Draft of an Appendix regarding Corrections and Additions

(revised, second posted version)

Extracts from a working draft for
Volume 2 of Foundations of Cryptography

Oded Goldreich

Department of Computer Science and Applied Mathematics
Weizmann Institute of Science, Rehovot, ISRAEL.

February 4, 2003

Appendix C

Corrections and Additions to Volume 1

There is no 100% guarantee in the world;
whoever wants 100% guarantee should not build a thing.

Eng. Isidor Goldreich (1906–1995)

In this appendix we list a few corrections and additions to the previous chapters of this work (which appeared in [135]).

C.1 Enhanced Trapdoor Permutations

Recall that a collection of trapdoor permutations, as defined in Definition 2.4.5, is a collection of permutations, $\{f_\alpha\}_\alpha$, armed with four probabilistic polynomial-time algorithms, denoted here by I, S, F and B (for *index*, *sample*, *forward* and *backward*), such that the following (syntactic) conditions hold

1. On input 1^n, algorithm I selects a random n-bit long *index* α of a permutation f_α, along with a corresponding trapdoor τ;
2. On input α, algorithm S *samples* the domain of f_α, returning a random element in it;
3. For x in the domain of f_α, given α and x, algorithm F returns $f_\alpha(x)$ (i.e., $F(\alpha, x) = f_\alpha(x)$);
4. For y in the range of f_α if (α, τ) is a possible output of $I(1^n)$ then, given τ and y, algorithm B returns $f_\alpha^{-1}(y)$ (i.e., $B(\tau, y) = f_\alpha^{-1}(y)$).

The hardness condition in Definition 2.4.5 refers to the difficulty of inverting f_α on a random element of its range, when given only the range-element and α. That is, let $I_1(1^n)$ denote the first element in the output of $I(1^n)$ (i.e., the

index), then for every probabilistic polynomial-time algorithm A (resp., every non-uniform family of polynomial-size circuit $A = \{A_n\}_n$), every polynomial p and all sufficiently large n's

$$\mathsf{Pr}[A(I_1(1^n), f_{I_1(1^n)}(S(I_1(1^n))) = S(I_1(1^n))] \ < \ \frac{1}{p(n)} \qquad \text{(C.1)}$$

Namely, A (resp., A_n) fails to invert f_α on $f_\alpha(x)$, where α and x are selected by I and S as above. An equivalent way of writing Eq. (C.1) is

$$\mathsf{Pr}[A(I_1(1^n), S'(I_1(1^n), R_n)) = f_{I_1(1^n)}(S'(I_1(1^n)), R_n)] \ < \ \frac{1}{p(n)} \qquad \text{(C.2)}$$

where S' is the residual two-input (deterministic) algorithm obtained from S when treating the coins of the latter as an auxiliary input, and R_n denote the distribution of the coins of S on n-bit inputs.

Although the above definition suffices for many applications, in some cases we will need an enhanced hardness condition. Specifically, we will require that it is hard to invert f_α on a random input x (in the domain of f_α) even when given the coins used by S in the generation of x. (Note that given these coins (and the index α), the resulting domain element x is easily determined.)

Definition C.1.1 (enhanced trapdoor permutations): *Let $\{f_\alpha : D_\alpha \to D_\alpha\}$ be a collection of trapdoor permutations as in Definition 2.4.5. We say that this collection is* enhanced (and call it an enhanced collection of trapdoor permutations) *if for every probabilistic polynomial-time algorithm A every polynomial p and all sufficiently large n's*

$$\mathsf{Pr}[A(I_1(1^n), R_n) = f_{I_1(1^n)}(S'(I_1(1^n)), R_n)] \ < \ \frac{1}{p(n)} \qquad \text{(C.3)}$$

where S' is as above. The non-uniform version is defined analogously.

We comment that the RSA collection (presented in Section 2.4.3.1 and further discussed in Section 2.4.4.2) is in fact an *enhanced* collection of trapdoor permutations,[1] provided that RSA is hard to invert in the same sense as assumed in Section 2.4.3.1. In contrast, the Rabin Collection (as defined in Section 2.4.3), does not satisfy Definition C.1.1 (because the coins of the sampling algorithm give away a modular square root of the domain element). Still, the Rabin Collection can be easily modify to yield an *enhanced* collection of trapdoor permutations, provided that factoring is hard (in the same sense as assumed in Section 2.4.3). Actually, we present two such possible modifications:

[1] Here and below we assume that sampling Z_N^*, for a composite N, is trivial. However, sampling Z_N^* (or even Z_N) by using a sequence of unbiased coins is not that trivial. The straightforward sampler may take $\ell \stackrel{\text{def}}{=} 2\lfloor \log_2 N \rfloor$ random bits, view them as an integer in $i \in \{0, 1, ..., 2^\ell - 1\}$, and output $i \bmod N$. This yields an almost uniform sample in Z_N. Also note that given an element $e \in Z_N$, one can uniformly sample an $i \in \{0, 1, ..., 2^\ell - 1\}$ such that $i \equiv e \pmod N$. Thus, the actual sampler does not cause trouble with respect to the enhanced hardness requirement.

1. *Modifying the functions.* Rather than squaring modulo the composite N, we consider the function of raising to the power of 4 modulo N. It can be shown that the resulting permuations over the quadratic residues modulo N satisfy Definition C.1.1, provided that factoring is hard. Specifically, given N and a random $r \in Z_N$, ability to extract the 4th root of r^2 mod N (modulo N), yields ability to factor N, where the algorithm is similar to the one used in order to establish the intractability of extracting square roots.

2. *Changing the domains.* Rather than considering the permutation induced (by the modoluar squaring function) on the set Q_n of the quadratic residues modulo N, we consider the permulations induced on the set M_n, where M_n contains all integers is $\{1, ..., N/2\}$ that have Jacobi symbol modulo N that equals 1. Note that, as in case of Q_n, each quadratic residue has a unique square root in M_n (because exactly two square roots have Jacobi symbol that equals 1 and their sum equals N).[2] However, unlike Q_N, membership in M_N can be determined in polynomial-time (when given N without its factorization). Thus, sampling M_N can be done in probabilistic polynomial-time.

   Actually, squaring modulo N is a 1-1 mapping of M_N to Q_N. In order to obtain a permutation over M_N, we modify the function a little such that if the result of modular squaring is bigger than $N/2$ then we use its additive inverse (i.e., rather than outputting $y > N/2$, we output $N - y$).

We comment that the special case of Definition 2.4.5 in which the domain of f_α equals $\{0,1\}^{|\alpha|}$ is a special case of Definition C.1.1 (because, without loss of generality, the sampling algorithm may satisfy $S'(\alpha, r) = r$). Clearly, the above examples can be slightly modified to fit this special case.

Correction to Volume 1: Theorems 4.10.10, 4.10.14 and 4.10.16 (which in turn are based on Remark 4.10.6) refer to the existence of certain non-interactive zero-knowledge proofs. The claimed non-interactive zero-knowledge proof systems can be constructed assuming the existence of an *enhanced* collection of trapdoor permutations. However, in contrast to the original text, it is not known how to derive these proof systems based on the existence of a (regular) collection of trapdoor permutations. See further discussion in Section C.4.2.

C.2 On Variants of Pseudorandom Functions

The focus of Section 3.6 was on a special case of pseudorandom functions, hereafter referred to as the fixed-length variant. For some function $\ell : \mathbb{N} \to \mathbb{N}$ (e.g., $\ell(n) = n$), these functions map $\ell(n)$-bit long strings to $\ell(n)$-bit long strings, where n denotes the lengths of the function's seed. More general definitions

[2] As in case of Q_n, we use the fact that -1 has Jacobi symbol 1.

were presented in Section 3.6.4. In particular, functions mapping strings of *arbitrary length* to $\ell(n)$-bit long strings were considered. Here we refer to the latter as to the variable-length variant.

A natural question regarding these variants is how to directly (or efficiently) transform functions of the fixed-length variant into functions of the variable-length variant.[3] Exercises 30 and 31 in Chapter 3 *implicitly sugggest such a transformation*, and so does Proposition 6.3.7. Because of the interest in the question, we now state the actual result explicitly.

Proposition C.2.1 *Let* $\{f_s : \{0,1\}^{\ell(|s|)} \to \{0,1\}^{\ell(|s|)}\}_s$ *be a* (fixed-length) *pseudorandom function ensemble, and* $\{h_r : \{0,1\}^* \to \{0,1\}^{\ell(|r|)}\}_r$ *be a generalized hashing ensemble with a* $(t, 1/t)$*-collision property,*[4] *for some super-polynomial function* $t : \mathbb{N} \to \mathbb{N}$. *Then* $\{g_{s,r} = f_s \circ h_r\}_{s,r:|s|=|r|}$ *is a* (variable-length) *pseudorandom function ensemble.*

Proof: The proofs of Propositions 6.3.6 and 6.3.7 actually establish Proposition C.2.1. ■

Comment: Alternative constructions of variable-length pseudorandom functions based on fixed-length pseudorandom functions are presented in [32, 29, 21]. In these works, the fixed-length pseudorandom functions is applied to each block of the input, and so the number of applications is linearly-related to the input length (rather than being a single one). On the other hand, these works do not use variable-length hashing. Indeed, these works presuppose that a fixed-length pseudorandom function (rather than a variable-length one) is non-expensive (and, in practice, is available as an off-the-shelf product).

C.3 On Strong Witness Indistinguishability

Unfortunately, we have to withdraw two claims regarding *strong* witness indistinguishable proofs as defined in Definition 4.6.2.[5] Specifically, in general, *strong* witness indistinguishability is not closed under parallel composition (and so Lemma 4.6.7 is wrong). Consequently, in contrary to what is stated in Theorem 4.6.8, we do not know whether there exist constant-round public-coin proofs with negligible error that are *strong* witness indistinguishable for languages out of $\mathcal{BPP}$. Before discussing the reasons for withdrawing these claims and the consequences of doing so, we stress that the flaws pointed out here only refer to *strong* witness indistinguishability and not to (regular) witness indistinguishability. That is, as stated in Lemma 4.6.6, (regular) witness indistinguishability

[3] An indirect construction may use the fixed-length variant in order to obtain a one-way function, and then construct the variable-length variant using this one-way function.

[4] Recall that the $(t, 1/t)$-collision property means that every $n \in \mathbb{N}$ and every $x \neq y$ such that $|x|, |y| \leq t(n)$, the probability that $h_r(x) = h_r(y)$ when r is uniformly selected in $\{0,1\}^n$ is at most $1/t(n)$.

[5] We comment that the notion of *strong* witness indistinguishability was introduced by the author at a late stage of writing [135].

is closed under parallel composition and thus the part of Theorem 4.6.8 that refers to regular witness indistinguishability is valid (i.e., providing constant-round public-coin proofs with negligible error that are witness indistinguishable for $\mathcal{NP}$).

C.3.1 On parallel composition

A counter-example to Lemma 4.6.7 can be derived by using the protocol presented at the end of Section 4.5.4.1 (and assuming the existence of one-way functions). In contrary to what is claimed in Section 4.6.2, the proof of Lemma 4.6.6 (i.e., parallel composition of regular witness indistinguishability) cannot be extended to the case of strong witness indistinguishability, because (in general) the simulating machine V^* may not be given the NP-witnesses required for emulation of the other copies.[6] For example, parallel composition may refer to parallel executions on the same common input while the prover is given the same NP-witnesses. This point is further clarified below.

Parallel composition of strong witness indistinguishable proofs does hold when restricted to input sequences that are identical independent copies of one input distribution. More generally, we can prove the following fact.

Lemma C.3.1 (Parallel Composition for Strong Witness Indistinguishability, Revisited): *Let $L \in \mathcal{NP}$, R_L, (P,V), Q, R_L^Q and P_Q be as in Lemma 4.6.6, and suppose that (P,V) is strong witness indistinguishable. Then for every probabilistic polynomial-time machine V_Q^* and every two probability ensembles $\{(\overline{X}_n^1, \overline{Y}_n^1, \overline{Z}_n^1)\}_{n\in\mathbb{N}}$ and $\{(\overline{X}_n^2, \overline{Y}_n^2, \overline{Z}_n^2)\}_{n\in\mathbb{N}}$ such that $\overline{X}_n^j$ (resp., $\overline{Y}_n^j$ and $\overline{Z}_n^j$) is a sequence of $Q(n)$ independently distributed variables the following holds*

> *if $\{(\overline{X}_n^1, \overline{Z}_n^1)\}_{n\in\mathbb{N}}$ and $\{(\overline{X}_n^2, \overline{Z}_n^2)\}_{n\in\mathbb{N}}$ are computationally indistinguishable then so are $\{\langle P_Q(\overline{Y}_n^1), V_Q^*(\overline{Z}_n^1)\rangle(\overline{X}_n^1)\}_{n\in\mathbb{N}}$ and $\{\langle P_Q(\overline{Y}_n^2), V_Q^*(\overline{Z}_n^2)\rangle(\overline{X}_n^2)\}_{n\in\mathbb{N}}$.*

We stress that the components of $\overline{Y}_n^j$ (resp., $\overline{Z}_n^j$) may depend on the corresponding components of $\overline{X}_n^j$, but they are independent of the other components of $\overline{Y}_n^j$ and $\overline{Z}_n^j$ (as well as of $\overline{X}_n^j$). Lemma C.3.1 is proved by extending the proof of Lemma 4.6.6. Specifically, we consider hybrids as in the original proof, and construct a verifier V^* that interacts with P on the i^{th} copy while emulating all the other copies. Towards this emulation, we provide V^* with the corresponding $Q(n)-1$ components of both $\overline{Y}_n^j$'s (as well as of both $\overline{X}_n^j$'s and $\overline{Z}_n^j$'s). Fixing the best possible choice for these $Q(n)-1$ components, we derive a verifier that interacts with P and contradicts the hypothesis that (P,V) is strong witness indistinguishable. The key point is that revealing (or fixing) the other $Q(n)-1$

[6] The point is that indistinguishability of the transcripts of executions on different common inputs (as required in strong witness indistinguishability) may not hold when the verifier V^* is also given both corresponding NP-witnesses (i.e., the Y_n^i's of Definition 4.6.2). In contrast, indistinguishability of the transcripts of executions on the same common input (as required in regular witness indistinguishability) does hold also when the verifier is given both NP-witnesses for this input.

components of both $\overline{Y}^j_n$'s does not allow to distinguish the i^{th} component of $\overline{X}^1_n$ and $\overline{Z}^1_n$ from the i^{th} component of $\overline{X}^2_n$ and $\overline{Z}^2_n$.

C.3.2 On Theorem 4.6.8 and an afterthought

Unfortunately, Theorem 4.6.8 is proved by parallel composition that refers to the same common input (and the same NP-witness). Thus, Lemma C.3.1 is not applicable, and consequently we do not know whether the part of Theorem 4.6.8 that refers to *strong* witness indistinguishable proofs is valid. This is indeed an interesting open problem.

We comment that one can reduce the construction of constant-round (public-coin) *strong* witness indistinguishable proofs with negligible error for $\mathcal{NP}$ to the construction of such proofs for the special case in which the X^j_n's are identically distributed (and the Z^j_n's are only computationally indistinguishable). Consider, for example, the following protocol:

1. The prover sends a commitment to the value 0.
2. Using a (regular) witness indistinguishable proof (as provided by Theorem 4.6.8), the prover proves that either the common input is in the langauge or the string sent at Step 1 is a commitment to 1.

   Let us denote by T^j_n the transcript of the execution of this step, when the common input is X^j_n (and the parties use auxiliary inputs Y^j_n and Z^j_n, respectively). It can be proven that the T^j_n's are computationally indistinguishable (by considering what happens if at Step 1 the prover sends a commitment to 1).
3. Using a *strong* witness indistinguishable proof (which is indeed the missing component or the subprotocol to which the current protocol is reduced), the prover proves that the string sent at Step 1 is a commitment to 0.

   Note that it suffices to show that the verifier cannot distinguish the possible transcript distributions of the current step, where both transcript distributions refer to the same common input (i.e., the commitment), the same prover's auxiliary input (i.e., the decommitment information), but to the verifier's auxiliary inputs T^1_n and T^2_n, which are different (but indistinguishable).

The foregoing reduction demonstrates that the notion of strong witness indistinguishability actually refers to issues that are fundamentally different from witness indistinguishability. Specifically, the issue is whether the interaction with the prover helps to distinguish some auxiliary information (which is indistinguishable without such an interaction).

C.3.3 Consequences

In view of the fact that we do not have constant-round public-coin strong witness indistinguishable proofs with negligible error for $\mathcal{NP}$, we suggest to replace the

use of such proofs by some cumbersome patches. A typicaly example is the construction of non-oblivious commitment schemes (i.e., Theorem 4.9.4).

Non-oblivious commitment schemes. We begin the discussion by noting that the specific formulation as appearing in Definition 4.9.3 is wrong. One should partition the commit phase into two sub-phases such that the second sub-phase is a proof-of-knowledge of the input and coins used by the sender at the first sub-phase, which in turn should constitute a commitment scheme by itself. That is, the view in the relation displayed in Definition 4.9.3 should be the view of the first sub-phase (rather than the view of the entire commit phase). In fact, for the current implementation we need a relaxed definition in which one only proves knowledge of the input (but not of the coins) used by the sender at the first sub-phase. We stress that the input value proved to be known must be such that it is impossible for the sender to later decommit to a different value. Indeed, in this relaxed form, we do not require that a later decommitment is at all possible; we only require that if it takes place than the outcome matches the above value. Note that this relaxed form suffices for the proof presented in Section 4.9.2.2.

Next, we modify the construction used in the proof of Theorem 4.9.4 as follows. First, rather than sending one ordinary commitment to the input, we send many such (independent) commitments. Secondly, rather than using a (constant-round) proof-of-knowledge with negligible error, we use one that has constant error. The point is that such a (constant-round) proof-of-knowledge that is zero-knowledge (and hence strong witness indistinguishable) is known. We invoke this proof systems many times, in parallel, where each invocation is applied to a different commitment. Thus, we can apply Lemma C.3.1 and conclude that these executions are strong witness indistinguishable (where the witnesses are the coins used in the ordinary commitments), and therefore the entire protocol constitutes a (complicated) commitment scheme. Finally, one can establish the non-oblivious property by using the knowledge extractor associated with the proof system. Note that we can only extract the commited input and part of the coins used at the first stage (i.e., the coin used in some of the ordinary commitments but not necessarily the coins used in all of them). Furthermore, it may be that we accept also in case the sequence of strings sent at the first stage does not correspond to any legitimate sequence (i.e., of commitments to the same value). However, if we extract one value then it is impossible for the sender to later decommit to a different value, because the extracted value does fit one of the individual commitments.

Other applications. Fortunately, Theorem 4.9.4 is the only place where *strong* witness indistinguishable proofs are used in this work. We believe that in many other applications of *strong* witness indistinguishable proofs, a modification analogous to the above can be carried out (in order to salvaged the application). A typical example appears in [15]. Indeed, the current situation is very unfortunate and we hope that it will be redeemed in the future. Specifically, we propose

the following open problem:

Open problem: *Construct constant-round* public-coin *strong witness indistinguishable proofs* (and proofs-of-knowledge) *with negligible error for* $\mathcal{NP}$, or prove that this cannot be done. Recall that zero-knowledge *arguments* of the above nature are known [13]. The challenge is in providing such *proofs*.

C.4 On Non-Interactive Zero-Knowledge

C.4.1 On NIZK with efficient prover strategies

In continuation to Remark 4.10.6 and following [39], we briefly discuss the issues that arise when wishing to extend Construction 4.10.4 to arbitrary trapdoor permutations. Recall that Remark 4.10.6 focuses on a family of trapdoor permutations of the form $\{f_\alpha : \{0,1\}^{|\alpha|} \to \{0,1\}^{|\alpha|}\}_{\alpha \in \overline{I}}$, where $\overline{I}$ is efficiently recognizable. Unfortunately, no such family is known, and thus we first extend the treatment to the case in which $\overline{I}$ is not necessarily efficiently recognizable. The problem we encounter is that the prover may select (and send along) a function that is not in the family (i.e., an α not in $\overline{I}$). In such a case, the function is not necessarily 1-1, and consequently, the soundness property may be violated. This concern can be addressed by using a simple non-interactive (zero-knowledge) proof that the function is "typically 1-1" (or, equivalently, is "almost onto the designated range"). The proof proceeds by presenting inverses (under the function) of random elements specified in the reference string. Note that, for any fixed polynomial p, we can only prove that the function is 1-1 on at least a $1 - (1/p(n))$ fraction of the designated range, but this suffices for moderate soundness of the entire proof system (which in turn can be amplified by repetitions). For further details, consult [39].

Although the known candidate trapdoor permutations can be modified to fit the above form, we wish to further generalize the result so that any *enhanced* trapdoor permutation (as in Definition C.1.1) can be used. This can be done by letting the reference string consist of the coin-sequences used by the domain-sampling algorithm (rather than of elements of the function's domain). By virtue of the enhanced hardness condition (i.e., Eq. (C.3)), the security of the hardcore is preserved, and so is the zero-knowledge property.

As stated at the end of Section C.1, in contrast to what was claimed in Remark 4.10.6, we do not known how to extend the construction to arbitrary (rather than enhanced) trapdoor permutation.

C.4.2 On Adaptive NIZKs

In Definition 4.10.15, the *adaptive zero-knowledge* condition should be quantified only over efficiently computable input-selection strategies. Furthermore, it seems that also the witness-selection strategies should be restricted to ones

implemented by polynomial-size circuits. The revised form is presented in Definition 5.4.22.

A few words regarding the proof of Theorem 4.10.16 are in place. The (two-stage) simulation procedure itself is sketched in Footnote 29. Recall that at the first stage, we generate matrices at random, and replace the useful matrices by all-zero matrices (i.e., matrices of f-images that have preimages with hard-core value equal to zero). In the second stage, when given an adaptively chosen graph, we reveal all elements of all non-useful matrices and the required elements of the useful matrices, where revealing an element means revealing the corresponding f-preimage. In establishing the quality of this simulation procedure, we rely on the fact that the input graph as well as a Hamiltonian cycle in it are determined by a polynomial-size circuit. Loosely speaking, assuming towards the contradiction that the simulation can be distinguished from the real proof, we construct a circuit that distinguishes a sequence of random $f(x)$'s with $b(x) = 0$ from random $f(x)$'s with $b(x) = 1$. The "b-distinguisher" places the tested f-images in the suitable entries of useful matrices, fills up the rest of the entries of the useful matrices with elements it generates in $\{f(x) : b(x) = 0\}$, and fills the entries of non-useful by random f-images that it generates. Next, it determines the input graph and the corresponding Hamiltonian cycle (by using the above polynomial-size circuit), and acts as the real prover. Finally, it feeds the original distinguisher with the corresponding output. Observe that in case the given sequence of $f(x)$'s satisfies $b(x) = 0$ (resp. $b(x) = 1$) for each $f(x)$, the b-distinguisher produces outputs distributed exactly as in the simulation (resp., the real proof).

C.5 Some developments regarding zero-knowledge

A recent result by Barak [13] calls for re-evaluation of the significance of all negative results regarding black-box zero-knowledge[7] (as defined in Definition 4.5.10). In particular, relying on standard intractability assumptions, Barak presents round-efficient public-coin zero-knowledge arguments for $\mathcal{NP}$ (using non-black-box simulators), whereas only $\mathcal{BPP}$ can have such black-box zero-knowledge arguments (see comment following Theorem 4.5.11). Interestingly, Barak's simulator works in strict (rather than expected) probabilistic polynomial-time, addressing an open problem mentioned in Section 4.12.3. Barak's result is further described in Section C.5.2

In Section C.5.1, we review some recent progress achieved with respect to the preservation of zero-knowledge under concurrent composition. We seize the oppertunity to provide a wider perspective on the question of preservation of zero-knowledge under various forms of protocol composition operations.

We mention that the two problems discussed in this section (i.e., the "preservation of security under various forms of protocol composition" and the "use of of

[7] Specifically, one should reject the interpretation, offered in Section 4.5 (see Sections 4.5.0, 4.5.4.0 and 4.5.4.2), by which such results regarding black-box zero-knowledge indicate inherent limitations of zero-knowledge.

the adversary's program within the proof of security") arise also with respect to the security of other cryptographic primitives. Thus, the study of zero-knowledge proofs serve as a good bench-mark for the study of various problems regarding cryptographic protocols.

C.5.1 Composing zero-knowledge protocols

A natural question regarding zero-knowledge proofs (and arguments) is whether the zero-knowledge condition is preserved under a variety of composition operations. Three types of composition operation were considered in the literature: *sequential composition*, *parallel composition* and *concurrent composition*. We note that the preservation of zero-knowledge under these forms of composition is not only interesting on its own sake, but rather also sheds light of the preservation of the security of general protocols under these forms of composition.

We stress that when we talk of composition of protocols (or proof systems) we mean that the honest users are supposed to follow the prescribed program (specified in the protocol description) that refers to a single execution. That is, the actions of honest parties in each execution are independent of the messages they received in other executions. The adversary, however, may coordinate the actions it takes in the various executions, and in particular its actions in one execution may depend also on messages it received in other executions.

Let us motivate the asymmetry between the independence of executions assumed of honest parties but not of the adversary. Coordinating actions in different executions is typically difficult but not impossible. Thus, it is desirable to use composition (as defined above) rather than to use protocols that include inter-execution coordination-actions, which require users to keep track of all executions that they perform. Actually, trying to coordinate honest executions is even more problematic than it seems because one may need to coordinate executions of *different* honest parties (e.g., all employees of a big cooperation or an agency under attack), which in many cases is highly unrealistic. On the other hand, the adversary attacking the system may be willing to go into the extra trouble of coordinating its attack in the various executions of the protocol.

For $T \in \{\texttt{sequential}, \texttt{parallel}, \texttt{concurrent}\}$, we say that a protocol is T-zero-knowledge if it is zero-knowledge under a composition of type T. The definitions of T-zero-knowledge are derived from the standard definition by considering appropriate adversaries (i.e., adversarial verifiers); that is, adversaries that can initiate a polynomial number of interactions with the prover, where these interactions are scheduled according to the type T.[8] The corresponding simulator (which, as usual, interacts with nobody) is required to produce an output that is computationally indistinguishable from the output of such a type T adversary.

[8] Without loss of generality, we may assume that the adversary never violates the scheduling condition; it may instead send an illegal message at the latest possible adequate time. Furthermore, without loss of generality, we may assume that all the adversary's messages are delivered at the latest possible adequate time.

C.5.1.1 Sequential Composition

In this case, the protocol is invoked (polynomially) many times, where each invocation follows the termination of the previous one. At the very least, security (e.g., zero-knowledge) should be preserved under sequential composition, or else the applicability of the protocol is highly limited (because one cannot safely use it more than once).

We mention that whereas the "simplified" version (i.e., without auxiliary inputs, as in Definition 4.3.2) is not closed under sequential composition (see [142]), the actual version (i.e., with auxiliary inputs, as in Definition 4.3.10) is closed under sequential composition (see Section 4.3.4). We comment that the same phenomena arises when trying to use a zero-knowledge proof as a sub-protocol inside larger protocols. Indeed, it is for these reasons that the augmentation of the "most basic" definition by auxiliary inputs was adopted in all subsequent works.[9]

C.5.1.2 Parallel Composition

In this case, (polynomially) many instances of the protocol are invoked at the same time and proceed at the same pace. That is, we assume a synchronous model of communication, and consider (polynomially) many executions that are totally synchronized so that the ith message in all instances is sent exactly (or approximately) at the same time. (Natural variants on this model are discussed below as well as at the end of Section C.5.1.3.)

It turns out that, in general, zero-knowledge is not closed under parallel composition. A simple counter-example (to the "parallel composition conjecture") is depicted in Figure C.1. This counter-example, which is adapted from [142], consists of a simple protocol that is zero-knowledge (in a strong sense), but is not closed under parallel composition (not even in a very weak sense).

We comment that, at the 1980's, the study of parallel composition was interpreted mainly in the context of *round-efficient error reduction* (cf. [110, 142]); that is, the construction of full-fledge zero-knowledge proofs (with negligible soundness error) by composing (in parallel) a basic zero-knowledge protocol of high (but bounded away from 1) soundness error. Since alternative ways of constructing constant-round zero-knowledge proofs (and arguments) were found (cf. [141, 109, 61]), interest in parallel composition (of zero-knowledge protocols) has died. In retrospect, this was a conceptual mistake, because parallel composition (and mild extensions of this notion) capture the preservation of security in a fully synchronous (or almost-fully synchronous) communication network. We note that the almost-fully synchronous communication model is quite realistic in many settings, although it is certainly preferable not to assume even weak synchronism.

[9] Interestingly, the preliminary version of Goldwasser, Micali and Rackoff's work [164] used the "most basic" definition, whereas the final version of this work used the augmented definition. In some works, the "most basic" definition is used for simplicity, but typically one actually needs and means the augmented definition.

Consider a party P holding a random (or rather pseudorandom) function $f:\{0,1\}^{2n} \to \{0,1\}^n$, and willing to participate in the following protocol (with respect to security parameter n). The other party, called A for adversary, is supposed to send P a binary value $v \in \{1,2\}$ specifying which of the following cases to execute:

Case $v=1$: Party P uniformly selects $\alpha \in \{0,1\}^n$, and sends it to A, which is supposed to reply with a pair of n-bit long strings, denoted (β,γ). Party P checks whether or not $f(\alpha\beta)=\gamma$. In case equality holds, P sends A some secret information.

Case $v=2$: Party A is supposed to uniformly select $\alpha \in \{0,1\}^n$, and sends it to P, which selects uniformly $\beta \in \{0,1\}^n$, and replies with the pair $(\beta, f(\alpha\beta))$.

Observe that P's strategy is zero-knowledge (even w.r.t auxiliary-inputs): Intuitively, if the adversary A chooses the case $v=1$, then it is infeasible for A to guess a passing pair (β,γ) with respect to the random α selected by P. Thus, except with negligible probability (when it may get secret information), A does not obtain anything from the interaction. On the other hand, if the adversary A chooses the case $v=2$, then it obtains a pair that is indistinguishable from a uniformly selected pair of n-bit long strings (because β is selected uniformly by P, and for any α the value $f(\alpha\beta)$ looks random to A).
In contrast, if the adversary A can conduct two concurrent[a] executions with P, then it may learn the desired secret information: In one session, A sends $v=1$ while in the other it sends $v=2$. Upon receiving P's message, denoted α, in the first session, A sends α as its own message in the second session, obtaining a pair $(\beta, f(\alpha\beta))$ from P's execution of the second session. Now, A sends the pair $(\beta, f(\alpha\beta))$ to the first session of P, this pair passes the check, and so A obtains the desired secret.

[a] Dummy messages may be added (in both cases) in order to obtain the following scheduling in the perfectly parallel case.

Figure C.1: A counter-example (adapted from [142]) to the parallel repetition conjecture for zero-knowledge protocols.

Although, in general, zero-knowledge is not closed under parallel composition, under standard intractability assumptions (e.g., the intractability of factoring), there exists zero-knowledge protocols for $\mathcal{NP}$ that are closed under parallel composition. Furthermore, these protocols have a constant number of rounds (cf. [136] for proofs and [100] for arguments).[10] Both results extend also to concurrent composition in a synchronous communication model, where the extension is in allowing protocol invocations to start at different (synchronous) times (and in particular executions may overlap but not run simultaneously).

We comment that parallel composition is problematic also in the context of reducing the soundness error of arguments (cf. [31]), but our focus here is on the zero-knowledge aspect of protocols regardless if they are proofs, arguments or neither.

[10] In case of parallel-zero-knowledge *proofs*, there is no need to specify the soundness error because it can always be reduced via parallel composition. As mentioned below, this is not the case with respect to arguments.

C.5.1.3 Concurrent Composition (with and without timing)

Concurrent composition generalizes both sequential and parallel composition. Here (polynomially) many instances of the protocol are invoked at arbitrary times and proceed at arbitrary pace. That is, we assume an asynchronous (rather than synchronous) model of communication.

In the 1990's, when extensive two-party (and multi-party) computations became a reality (rather than a vision), it became clear that it is (at least) desirable that cryptographic protocols maintain their security under concurrent composition (cf. [94]). In the context of zero-knowledge, concurrent composition was first considered by Dwork, Naor and Sahai [100]. Actually, two models of concurrent composition were considered in the literature, depending on the underlying model of communication (i.e., a *purely asynchronous model* and an *asynchronous model with timing*). Both models cover sequential and parallel composition as special cases.

Concurrent composition in the pure asynchronous model. Here we refer to the standard model of asynchronous communication. In comparison to the timing model, the pure asynchronous model is a simpler model and using it requires no assumptions about the underlying communication channels. However it seems harder to construct concurrent zero-knowledge protocols for this model. In particular, for a while it was not known whether concurrent zero-knowledge proofs for $\mathcal{NP}$ exist at all (in this model). Under standard intractability assumptions (e.g., the intractability of factoring), this question was affirmatively resolved by Richardson and Kilian [246]. Following their work, research has focused on determining the round-complexity of concurrent zero-knowledge proofs for $\mathcal{NP}$. Currently, this question is still opened, and the state of the art regarding it is as follows:

- Under standard intractability assumptions, every language in $\mathcal{NP}$ has a concurrent zero-knowledge proof with *almost-logarithmically* many rounds (cf. [238], building upon [188], which in turn builds over [246]). Furthermore, the zero-knowledge property can be demonstrated using a black-box simulator (see definition in Section 4.5.4.2 and discussion in Section C.5.2).

- Black-box simulator cannot demonstrated the concurrent zero-knowledge property of non-trivial proofs (or arguments) having significantly less than logarithmically-many rounds (cf. Canetti *et. al.* [73]).[11]

- Recently, Barak [13] demonstrated that the "black-box simulation barrier" can be bypassed. With respect to concurrent zero-knowledge he only obtained the following partial result: under standard intractability assumptions, every language in $\mathcal{NP}$ has a constant-round zero-knowledge

[11] By *non-trivial* proof systems we mean ones for languages outside $\mathcal{BPP}$, whereas by *significantly less than logarithmic* we mean any function $f : \mathbb{N} \to \mathbb{N}$ satisfying $f(n) = o(\log n / \log\log n)$. In contrast, by *almost-logarithmic* we mean any function f satisfying $f(n) = \omega(\log n)$.

argument (rather than proof) that maintain security as long as an a-priori bounded (polynomial) number of executions take place concurrently. (The length of the messages in his protocol grows linearly with this a-priori bound.)

Thus, it is currently unknown whether or not *constant-round* arguments for $\mathcal{NP}$ may be concurrent zero-knowledge (in the pure asynchronous model).

Concurrent composition under the timing model: A model of naturally-limited asynchronousness (which certainly covers the case of parallel composition) was introduced by Dwork, Naor and Sahai [100]. Essentially, they assume that each party holds a local clock such that the relative clock rates are bounded by an a-priori known constant, and consider protocols that employ time-driven operations (i.e., `time-out` in-coming messages and `delay` out-going messages). The benefit of the timing model is that it is known to construct concurrent zero-knowledge protocols for it. Specifically, using standard intractability assumptions, *constant-round* arguments and proofs that are concurrent zero-knowledge under the timing model do exist (cf. [100] and [136], respectively). The disadvantages of the timing model are discussed next.

The timing model consists of the *assumption* that talking about the actual timing of events is meaningful (at least in a weak sense) and of the *introduction of time-driven operations*. The timing assumption amounts to postulating that each party holds a local clock and knows a global bound, denoted $\rho \geq 1$, on the relative rates of the local clocks.[12] Furthermore, it is postulated that the parties know a (pessimistic) bound, denoted Δ, on the message-delivery time (which also includes the local computation and handling times). In our opinion, these timing assumptions are most reasonable, and are unlikely to restrict the scope of applications for which concurrent zero-knowledge is relevant. We are more concerned about the effect of the time-driven operations introduced in the timing model. Recall that these operations are the `time-out` of in-coming messages and the `delay` of out-going messages. Furthermore, typically the delay period is at least as long as the time-out period, which in turn is at least Δ (i.e., the time-out period must be at least as long as the pessimistic bound on message-delivery time so not to disrupt the proper operation of the protocol). This means that the use of these time-driven operations yields slowing down the execution of the protocol (i.e., running it at the rate of the pessimistic message-delivery time rather than at the rate of the actual message-delivery time, which is typically much faster). Still, in absence of more appealing alternatives (i.e., a constant-round concurrent zero-knowledge protocol for the pure asynchronous model), the use of the timing model may be considered reasonable. (We comment than other alternatives to the timing-model include various set-up assumptions; cf. [70, 89].)

[12] The rate should be computed with respect to reasonable intervals of time; for example, for Δ as defined below, one may assume that a time period of Δ units is measured as Δ' units of time on the local clock, where $\Delta/\rho \leq \Delta' \leq \rho\Delta$.

Back to parallel composition: Given our opinion about the timing model, it is not surprising that we consider the problem of parallel composition almost as important as the problem of concurrent composition in the timing model. Firstly, it is quite reasonable to assume that the parties' local clocks have approximately the same rate, and that drifting is corrected by occasional clock synchronization. Thus, it is reasonable to assume that the parties have approximately-good estimate of some global time. Furthermore, the global time may be partitioned into phases, each consisting of a constant number of rounds, so that each party wishing to execute the protocol just delays its invocation to the beginning of the next phase. Thus, concurrent execution of (constant-round) protocols in this setting amounts to a sequence of (time-disjoint) almost-parallel executions of the protocol. Consequently, proving that the protocol is parallel zero-knowledge suffices for concurrent composition in this setting.

Relation to resettable zero-knowledge. Going to the other extreme, we mention that there exists a natural model of zero-knowledge that is even stronger than concurrent zero-knowledge (even in the pure asynchronous model). Specifically, "resettable zero-knowledge" as defined in [70], implies concurrent zero-knowledge.

C.5.2 Using the adversary's program in the proof of security

Recall that the definition of zero-knowledge proofs states that whatever an efficient adversary can compute after interacting with the prover, can actually be efficiently computed from scratch by a so-called *simulator* (which works without interacting with the prover). Although the simulator may depend arbitrarily on the adversary, the need to present a simulator for each feasible adversary seems to require the presentation of a universal simulator that is given the adversary's strategy (or program) as another auxiliary input. The question addressed in this section is how can the universal simulator use the adversary's program.

The adversary's program (or strategy) is actually a function determining for each possible view of the adversary (i.e., its input, random choices and the message it has received so far) which message will be sent next. Thus, we identify the adversary's program with this next-message function. As stated above, until very recently, all universal simulators (constructed towards demonstrating zero-knowledge properties) have used the adversary's program (or rather its next-message function) as a black-box (i.e., the simulator invoked the next-message function on a sequence of arguments of its choice). Furthermore, in view of the presumed difficulty of "reverse engineering" programs, it was commonly believed that nothing is lost by restricting attention to simulators, called black-box simulators, that only make black-box usage of the adversary's program. Consequently, Goldreich and Krawczyk conjectured that impossibility results regarding black-box simulation represent inherent limitations of zero-knowledge itself, and studied the limitations of the former [142].

> In particular, they showed that parallel composition of the protocol of Construction 4.4.7 (as well as of any constant-round public-coin protocol) *cannot be proven to be zero-knowledge using a black-box simulator*, unless the language (i.e., 3-Colorability) is in $\mathcal{BPP}$. In fact their result refers to any constant-round public-coin protocol with negligible soundness error, regardless of how such a protocol is obtained. This result was taken as strong evidence towards the conjecture that constant-round public-coin protocol with negligible soundness error *cannot be zero-knowledge* (unless the language is in $\mathcal{BPP}$).
>
> Similarly, as mentioned in Section C.5.1.3, it was shown that protocols of sub-logarithmic number of rounds *cannot be proven to be concurrent zero-knowledge via a black-box simulator* [73], and this was taken as evidence towards the conjecture that such protocols cannot be *concurrent zero-knowledge*.

In contrast to these conjectures and supportive evidence, Barak showed how to constructed non-black-box simulators and obtained several results that were known to be unachievable via black-box simulators [13]. In particular, under standard intractability assumption (see also [15]), he presented constant-round public-coin zero-knowledge arguments with negligible soundness error for any language in $\mathcal{NP}$. (Moreover, the simulator runs in strict polynomial-time, which is impossible for black-box simulators of non-trivial constant-round protocols [17].) Furthermore, this protocol preserves zero-knowledge under a fixed[13] polynomial number of concurrent executions, in contrast to the result of [73] (regarding black-box simulators) that holds also in that restricted case. Thus, Barak's result calls for the re-evaluation of many common believes. Most concretely, it says that results regarding black-box simulators do not reflect inherent limitations of zero-knowledge (but rather an inherent limitation of a natural way of demonstrating the zero-knowledge property). Most abstractly, it says that there are meaningful ways of using a program other than merely invoking it as a black-box.

Does this means that a method was found to "reverse engineer" programs or to "understand" them? We believe that the answer is negative. Barak [13] is using the adversary's program in a significant way (i.e., more significant than just invoking it), without "understanding" it. *So how does he use the program?*

The key idea underlying Barak's protocol [13] is to have the prover prove that either the original NP-assertion is valid or that he (i.e., the prover) "knows the verifier's residual strategy" (in the sense that it can predict the next verifier message). Indeed, in a real interaction (with the honest verifier), it is infeasible for the prover to predict the next verifier message, and so computational-soundness of the protocol follows. However, a simulator that is given the code of the verifier's strategy (and not merely oracle access to that code), can produce a

[13] The protocol depends on the polynomial bounding the number of executions, and thus is not known to be concurrent zero-knowledge (because the latter requires to fix the protocol and then consider any polynomial number of concurrent executions).

valid proof of the disjunction by properly executing the sub-protocol using its knowledge of an NP-witness for the second disjunctive. The simulation is computational indistinguishable from the real execution, provided that one cannot distinguish an execution of the sub-protocol in which one NP-witness (i.e., an NP-witness for the original assertion) is used from an execution in which the second NP-witness (i.e., an NP-witness for the auxiliary assertion) is use. That is, the sub-protocol should be a *witness indistinguishable* argument system (see Sections 4.6 and 4.8). We warn the reader that the actual implementation of the above idea requires overcoming several technical difficulties (cf. [13, 15]).

Perspective. In retrospect, taking a wide perspective, it should not come as a surprise that the program's code yields extra power beyond black-box access to it. Feeding a program with its own code (or part of it) is the essence of the diagonalization technique, and this too is done without "reverse engineering". Furthermore, various non-black-box techniques have appeared before in the cryptographic setting, but they were used in the more natural context of *devising an attack* on an (artificial) insecure scheme (e.g., towards proving the failure of the "Random Oracle Methodology" [69] and the impossibility of software obfuscation [16]). In contrast, in [13] (and [14]) the code of the adversary is being used within a sophisticated proof of security. What we wish to highlight here is that *non-black-box usage of programs is relevant also to proving* (rather than to disproving) *the security of systems*.

Digest: Witness Indistinguishability and the FLS-Technique

The above description (of [13]), as well as several other sophisticated constructions of zero-knowledge protocols (e.g., [108, 246]), makes crucial use of a technique introduced by Feige, Lapidot and Shamir [108], which in turn is based on the notion of witness indistinguishability (introduced by Feige and Shamir [110]). This technique, hereafter referred to as the FLS-technique, was used in Construction 4.10.12, but we wish to further discuss it below.

Following is a sketchy description of a special case of the FLS-technique, whereas the abovementioned application uses a more general version (which refers to proofs of knowledge, as defined in Section 4.7).[14] In this special case, the technique consists of the following construction schema, which uses witness indistinguishable protocols for $\mathcal{NP}$ in order to obtain zero-knowledge protocols for $\mathcal{NP}$. On common input $x \in L$, where $L = L_R$ is the NP-set defined by the witness relation R, the following two steps are performed:

1. The parties generate an instance x' for an auxiliary NP-set L', where L' is defined by a witness relation R'. The generation protocol in use must satisfy the following two conditions:

[14] In the general case, the generation protocol may generate an instance x' in L', but it is infeasible for the prover to obtain a corresponding witness (i.e., a w' such that $(x', w') \in R'$). In the second step, the sub-protocol in use ought to be a proof of knowledge, and computational-soundness of the main protocol will follows (because otherwise the prover, using a knowledge extractor, can obtain a witness for $x' \in L'$).

(a) If the verifier follows its prescribed strategy then no matter which feasible strategy is used by the prover, with high probability, the protocol's outcome is a NO-instance of L'.

(b) Loosely speaking, there exists an efficient (non-interactive) procedure for producing a (random) transcript of the generation protocol *along with an NP-witness for the corresponding outcome* (which is a YES-instance of L') such that the procuded transcript is computationally indistinguishable from the transcript of a real execution of the protocol.

2. The parties execute a witness indistinguishable protocol for the set L'' defined by the witness relation $R'' = \{((u,u'),(v,v')) : (u,v) \in R \vee (u',v') \in R'\}$. The sub-protocol is such that the corresponding prover can be implemented in probabilistic polynomial-time given an NP-witness for $(u,u') \in L''$. The sub-protocol is invoked on common input (x,x'), where x' is the outcome of Step 1, and the sub-prover is invoked with the corresponding NP-witness as auxiliary input (i.e., with (w,λ), where w is the NP-witness for x given to the main prover).

The computational-soundness of the above protocol follows by Property (a) of the generation protocol (i.e., with high probability $x' \notin L'$, and so $x \in L$ follows by the soundness of the protocol used in Step 2). To demonstrate the zero-knowledge property, we first generate a simulated transcript of Step 1 (with outcome $x' \in L'$) along with an adequate NP-witness (i.e., w' such that $(x',w') \in R'$), and then emulates Step 2 by feeding the sub-prover strategy with the NP-witness (λ,w'). Combining Property (b) of the generation protocol and the witness indistinguishability property of the protocol used in Step 2, the simulation is indistinguishable from the real execution.

C.6 Miscellaneous

C.6.1 Additional Corrections

1. Regarding Constriction 4.10.7 and the proof of Proposition 4.10.9: The current description in terms of two mappings π_1, π_2 is confusing and even inaccurate. Instead one should identify the rows (resp., columns) of H with $[n]$ and use one permutation π over $[n]$ (which supposedly maps the vertices of G to those of H). Alternatively, one may compose this permutation π with the two (1-1) mappings ψ_i's (where $\psi_i : [n] \to [n^3]$), and obtain related π_i's (i.e., $\pi_i(v) = \psi_i(\pi(v))$), which should be used as in the original text.

C.6.2 Additional Comments

1. In continuation to Sections 4.7 and 4.9.2, we mention that the round-efficient argument system of [109] is actually an "argument of knowledge"

(with negligible error). The interested reader is referred to [17] for further improvements regarding such proof systems. Essentially, using a relaxed but satisfactory definition of an argument-of-knowledge, the latter work presents a constant-round zero-knowledge argument-of-knowledge with *strict* (rather than expected) probabilistic polynomial-time simulator and knowledge-extractor.

2. The sequential composition lemma for zero-knowledge protocols (i.e., Lemma 4.3.11) is due to [153].

3. We mention that the notions of *strong* witness indistinguishability (Definition 4.6.2) and *strong* proofs of knowledge (Section 4.7.6), and the Hidden Bit Model (Section 4.10.2) have first appeared in early versions of this work.

C.6.3 Typos etc

1. In the guideline for Exercise 11 of Chapter 2, the term $\mathsf{E}\mathrm{cyc}_f(U_n)]$ should be $\mathsf{E}[\mathrm{cyc}_f(U_n)]$. In the exercise itself, one should also address the case in which $\mathrm{cyc}_f(x)$ is undefined for some x's.

   Author's Note: *First draft written mainly in 2002. Revised in January 2003.*

Bibliography

[1] L.M. Adleman and M. Huang. *Primality Testing and Abelian Varieties Over Finite Fields.* Springer-Verlag Lecture Notes in Computer Science (Vol. 1512), 1992. Preliminary version in *19th ACM Symposium on the Theory of Computing*, 1987.

[2] W. Aiello and J. Håstad. Perfect Zero-Knowledge Languages can be Recognized in Two Rounds. In *28th IEEE Symposium on Foundations of Computer Science*, pages 439–448, 1987.

[3] M. Ajtai. Generating Hard Instances of Lattice Problems. In *28th ACM Symposium on the Theory of Computing*, pages 99–108, 1996.

[4] M. Ajtai, J. Komlos, E. Szemerédi. Deterministic Simulation in LogSpace. In *19th ACM Symposium on the Theory of Computing*, pages 132–140, 1987.

[5] W. Alexi, B. Chor, O. Goldreich and C.P. Schnorr. RSA/Rabin Functions: Certain Parts are As Hard As the Whole. *SIAM Journal on Computing*, Vol. 17, April 1988, pages 194–209.

[6] N. Alon and J.H. Spencer. *The Probabilistic Method*, John Wiley & Sons, Inc., 1992.

[7] J.H. An and M. Bellare. Constructing VIL-MACs from FIL-MACs: Message Authentication under Weakened Assumptions. In *Crypto99*, Springer Lecture Notes in Computer Science (Vol. 1666), pages 252–269.

[8] T.M. Apostol. *Introduction ot Analytic Number Theory.* Springer, 1976.

[9] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics.* McGraw-Hill, 1998.

[10] L. Babai. Trading Group Theory for Randomness. In *17th ACM Symposium on the Theory of Computing*, pages 421–420, 1985.

[11] E. Bach. *Analytic Methods in the Analysis and Design of Number-Theoretic Algorithms.* ACM Distinguished Dissertation (1984), MIT Press, Cambridge MA, 1985.

[12] E. Bach and J. Shallit. *Algorithmic Number Theory* (Volume I: Efficient Algorithms). MIT Press, 1996.

[13] B. Barak. How to Go Beyond the Black-Box Simulation Barrier. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 106–115, 2001.

[14] B. Barak. Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. In *43th IEEE Symposium on Foundations of Computer Science*, to appear, 2002.

[15] B. Barak and O. Goldreich, Universal arguments and their applications. In the *17th IEEE Conference on Computational Complexity*, pages 194–203, 2002.

[16] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im)possibility of software obfuscation. In *Crypto01*, Springer-Verlag Lecture Notes in Computer Science (Vol. 2139), pages 1–18.

[17] B. Barak and Y. Lindell. Strict Polynomial-time in Simulation and Extraction. In *34th ACM Symposium on the Theory of Computing*, pages 484–493, 2002.

[18] D. Beaver. Foundations of Secure Interactive Computing. In *Crypto91*, Springer-Verlag Lecture Notes in Computer Science (Vol. 576), pages 377–391.

[19] D. Beaver. Secure Multi-Party Protocols and Zero-Knowledge Proof Systems Tolerating a Faulty Minority. *Journal of Cryptology*, Vol. 4, pages 75–122, 1991.

[20] M. Bellare. A Note on Negligible Functions. *Journal of Cryptology*, Vol. 15, pages 271–284, 2002.

[21] M. Bellare, R. Canetti and H. Krawczyk. Pseudorandom functions Revisited: The Cascade Construction and its Concrete Security. In *37th IEEE Symposium on Foundations of Computer Science*, pages 514–523, 1996.

[22] M. Bellare, R. Canetti and H. Krawczyk. Keying Hash Functions for Message Authentication. In *Crypto96*, Springer Lecture Notes in Computer Science (Vol. 1109), pages 1–15.

[23] M. Bellare, R. Canetti and H. Krawczyk. Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols. In *30th ACM Symposium on the Theory of Computing*, pages 419–428, 1998.

[24] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Crypto98*, Springer Lecture Notes in Computer Science (Vol. 1462), pages 26–45.

[25] M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. In *Crypto92*, Springer-Verlag Lecture Notes in Computer Science (Vol. 740), pages 390–420.

[26] M. Bellare, O. Goldreich and S. Goldwasser. Incremental Cryptography: the Case of Hashing and Signing. In *Crypto94*, Springer-Verlag Lecture Notes in Computer Science (Vol. 839), pages 216–233, 1994.

[27] M. Bellare, O. Goldreich and S. Goldwasser. Incremental Cryptography and Application to Virus Protection. In *27th ACM Symposium on the Theory of Computing*, pages 45–56, 1995.

[28] M. Bellare, O. Goldreich and H. Krawczyk. Stateless Evaluation of Pseudorandom Functions: Security beyond the Birthday Barrier. In *Crypto99*, Springer Lecture Notes in Computer Science (Vol. 1666), pages 270–287.

[29] M. Bellare, R. Guerin and P. Rogaway. XOR MACs: New Methods for Message Authentication using Finite Pseudorandom Functions. In *Crypto95*, Springer-Verlag Lecture Notes in Computer Science (Vol. 963), pages 15–28.

[30] M. Bellare, S. Halevi, A. Sahai and S. Vadhan. Trapdoor Functions and Public-Key Cryptosystems. In *Crypto98*, Springer Lecture Notes in Computer Science (Vol. 1462), pages 283–298.

[31] M. Bellare, R. Impagliazzo and M. Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols? In *38th IEEE Symposium on Foundations of Computer Science*, pages 374–383, 1997.

[32] M. Bellare, J. Kilian and P. Rogaway. The Security of Cipher Block Chaining. In *Crypto94*, Springer-Verlag Lecture Notes in Computer Science (Vol. 839), pages 341–358.

[33] M. Bellare and S. Micali. How to Sign Given Any Trapdoor Function. *Journal of the ACM*, Vol. 39, pages 214–233, 1992.

[34] D. Beaver, S. Micali and P. Rogaway. The Round Complexity of Secure Protocols. In *22nd ACM Symposium on the Theory of Computing*, pages 503–513, 1990.

[35] M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. In *1st Conf. on Computer and Communications Security*, ACM, pages 62–73, 1993.

[36] M. Bellare and P. Rogaway. Entity Authentication and Key Distribution. In *Crypto93*, Springer-Verlag Lecture Notes in Computer Science (Vol. 773), pages 232–249, 1994.

[37] M. Bellare and P. Rogaway. Provably Secure Session Key Distribution: The Three Party Case. In *27th ACM Symposium on the Theory of Computing*, pages 57–66, 1995.

[38] M. Bellare and P. Rogaway. The Exact Security of Digital Signatures: How to Sign with RSA and Rabin. In *EuroCrypt96*, Springer Lecture Notes in Computer Science (Vol. 1070).

[39] M. Bellare and M. Yung. Certifying Permutations: Noninteractive Zero-Knowledge Based on Any Trapdoor Permutation. *Journal of Cryptology*, Vol. 9, pages 149-166, 1996.

[40] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the Theory of Average Case Complexity. *Journal of Computer and System Science*, Vol. 44, No. 2, April 1992, pages 193–219.

[41] M. Ben-Or, R. Canetti and O. Goldreich. Asynchronous Secure Computation. In *25th ACM Symposium on the Theory of Computing*, pages 52–61, 1993. See details in [64].

[42] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali and P. Rogaway. Everything Provable is Probable in Zero-Knowledge. In *Crypto88*, Springer-Verlag Lecture Notes in Computer Science (Vol. 403), pages 37–56, 1990

[43] M. Ben-Or, S. Goldwasser, J. Kilian and A. Wigderson. Multi-Prover Interactive Proofs: How to Remove Intractability. In *20th ACM Symposium on the Theory of Computing*, pages 113–131, 1988.

[44] M. Ben-Or, S. Goldwasser and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In *20th ACM Symposium on the Theory of Computing*, pages 1–10, 1988.

[45] E.R. Berlekamp. Factoring Polynomials over Large Finite Fields. *Mathematics of Computation*, Vol. 24, pages 713–735, 1970.

[46] E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg. On the Inherent Intractability of Certain Coding Problems. *IEEE Trans. on Inform. Theory*, 1978.

[47] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. UMAC: Fast and Secure Message Authentication. In *Crypto99*, Springer Lecture Notes in Computer Science (Vol. 1666), pages 216–233.

[48] M. Blum. How to Exchange Secret Keys. *ACM Trans. Comput. Sys.*, Vol. 1, pages 175–193, 1983.

[49] M. Blum. Coin Flipping by Phone. In *the 24th IEEE Computer Conference* (*CompCon*), pages 133–137, February 1982. See also *SIGACT News*, Vol. 15, No. 1, 1983.

[50] L. Blum, M. Blum and M. Shub. A Simple Secure Unpredictable Pseudo-Random Number Generator. *SIAM Journal on Computing*, Vol. 15, 1986, pages 364–383.

[51] M. Blum, A. De Santis, S. Micali, and G. Persiano. Non-Interactive Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, Vol. 20, No. 6, pages 1084–1118, 1991. (Considered the journal version of [52].)

[52] M. Blum, P. Feldman and S. Micali. Non-Interactive Zero-Knowledge and its Applications. In *20th ACM Symposium on the Theory of Computing*, pages 103–112, 1988. See [51].

[53] M. Blum and S. Goldwasser. An Efficient Probabilistic Public-Key Encryption Scheme which hides all partial information. In *Crypto84*, Lecture Notes in Computer Science (Vol. 196) Springer-Verlag, pages 289–302.

[54] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. *SIAM Journal on Computing*, Vol. 13, pages 850–864, 1984. Preliminary version in *23rd IEEE Symposium on Foundations of Computer Science*, 1982.

[55] R. Boppana, J. Håstad, and S. Zachos. Does Co-NP Have Short Interactive Proofs? *Information Processing Letters*, 25, May 1987, pp. 127-132.

[56] J.B. Boyar. Inferring Sequences Produced by Pseudo-Random Number Generators. *Journal of the ACM*, Vol. 36, pages 129–141, 1989.

[57] G. Brassard. A Note on the Complexity of Cryptography. *IEEE Trans. on Inform. Th.*, Vol. 25, pages 232–233, 1979.

[58] G. Brassard. Quantum Information Processing: The Good, the Bad and the Ugly. In *Crypto97*, Springer Lecture Notes in Computer Science (Vol. 1294), pages 337–341.

[59] G. Brassard, D. Chaum and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer and System Science*, Vol. 37, No. 2, pages 156–189, 1988. Preliminary version by Brassard and Crépeau in *27th IEEE Symposium on Foundations of Computer Science*, 1986.

[60] G. Brassard and C. Crépeau. Zero-Knowledge Simulation of Boolean Circuits. In *Crypto86*, Springer-Verlag Lecture Notes in Computer Science (Vol. 263), pages 223–233, 1987.

[61] G. Brassard, C. Crépeau and M. Yung. Constant-Round Perfect Zero-Knowledge Computationally Convincing Protocols. *Theoretical Computer Science*, Vol. 84, pages 23–52, 1991.

[62] E.F. Brickell and A.M. Odlyzko. Cryptanalysis: A Survey of Recent Results. In *Proceedings of the IEEE*, Vol. 76, pages 578–593, 1988.

[63] C. Cachin and U. Maurer. Unconditional security against memory-bounded adversaries. In *Crypto97*, Springer Lecture Notes in Computer Science (Vol. 1294), pages 292–306.

[64] R. Canetti. *Studies in Secure Multi-Party Computation and Applications*. Ph.D. Thesis, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel, June 1995. Available from from `http://theory.lcs.mit.edu/~tcryptol/BOOKS/ran-phd.html`.

[65] R. Canetti. Security and Composition of Multi-party Cryptographic Protocols. *Journal of Cryptology*, Vol. 13, No. 1, pages 143–202, 2000.

[66] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145, 2001. Full version (with different title) is available from *Cryptology ePrint Archive*, Report 2000/067.

[67] R. Canetti, I. Damgard, S. Dziembowski, Y. Ishai and T. Malkin. On adaptive versus non-adaptive security of multiparty protocols. *Journal of Cryptology*, to appear.

[68] R. Canetti, U. Feige, O. Goldreich and M. Naor. Adaptively Secure Multiparty Computation. In *28th ACM Symposium on the Theory of Computing*, pages 639–648, 1996.

[69] R. Canetti, O. Goldreich and S. Halevi. The Random Oracle Methodology, Revisited. In *30th ACM Symposium on the Theory of Computing*, pages 209–218, 1998.

[70] R. Canetti, O. Goldreich, S. Goldwasser, and S. Micali. Resettable Zero-Knowledge. In *32nd ACM Symposium on the Theory of Computing*, pages 235–244, 2000.

[71] R. Canetti, S. Halevi and A. Herzberg. How to Maintain Authenticated Communication in the Presence of Break-Ins. *Journal of Cryptology*, Vol. 13, No. 1, pages 61–106, 2000.

[72] R. Canetti and A. Herzberg. Maintaining Security in the Presence of Transient Faults. In *Crypto94*, Springer-Verlag Lecture Notes in Computer Science (Vol. 839), pages 425–439.

[73] R. Canetti, J. Kilian, E. Petrank and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires $\tilde{\Omega}(\log n)$ Rounds. In *33rd ACM Symposium on the Theory of Computing*, pages 570–579, 2001.

[74] R. Canetti, Y. Lindell, R. Ostrovsky and A. Sahai. Universally Composable Two-Party and Multi-Party Secure Computation. In *34th ACM Symposium on the Theory of Computing*, pages 494–503, 2002.

[75] E.R. Canfield, P. Erdos, and C. Pomerance. On a problem of Oppenheim concerning "factorisatio numerorum". *J. Number Theory*, Vol. 17, pages 1–28, 1983.

[76] L. Carter and M. Wegman. Universal Hash Functions. *Journal of Computer and System Science*, Vol. 18, 1979, pages 143–154.

[77] D. Chaum. Blind Signatures for Untraceable Payments. In *Crypto82*, Plenum Press, pages 199–203, 1983.

[78] D. Chaum, C. Crépeau and I. Damgård. Multi-party unconditionally Secure Protocols. In *20th ACM Symposium on the Theory of Computing*, pages 11–19, 1988.

[79] B. Chor, S. Goldwasser, S. Micali and B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In *26th IEEE Symposium on Foundations of Computer Science*, pages 383–395, 1985.

[80] B. Chor and E. Kushilevitz. A Zero-One Law for Boolean Privacy. *SIAM J. on Disc. Math.*, Vol. 4, pages 36–47, 1991.

[81] R. Cleve. Limits on the Security of Coin Flips when Half the Processors are Faulty. In *18th ACM Symposium on the Theory of Computing*, pages 364–369, 1986.

[82] J.D. Cohen and M.J. Fischer. A Robust and Verifiable Cryptographically Secure Election Scheme. In *26th IEEE Symposium on Foundations of Computer Science*, pages 372–382, 1985.

[83] A. Cohen and A. Wigderson. Dispensers, Deterministic Amplification, and Weak Random Sources. *30th IEEE Symposium on Foundations of Computer Science*, 1989, pages 14–19.

[84] R. Cramer and I. Damgård. New Generation of Secure and Practical RSA-based Signatures. In *Crypto96*, Springer Lecture Notes in Computer Science (Vol. 1109), pages 173–185.

[85] R. Cramer and V. Shoup. A Practical Public-Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attacks. In *Crypto98*, Springer-Verlag Lecture Notes in Computer Science (Vol. 1462), pages 13–25.

[86] C. Crépeau. Efficient Cryptographic Protocols Based on Noisy Channels. In *EuroCrypt97*, Springer, Lecture Notes in Computer Science (Vol. 1233), pages 306–317.

[87] I. Damgård. Collision Free Hash Functions and Public Key Signature Schemes. In *EuroCrypt87*, Springer-Verlag, Lecture Notes in Computer Science (Vol. 304), pages 203–216.

[88] I. Damgård. A Design Principle for Hash Functions. In *Crypto89*, Springer-Verlag Lecture Notes in Computer Science (Vol. 435), pages 416–427.

[89] I. Damgard. Concurrent Zero-Knowledge in Easy in Practice. Theory of Cryptography Library, 99-14, June 1999. `http://philby.ucsd.edu/cryptolib`. See also "Efficient Concurrent Zero-Knowledge in the Auxiliary String Model" (in *Eurocrypt'00*, 2000).

[90] I. Damgård, O. Goldreich, T. Okamoto and A. Wigderson. Honest Verifier vs Dishonest Verifier in Public Coin Zero-Knowledge Proofs. In *Crypto95*, Springer-Verlag Lecture Notes in Computer Science (Vol. 963), pages 325–338, 1995.

[91] A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano and A. Sahai. Robust Non-interactive Zero-Knowledge. In *Crypto01*, Springer Lecture Notes in Computer Science (Vol. 2139), pages 566–598.

[92] Y. Desmedt and Y. Frankel. Threshold Cryptosystems. In *Crypto89*, Springer-Verlag Lecture Notes in Computer Science (Vol. 435), pages 307–315.

[93] W. Diffie, and M.E. Hellman. New Directions in Cryptography. *IEEE Trans. on Info. Theory*, IT-22 (Nov. 1976), pages 644–654.

[94] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *23rd ACM Symposium on the Theory of Computing*, pages 542–552, 1991. Full version available from authors.

[95] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *Journal of the ACM*, Vol. 40 (1), pages 17–47, 1993.

[96] D. Dolev and A.C. Yao. On the Security of Public-Key Protocols. *IEEE Trans. on Inform. Theory*, Vol. 30, No. 2, pages 198–208, 1983.

[97] D. Dolev and H.R. Strong. Authenticated Algorithms for Byzantine Agreement. *SIAM Journal on Computing*, Vol. 12, pages 656–666, 1983.

[98] C. Dwork, U. Feige, J. Kilian, M. Naor and S. Safra. Low Communication Perfect Zero Knowledge Two Provers Proof Systems. In *Crypto92*, Springer Verlag, Lecture Notes in Computer Science (Vol. 740), pages 215–227.

[99] C. Dwork, and M. Naor. An Efficient Existentially Unforgeable Signature Scheme and its Application. *Journal of Cryptology*, Vol. 11 (3), pages 187–208, 1998

[100] C. Dwork, M. Naor, and A. Sahai. Concurrent Zero-Knowledge. In *30th STOC*, pages 409–418, 1998.

[101] S. Even and O. Goldreich. On the Security of Multi-party Ping-Pong Protocols. In *24th IEEE Symposium on Foundations of Computer Science*, pages 34–39, 1983.

[102] S. Even, O. Goldreich, and A. Lempel. A Randomized Protocol for Signing Contracts. *CACM*, Vol. 28, No. 6, 1985, pages 637–647.

[103] S. Even, O. Goldreich and S. Micali. On-line/Off-line Digital signatures. *Journal of Cryptology*, Vol. 9, 1996, pages 35–67.

[104] S. Even, A.L. Selman, and Y. Yacobi. The Complexity of Promise Problems with Applications to Public-Key Cryptography. *Inform. and Control*, Vol. 61, pages 159–173, 1984.

[105] S. Even and Y. Yacobi. Cryptography and NP-Completeness. In proceedings of *7th ICALP*, Springer-Verlag Lecture Notes in Computer Science (Vol. 85), pages 195–207, 1980. See [104].

[106] U. Feige. Error reduction by parallel repetition – the state of the art. Technical report CS95-32, Computer Science Department, Weizmann Institute of Science, Rehovot, ISREAL, 1995.

[107] U. Feige, A. Fiat and A. Shamir. Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, Vol. 1, 1988, pages 77–94.

[108] U. Feige, D. Lapidot, and A. Shamir. Multiple Non-Interactive Zero-Knowledge Proofs Under General Assumptions. *SIAM Journal on Computing*, Vol. 29 (1), pages 1–28, 1999.

[109] U. Feige and A. Shamir. Zero-Knowledge Proofs of Knowledge in Two Rounds. In *Crypto89*, Springer-Verlag Lecture Notes in Computer Science (Vol. 435), pages 526–544.

[110] U. Feige and A. Shamir. Witness Indistinguishability and Witness Hiding Protocols. In *22nd ACM Symposium on the Theory of Computing*, pages 416–426, 1990.

[111] W. Feller. *An Introduction to Probability Theory and Its Applications.* John Wiley, New York, 1968.

[112] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solution to Identification and Signature Problems. In *Crypto86*, Springer-Verlag Lecture Notes in Computer Science (Vol. 263), pages 186–189, 1987.

[113] M. Fischer, S. Micali, C. Rackoff, and D.K. Wittenberg. An Oblivious Transfer Protocol Equivalent to Factoring. Unpublished manuscript, 1986. Preliminary versions were presented in *EuroCrypt84*, and in the *NSF Workshop on Mathematical Theory of Security*, Endicott House (1985).

[114] R. Fischlin and C.P. Schnorr. Stronger Security Proofs for RSA and Rabin Bits. In *EuroCrypt97*, Springer Lecture Notes in Computer Science (Vol. 1233), pages 267–279, 1997.

[115] L. Fortnow, The Complexity of Perfect Zero-Knowledge. In *19th ACM Symposium on the Theory of Computing*, pages 204–209, 1987.

[116] A.M. Frieze, J. Håstad, R. Kannan, J.C. Lagarias, and A. Shamir. Reconstructing Truncated Integer Variables Satisfying Linear Congruences. *SIAM Journal on Computing*, Vol. 17, pages 262–280, 1988.

[117] O. Gaber and Z. Galil. Explicit Constructions of Linear Size Superconcentrators. *Journal of Computer and System Science*, Vol. 22, pages 407–420, 1981.

[118] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[119] P.S. Gemmell. An Introduction to Threshold Cryptography. In *CryptoBytes*, RSA Lab., Vol. 2, No. 3, 1997.

[120] R. Gennaro, M. Rabin and T. Rabin. Simplified VSS and Fast-track Multiparty Computations with Applications to Threshold Cryptography. In *17th ACM Symposium on Principles of Distributed Computing*, pages 101–112, 1998.

[121] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. *ECCC*, TR00-022, May 2000.

[122] E.N. Gilbert, F.J. MacWilliams, and N.J.A. Sloane. Codes which detect deception. *Bell Syst. Tech. J.*, Vol. 53, pages 405–424, 1974.

[123] O. Goldreich. Two Remarks Concerning the GMR Signature Scheme. In *Crypto86*, Springer-Verlag Lecture Notes in Computer Science (Vol. 263), pages 104–110, 1987.

[124] O. Goldreich. Towards a Theory of Software Protection and Simulation by Oblivious RAMs. In *19th ACM Symposium on the Theory of Computing*, pages 182–194, 1987.

[125] O. Goldreich. *Foundation of Cryptography – Class Notes*. Preprint, Spring 1989. Superseded by the current book in conjunction with [126].

[126] O. Goldreich. *Lecture Notes on Encryption, Signatures and Cryptographic Protocol*. Extracts from [125]. Available from `http://theory.lcs.mit.edu/~oded/ln89.html` Superseded by the combination of [133], [134], and [132].

[127] O. Goldreich. A Note on Computational Indistinguishability. *Information Processing Letters*, Vol. 34, pages 277–281, May 1990.

[128] O. Goldreich. A Uniform Complexity Treatment of Encryption and Zero-Knowledge. *Journal of Cryptology*, Vol. 6, No. 1, pages 21–53, 1993.

[129] O. Goldreich. *Foundation of Cryptography – Fragments of a Book*. February 1995. Available from `http://theory.lcs.mit.edu/~oded/frag.html` Superseded by the current book in conjunction with [133].

[130] O. Goldreich. Notes on Levin's Theory of Average-Case Complexity. *ECCC*, TR97-058, Dec. 1997.

[131] O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Algorithms and Combinatorics series (Vol. 17), *Springer*, 1999.

[132] O. Goldreich. *Secure Multi-Party Computation*. Unpublished manuscript, 1998. Available from `http://theory.lcs.mit.edu/~oded/gmw.html`.

[133] O. Goldreich. *Encryption Schemes – fragments of a chapter*. December 1999. Available from `http://www.wisdom.weizmann.ac.il/~oded/foc-book.html`

[134] O. Goldreich. *Signature Schemes – fragments of a chapter*. May 2000. Available from `http://www.wisdom.weizmann.ac.il/~oded/foc-book.html`

[135] O. Goldreich. *Foundation of Cryptography – Basic Tools*. Cambridge University Press, 2001.

[136] O. Goldreich. Concurrent Zero-Knowledge With Timing, Revisited. In *34th ACM Symposium on the Theory of Computing*, pages 332–340, 2002.

[137] O. Goldreich, S. Goldwasser, and S. Halevi. Collision-Free Hashing from Lattice Problems. *ECCC*, TR95-042, 1996.

[138] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *Journal of the ACM*, Vol. 33, No. 4, pages 792–807, 1986.

[139] O. Goldreich, S. Goldwasser, and S. Micali. On the Cryptographic Applications of Random Functions. In *Crypto84*, Springer-Verlag Lecture Notes in Computer Science (Vol. 263), pages 276–288, 1985.

[140] O. Goldreich, R. Impagliazzo, L.A. Levin, R. Venkatesan, and D. Zuckerman. Security Preserving Amplification of Hardness. In *31st IEEE Symposium on Foundations of Computer Science*, pages 318–326, 1990.

[141] O. Goldreich and A. Kahan. How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. *Journal of Cryptology*, Vol. 9, No. 2, pages 167–189, 1996. Preliminary versions date to 1988.

[142] O. Goldreich and H. Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, Vol. 25, No. 1, February 1996, pages 169–192.

[143] O. Goldreich and H. Krawczyk. On Sparse Pseudorandom Ensembles. *Random Structures and Algorithms*, Vol. 3, No. 2, (1992), pages 163–174.

[144] O. Goldreich, H. Krawcyzk and M. Luby. On the Existence of Pseudorandom Generators. *SIAM Journal on Computing*, Vol. 22-6, pages 1163–1175, 1993.

[145] O. Goldreich and E. Kushilevitz. A Perfect Zero-Knowledge Proof for a Decision Problem Equivalent to Discrete Logarithm. *Journal of Cryptology*, Vol. 6 (2), pages 97–116, 1993.

[146] O. Goldreich and L.A. Levin. Hard-core Predicates for any One-Way Function. In *21st ACM Symposium on the Theory of Computing*, pages 25–32, 1989.

[147] O. Goldreich and Y. Lindell. Session-Key Generation using Human Passwords. In *Crypto01*, Springer-Verlag Lecture Notes in Computer Science (Vol. 2139), pages 408–432.

[148] O. Goldreich, Y. Lustig and M. Naor. On Chosen Ciphertext Security of Multiple Encryptions. *Cryptology ePrint Archive*, Report 2002/089, 2002.

[149] O. Goldreich and B. Meyer. Computational Indistinguishability – Algorithms vs. Circuits. *Theoretical Computer Science*, Vol. 191, pages 215–218, 1998.

[150] O. Goldreich, S. Micali and A. Wigderson. Proofs that Yield Nothing but their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *Journal of the ACM*, Vol. 38, No. 1, pages 691–729, 1991. Preliminary version in *27th IEEE Symposium on Foundations of Computer Science*, 1986.

[151] O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority. In *19th ACM Symposium on the Theory of Computing,* pages 218–229, 1987.

[152] O. Goldreich, N. Nisan and A. Wigderson. On Yao's XOR-Lemma. *ECCC*, TR95-050, 1995.

[153] O. Goldreich and Y. Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *Journal of Cryptology*, Vol. 7, No. 1, pages 1–32, 1994.

[154] O. Goldreich and E. Petrank. Quantifying Knowledge Complexity. *Computational Complexity*, Vol. 8, pages 50–98, 1999.

[155] O. Goldreich, R. Rubinfeld and M. Sudan. Learning polynomials with queries: the highly noisy case. To appear in *SIAM Journal on Discrete Mathematics*.

[156] O. Goldreich, A. Sahai, and S. Vadhan. Honest-Verifier Statistical Zero-Knowledge equals general Statistical Zero-Knowledge. In *30th ACM Symposium on the Theory of Computing*, pages 399–408, 1998.

[157] O. Goldreich and M. Sudan. Computational Indistinguishability: A Sample Hierarchy. *Journal of Computer and System Science*, Vol. 59, pages 253–269, 1999.

[158] O. Goldreich and S. Vadhan. Comparing Entropies in Statistical Zero-Knowledge with Applications to the Structure of SZK. In *14th IEEE Conference on Computational Complexity*, pages 54–73, 1999.

[159] O. Goldreich and R. Vainish. How to Solve any Protocol Problem – An Efficiency Improvement. In *Crypto87*, Springer Verlag, Lecture Notes in Computer Science (Vol. 293), pages 73–86.

[160] S. Goldwasser and J. Kilian. Primality Testing Using Elliptic Curves. *Journal of the ACM*, Vol. 46, pages 450–472, 1999. Preliminary version in *18th ACM Symposium on the Theory of Computing*, 1986.

[161] S. Goldwasser and L.A. Levin. Fair Computation of General Functions in Presence of Immoral Majority. In *Crypto90*, Springer-Verlag Lecture Notes in Computer Science (Vol. 537), pages 77–93.

[162] S. Goldwasser and Y. Lindell. Secure Computation Without Agreement. In *16th International Symposium on Distributed Computing* (DISC), Springer-Verlag (LNCS 2508), pages 17–32, 2002.

[163] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Science*, Vol. 28, No. 2, pages 270–299, 1984. Preliminary version in *14th ACM Symposium on the Theory of Computing*, 1982.

[164] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, Vol. 18, pages 186–208, 1989. Preliminary version in *17th ACM Symposium on the Theory of Computing*, 1985.

[165] S. Goldwasser, S. Micali, and R.L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM Journal on Computing*, April 1988, pages 281–308.

[166] S. Goldwasser, S. Micali and P. Tong. Why and How to Establish a Private Code in a Public Network. In *23rd IEEE Symposium on Foundations of Computer Science*, 1982, pages 134–144.

[167] S. Goldwasser, S. Micali and A.C. Yao. Strong Signature Schemes. In *15th ACM Symposium on the Theory of Computing*, pages 431–439, 1983.

[168] S. Goldwasser and M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems. *Advances in Computing Research: a research annual*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 73–90, 1989.

[169] S. Haber and S. Micali. Private communication, 1986.

[170] J. Håstad, R. Impagliazzo, L.A. Levin and M. Luby. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing*, Volume 28, Number 4, pages 1364–1396, 1999. Preliminary versions by Impagliazzo et. al. in *21st ACM Symposium on the Theory of Computing* (1989) and Håstad in *22nd ACM Symposium on the Theory of Computing* (1990).

[171] J. Håstad, A. Schrift and A. Shamir. The Discrete Logarithm Modulo a Composite Hides $O(n)$ Bits. *Journal of Computer and System Science*, Vol. 47, pages 376–404, 1993.

[172] M. Hirt and U. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation. *Journal of Cryptology*, Vol. 13, No. 1, pages 31–60, 2000.

[173] R. Impagliazzo and M. Luby. One-Way Functions are Essential for Complexity Based Cryptography. In *30th IEEE Symposium on Foundations of Computer Science*, pages 230-235, 1989.

[174] R. Impagliazzo and M. Naor. Efficient Cryptographic Schemes Provable as Secure as Subset Sum. *Journal of Cryptology*, Vol. 9, 1996, pages 199–216.

[175] R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In *21st ACM Symposium on the Theory of Computing*, pages 44–61, 1989.

[176] R. Impagliazzo and A. Wigderson. P=BPP if E requires exponential circuits: Derandomizing the XOR Lemma. In *29th ACM Symposium on the Theory of Computing*, pages 220–229, 1997.

[177] R. Impagliazzo and D. Zuckerman. How to Recycle Random Bits. In *30th IEEE Symposium on Foundations of Computer Science*, 1989, pages 248–253.

[178] R. Impagliazzo and M. Yung. Direct Zero-Knowledge Computations. In *Crypto87*, Springer-Verlag Lecture Notes in Computer Science (Vol. 293), pages 40–51, 1987.

[179] A. Juels, M. Luby and R. Ostrovsky. Security of Blind Digital Signatures. In *Crypto97*, Springer Lecture Notes in Computer Science (Vol. 1294).

[180] J. Justesen. A class of constructive asymptotically good alegbraic codes. *IEEE Trans. Inform. Theory*, Vol. 18, pages 652–656, 1972.

[181] N. Kahale. Eigenvalues and Expansion of Regular Graphs. *Journal of the ACM*, Vol. 42 (5), pages 1091–1106, 1995.

[182] J. Kahn, M. Saks, and C. Smyth. A Dual Version of Reimer's Inequality and a Proof of Rudich's Conjecture. In *15th IEEE Conference on Computational Complexity*, 2000.

[183] B.S. Kaliski. Elliptic Curves and Cryptography: A Pseudorandom Bit Generator and Other Tools. Ph.D. Thesis, LCS, MIT, 1988.

[184] J. Katz and M. Yung. Complete Characterization of Security Notions for Probabilistic Private-Key Encryption. In *32nd ACM Symposium on the Theory of Computing*, pages 245–254, 2000.

[185] J. Kilian. Basing Cryptography on Oblivious Transfer. In *20th ACM Symposium on the Theory of Computing*, pages 20–31, 1988.

[186] J. Kilian. A Note on Efficient Zero-Knowledge Proofs and Arguments. In *24th ACM Symposium on the Theory of Computing*, pages 723–732, 1992.

[187] J. Kilian and E. Petrank. An Efficient Non-Interactive Zero-Knowledge Proof System for NP with General Assumptions. *Journal of Cryptology*, Vol. 11, pages 1–27, 1998.

[188] J. Kilian and E. Petrank Concurrent and Resettable Zero-Knowledge in Poly-logarithmic Rounds In *33rd ACM Symposium on the Theory of Computing*, pages 560–569, 2001.

[189] H. Krawczyk. LFSR-based Hashing and Authentication. In *Crypto94*, Lecture Notes in Computer Science (Vol. 839), Springer-Verlag, pages 129–139.

[190] H. Krawczyk. New Hash Functions For Message Authentication. In *EuroCrypt95*, Springer-Verlag, Lecture Notes in Computer Science (Vol. 921), pages 301–310.

[191] J.C. Lagarias and A.M. Odlyzko. Solving Low-Density Subset Sum Problems. *Journal of the ACM*, Vol. 32, pages 229–246, 1985.

[192] D. Lapidot and A. Shamir. Fully parallelized multi-prover protocols for NEXP-time. *Journal of Computer and System Science*, Vol. 54 (2), pages 215–220, April 1997.

[193] A. Lempel. Cryptography in Transition. *Computing Surveys*, Dec. 1979.

[194] A.K. Lenstra, H.W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen* 261, pages 515–534, 1982.

[195] L.A. Levin. Average Case Complete Problems. *SIAM Journal on Computing*, Vol. 15, pages 285–286, 1986.

[196] L.A. Levin. One-Way Function and Pseudorandom Generators. *Combinatorica*, Vol. 7, pages 357–363, 1987.

[197] L.A. Levin. Randomness and Non-determinism. *J. Symb. Logic*, Vol. 58(3), pages 1102–1103, 1993.

[198] M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer Verlag, August 1993.

[199] Y. Lindell. A Simpler Construction of CCA2-Secure Public-Key Encryption Under General Assumptions. In preparation, 2002.

[200] Y. Lindell. Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation. In *Crypto01*, Springer Lecture Notes in Computer Science (Vol. 2139), pages 171–189, 2001.

[201] Y. Lindell, A. Lysyanskaya and T. Rabin. On the Composition of Authenticated Byzantine Agreement. In *34th ACM Symposium on the Theory of Computing*, pages 514–523, 2002.

[202] J.H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, Graduate Texts in Mathematics (#88), New York, 1982.

[203] A. Lubotzky, R. Phillips, P. Sarnak, Ramanujan Graphs. *Combinatorica*, Vol. 8, pages 261–277, 1988.

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

[205] M. Luby and C. Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM Journal on Computing*, Vol. 17, 1988, pages 373–386.

[206] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic Methods for Interactive Proof Systems. *Journal of the ACM*, Vol. 39, No. 4, pages 859–868, 1992.

[207] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, San Mateo, CA, 1996.

[208] U. Maurer. Secret Key Agreement by Public Discussion from Common Information. *IEEE Trans. on Inform. Th.* , Vol. 39 (No. 3), pages 733–742, May 1993.

[209] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[210] R.C. Merkle. Secure Communication over Insecure Channels. *CACM*, Vol. 21, No. 4, pages 294–299, 1978.

[211] R.C. Merkle. Protocols for public key cryptosystems. In *Proc. of the 1980 Symposium on Security and Privacy*.

[212] R.C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In *Crypto87*, Springer-Verlag Lecture Notes in Computer Science (Vol. 293), 1987, pages 369-378.

[213] R.C. Merkle. A Certified Digital Signature Scheme. In *Crypto89*, Springer-Verlag Lecture Notes in Computer Science (Vol. 435), pages 218–238.

[214] R.C. Merkle and M.E. Hellman. Hiding Information and Signatures in Trapdoor Knapsacks. *IEEE Trans. Inform. Theory*, Vol. 24, pages 525–530, 1978.

[215] S. Micali, C. Rackoff, and B. Sloan. The Notion of Security for Probabilistic Cryptosystems. *SIAM Journal on Computing*, Vol. 17, pages 412–426, 1988.

[216] S. Micali and P. Rogaway. Secure Computation. In *Crypto91*, Springer-Verlag Lecture Notes in Computer Science (Vol. 576), pages 392–404.

[217] D. Micciancio. Oblivious Data Structures: Applications to Cryptography. In *29th ACM Symposium on the Theory of Computing*, pages 456–464, 1997.

[218] G.L. Miller. Riemann's Hypothesis and Tests for Primality. *Journal of Computer and System Science*, Vol. 13, pages 300–317, 1976.

[219] R. Motwani and P. Raghavan. *Randomized Algorithms*, Cambridge University Press, 1995.

[220] National Bureau of Standards. *Federal Information Processing Standards*, Publ. 46 (DES 1977).

[221] National Institute for Standards and Technology. Digital Signature Standard (DSS), *Federal Register*, Vol. 56, No. 169, August 1991.

[222] M. Naor. Bit Commitment using Pseudorandom Generators. *Journal of Cryptology*, Vol. 4, pages 151–158, 1991.

[223] M. Naor, R. Ostrovsky, R. Venkatesan and M. Yung. Zero-Knowledge Arguments for NP can be Based on General Assumptions. *Journal of Cryptology*, Vol. 11, pages 87–108, 1998.

[224] M. Naor and O. Reingold. Synthesizers and their Application to the Parallel Construction of Pseudo-Random Functions. In *36th IEEE Symposium on Foundations of Computer Science*, pages 170–181, 1995.

[225] M. Naor and O. Reingold. On the Construction of Pseudo-Random Permutations: Luby-Rackoff Revisited. *Journal of Cryptology*, Vol. 12 (1), pages 29–66, 1999.

[226] M. Naor and O. Reingold. From Unpredictability to Indistinguishability: A Simple Construction of Pseudorandom Functions from MACs. In *Crypto98*, Springer Lecture Notes in Computer Science (Vol. 1464), pages 267–282.

[227] M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Application. *21st ACM Symposium on the Theory of Computing*, 1989, pages 33–43.

[228] M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks. In *22nd ACM Symposium on the Theory of Computing*, pages 427-437, 1990.

[229] N. Nisan and D. Zuckerman. Randomness is Linear in Space. *Journal of Computer and System Science*, Vol. 52 (1), pages 43–52, 1996.

[230] A.M. Odlyzko. The future of integer factorization. *CryptoBytes* (The technical newsletter of RSA Laboratories), Vol. 1 (No. 2), pages 5-12, 1995. Available from `http://www.research.att.com/~amo`

[231] A.M. Odlyzko. Discrete logarithms and smooth polynomials. In *Finite Fields: Theory, Applications and Algorithms*, G. L. Mullen and P. Shiue, eds., Amer. Math. Soc., Contemporary Math. Vol. 168, pages 269–278, 1994. Available from `http://www.research.att.com/~amo`

[232] T. Okamoto. On relationships between statistical zero-knowledge proofs. In *28th ACM Symposium on the Theory of Computing*, pages 649–658, 1996.

[233] R. Ostrovsky, R. Venkatesan and M. Yung, "Secure Commitment Against Powerful Adversary: A Security Primitive based on Average Intractability. In *Proceedings of the 9th Symposium on Theoretical Aspects of Computer Science* (*STACS92*), pages 439–448.

[234] R. Ostrovsky and A. Wigderson. One-Way Functions are essential for Non-Trivial Zero-Knowledge. In *2nd Israel Symp. on Theory of Computing and Systems*, IEEE Comp. Soc. Press, pages 3–17, 1993.

[235] R. Ostrovsky and M. Yung. How to Withstand Mobile Virus Attacks. In *10th ACM Symposium on Principles of Distributed Computing*, pages 51–59, 1991.

[236] T.P. Pedersen and B. Pfitzmann. Fail-Stop Signatures. *SIAM Journal on Computing*, Vol. 26/2, pages 291–330, 1997. Based on several earlier work (see first footnote in the paper).

[237] B. Pfitzmann. *Digital Signature Schemes* (*General Framework and Fail-Stop Signatures*). Springer Lecture Notes in Computer Science (Vol. 1100), 1996.

[238] M. Prabhakaran, A. Rosen and A. Sahai. Concurrent Zero-Knowledge Proofs in Logarithmic Number of Rounds. In *43rd IEEE Symposium on Foundations of Computer Science*, 2002.

[239] V. Pratt. Every Prime has a Succinct Certificate. *SIAM Journal on Computing*, Vol. 4, pages 214–220, 1975.

[240] M.O. Rabin. Probabilistic Algorithm for Testing Primality. *Journal of Number Theory*, Vol. 12, pages 128–138, 1980.

[241] M.O. Rabin. Digitalized Signatures. In *Foundations of Secure Computation* (R.A. DeMillo et. al. eds.), Academic Press, 1977.

[242] M.O. Rabin. Digitalized Signatures and Public Key Functions as Intractable as Factoring. MIT/LCS/TR-212, 1979.

[243] M.O. Rabin. How to Exchange Secrets by Oblivious Transfer. Tech. Memo TR-81, Aiken Computation Laboratory, Harvard U., 1981.

[244] C. Rackoff and D.R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto91*, Springer Verlag, Lecture Notes in Computer Science (Vol.), pages 433–444.

[245] R. Raz. A Parallel Repetition Theorem. *SIAM Journal on Computing*, Vol. 27 (3), pages 763–803, 1998.

[246] R. Richardson and J. Kilian. On the Concurrent Composition of Zero-Knowledge Proofs. In *EuroCrypt99*, Springer LNCS 1592, pages 415–413.

[247] R. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *CACM*, Vol. 21, Feb. 1978, pages 120–126.

[248] P. Rogaway. The Round Complexity of Secure Protocols. MIT Ph.D. Thesis, June 1991. Available from `http://www.cs.ucdavis.edu/~rogaway/papers`.

[249] J. Rompel. One-way Functions are Necessary and Sufficient for Secure Signatures. In *22nd ACM Symposium on the Theory of Computing*, 1990, pages 387–394.

[250] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Achieving Chosen-Ciphertext Security. In *40th IEEE Symposium on Foundations of Computer Science*, pages 543–553, 1999.

[251] A. Sahai. Improved Constructions Achieving Chosen-Ciphertext Security. In preparation, 2001. See [91].

[252] A. Sahai and S. Vadhan. A Complete Promise Problem for Statistical Zero-Knowledge. In *38th IEEE Symposium on Foundations of Computer Science*, pages 448–457, 1997.

[253] C.P. Schnorr and H.H. Horner, Attacking the Chor-Rivest Cryptosystem by Improved Lattice Reduction. In *EuroCrypt95*, Springer-Verlag Lecture Notes in Computer Science (Vol. 921), pages 1–12.

[254] A. Shamir. How to Share a Secret. *CACM*, Vol. 22, Nov. 1979, pages 612–613.

[255] A. Shamir. A Polynomial-Time Algorithm for Breaking the Merkle-Hellman Cryptosystem. In *23rd IEEE Symposium on Foundations of Computer Science*, pages 145–152, 1982.

[256] A. Shamir. IP = PSPACE. *Journal of the ACM*, Vol. 39, No. 4, pages 869–877, 1992.

[257] A. Shamir, R.L. Rivest, and L. Adleman. Mental Poker. MIT/LCS Report TM-125, 1979.

[258] C.E. Shannon. Communication Theory of Secrecy Systems. *Bell Sys. Tech. J.*, Vol. 28, pages 656–715, 1949.

[259] M. Sipser. A Complexity Theoretic Approach to Randomness. In *15th ACM Symposium on the Theory of Computing*, pages 330–335, 1983.

[260] M. Sipser. *Introduction to the Theory of Computation*. PWS Publishing Company, 1997.

[261] R. Solovay and V. Strassen. A Fast Monte-Carlo Test for Primality. *SIAM Journal on Computing*, Vol. 6, pages 84–85, 1977. Addendum in *SIAM Journal on Computing*, Vol. 7, page 118, 1978.

[262] D. Stinson Universal hashing and authentication codes. *Designs, Codes and Cryptography*, Vol. 4, pages 369–380, 1994.

[263] M. Sudan. Decoding of Reed-Solomon Codes beyond the error-correction Bound. *Jour. of Complexity*, Vol. 13 (1), pages 180–193, 1997.

[264] M. Tompa and H. Woll, Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information. In *28th IEEE Symposium on Foundations of Computer Science*, pages 472–482, 1987.

[265] S. Vadhan. A Study of Statistical Zero-Knowledge Proofs. PhD Thesis, Department of Mathematics, MIT, 1999.

[266] S. Vadhan. On Constructing Locally Computable Extractors and Cryptosystems in the Bounded Storage Model. *Cryptology ePrint Archive*, Report 2002/162, 2002.

[267] A. Vardi. Algorithmic Complexity in Coding Theory and the Minimun Distnace Problem. In *29th ACM Symposium on the Theory of Computing*, pages 92–108, 1997.

[268] U.V. Vazirani and V.V. Vazirani. Efficient and Secure Pseudo-Random Number Generation. *25th IEEE Symposium on Foundations of Computer Science*, pages 458–463, 1984.

[269] M. Wegman and L. Carter. New Hash Functions and their Use in Authentication and Set Equality. *Journal of Computer and System Science*, Vol. 22, 1981, pages 265–279.

[270] A. D. Wyner. The Wire-Tap Channel. *Bell System Technical Journal*, Vol. 54 (No. 8), pages 1355–1387, Oct. 1975.

[271] A.C. Yao. Theory and Application of Trapdoor Functions. In *23rd IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

[272] A.C. Yao. How to Generate and Exchange Secrets. In *27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.