to deduce any information about which message (of length at least $n+1$) was sent from overhearing the conversation on the insecure channel. This notion of an interactive probabilistic private key encryption system was suggested in [Rackoff 88]. More formally:

Definition (weak private key encryption protocol) : Let $l(n) > n$ be polynomial in n. The idea is that Alice and Bob share a secret key k of length n, and Alice wants to send to Bob a message m of length $l(n)$ securely, where m is randomly chosen. i.e. so that no feasible listener L (who does not know k) can determine anything about m. Alice and Bob are probabilistic polynomial time Turing machines. A has input the key k, the message m and a random tape x. B has input the key k and a random tape y. A and B have a conversation such that at the end of the conversation with high probability B knows m. (The fact that m is assumed to be randomly chosen is the reason we call this a weak protocol; usually m is assumed to be chosen from an arbitrary probability distribution. Our result that a weak protocol implies the existence of a pseudo-random generator immediately implies that a weak protocol can be used to build a strong private key encryption protocol.) More formally, a *secure weak private key encryption protocol* satisfies the following properties:

- With probability at least .9 Bob outputs the message m at the end of the protocol, i.e. $\Pr[OUT_B^{AB}((k,m,x),(k,y)) = m] \geq .9$.
- No feasible algorithm L can learn anything about m from the conversation between A and B, i.e. the distribution defined by $C_{AB}((k,m,x),(k,y)) \circ m$ is computationally indistinguishable from the distribution defined by $C_{AB}((k,m,x),(k,y)) \circ r$, where k, x, y, m, r are all uniformly and randomly chosen and $|r| = |m|$.

Theorem : Any secure weak private key encryption protocol can be used to construct a pseudo-random generator. **Proof Sketch :** Let $f(x \circ y \circ k \circ m) = C_{AB}((k,m,x),(k,y)) \circ m$. We claim that the entropy of the distribution $C_{AB}((k,m,x),(k,y)) \circ m$ is significantly smaller than that of the distribution $C_{AB}((k,m,x),(k,y)) \circ R$. Thus, f has false entropy. (This is true in both the uniform and non-uniform

model of security, since both distributions are polynomial samplable and they are computationally indistinguishable by assumption.) The theorem follows from Proposition (false entropy → prg).

The reason for the difference in entropies is that the $l(n)$ bit string m is determined with probability at least .9 by the conversation and the n bit string k, i.e. there is an algorithm (not necessarily efficient) that given the conversation and k produces m with probability at least .9. The algorithm randomly generates a y' so that Bob's behavior on input (k, y') is consistent with the conversation. Note that, for a fixed conversation and key k, the distribution on (m, y') is the same as the distribution on (m, y), where y is the original random tape for Bob. Therefore, the probability that the algorithm outputs m is the same as the corresponding probability for B. Thus, the expected conditional entropy on m given the conversation is at most $n+.1(l(n)-n) \leq l(n)-.9$ (because $l(n) \geq n+1$), whereas that of R is $l(n)$. From this it follows that f has false entropy at least .9. A more complete argument will be included in the final version of this paper. □

4.4 Coin Flipping on the Phone

Coin flipping by telephone [Blum 82] is a method for two participants who do not trust each other to choose a random bit. We will give a formal definition and prove the following in the full paper.

Theorem : If there is a secure protocol for coin-flipping by telephone, then there is a distributionally one-way function.

5 Summary

The results in this paper and several other recent papers in theoretical cryptography as summarized as follows. Constuctions such as one-time pads and secret sharing [Shamir 79] are possible information theoretically. For convenience, the following summarizes the results from many paper, including those in this paper. The existence of the following are equivalent to the existence of a one-way function: informationally one-way function; function with false entropy; pseudo-random generators; pseudo-random function generator; pseudo-random permutation generator; weak identification protocol; strong identification protocol [Fiat, Shamir 86]; bit commitment; bit commitment versus a strong committer; coin flipping by telephone.

References

[1] Bellare, M., Cowen, L., Goldwasser, S., "Secret Key Exchange", unpublished manuscript, May 1989.

[2] Blum, M., "Coin Flipping by Phone," *IEEE Spring COMPCOM*, Feb. 1982, pp. 133-137.

[3] Brassard, G., Crepeau, C., "Non-Transitive Transfer of Confidence: A Perfect Zero-Knowledge Interactive Protocol for SAT and Beyond," *Proceedings of FOCS 1986.*

[4] Blum, M., and Micali, S., "How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits," *SIAM J. on Computing*, Vol. 13, 1984, pp. 850-864, *FOCS 1982.*

[5] Carter, J., and Wegman, M., "Universal Classes of Hash Functions," *JCSS*, 1979, Vol. 18, pp. 143-154.

[6] Chaum, D., "Demonstrating that a public predicate can be satisfied without revealing any information about how," *CRYPTO 1986.*

[7] Fiat, A., Shamir, A., "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," *Proceedings of CRYPTO 1986.*

[8] Goldreich, O., S. Goldwasser, and S. Micali, "How to Construct Random Functions," *J. of ACM*, Vol. 33, No. 4, 1986, pp. 792-807, *FOCS 1984.*

[9] Goldreich, O., Micali, M. and Wigderson, A., "Proofs that Yield Nothing but Their Validity and a Methodology of Cryptographic Protocol Design," 27^{th} *FOCS*, 1986, pp. 174-187, *Tech. Report TR498*, Comp. Sci. Dept., Technion, submitted to *JACM.*

[10] Goldwasser, S. and Micali, S., "Probabilistic Encryption," *JCSS*, Vol. 28, No. 2, April 1984, pp. 270-299, *STOC 1982.*

[11] Goldwasser, S., Micali, S. and Rackoff, C., "The Knowledge Complexity of Interactive Proof Systems," *SIAM J. on Computing*, Vol. 18, No. 1, 1989, pp. 186-208, *STOC 1985.*

[12] Impagliazzo, R., Levin, L., Luby, M., "Pseudo-random generation from one-way functions," *Proceedings of STOC 1989.*

[13] Impagliazzo, R., Luby, M., "Pseudo-Random Number Generation from any One-way Function," International Computer Science Institute technical report *TR-89-002*, February, 1989.

[14] Luby M., and Rackoff, C., "How to Construct Pseudorandom Permutations From Pseudorandom Functions," *SIAM J. on Computing*, Vol. 17, 1988, pp. 373-386, *STOC 1986*

[15] Naor, M., personal communication, 1988.

[16] Rackoff, C., "A Basic Theory of Public and Private Cryptosystems," *Proceedings of CRYPTO 1988*

[17] Rudich, S., "Limits on the Provable Consequences of One-Way Functions," *Ph.D. Thesis*, Department of Computer Science, U.C. Berkeley, 1988.

[18] Shamir, A., "How to Share a Secret," *CACM*, 1979, 22, pp. 612-613.

[19] Shannon, C., "A Mathematical Theory of Communication," *Bell Systems Technical Journal*, 27, 1948, pp. 379-423 and pp. 623-656.

[20] Yao, A.C., "Theory and Applications of Trapdoor Functions," 23^{rd} *FOCS*, 1982, pp. 80-91.