Limits on the Provable Consequences of One-way Permutations.

Russell Impagliazzo*
Computer Science Department
University of California, San Diego
La Jolla, CA 92093

Steven Rudich†
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

September 19, 1995

Abstract

We present strong evidence that the implication, "if one-way permutations exist, then secure secret agreement is possible", is not provable by standard techniques. Since both sides of this implication are widely believed true in real life, to show that the implication is false requires a new model. We consider a world where all parties have access to a black box for a randomly selected permutation. Being totally random, this permutation will be strongly one-way in a provable, information-theoretic sense. On the other hand, we show that, if $P = NP$, no protocol for secret agreement is secure in such a setting. We show as a consequence that there is no "black box" reduction from a one-way function to a secret-key agreement protocol. With only one exception that we know of, results reducing one cryprographic task or tool to another have been "black box" reductions. Furthermore, we show the existence of an oracle relative to which the implication is false, i.e., there is a one-way permutation, yet secret-agreement is impossible. Thus, no technique which relativizes can prove that secret exchange can be based on any one-way permutation. Our results present a general method for proving statements of the form, "Cryptographic application X has no black box reduction to complexity assumption Y." Our main theorem immediately yields many corollaries of this form.

1 Introduction.

A typical result in cryptography will be of the form: "with assumption X, we can prove that a secure protocol for task P is possible." Because the standard cryptographic assumptions are, at present, unproved, many results focus on weakening the assumptions needed to imply that a given task is possible. However, some tasks seem to inherently require stronger assumptions than others. As a consequence, we ask a new form of question: which assumptions are too weak to yield a proof that a secure protocol for P is possible?

The task we will study is secure secret agreement, one of the most important cryptographic tasks. A secret agreement protocol is a way for Alice and Bob, having no secret information in common, to

*Research partially supported by NSF grant CCR 88-13632.
†Research partially supported by NSF grant CCR-9119319

agree on a secret string over a public channel. Such a protocol is secure when no eavesdropper Eve listening to the conversation can guess the secret in a feasible amount of time with a non-negligible probability. Secure secret agreement is known to be possible under the assumption that trapdoor functions exist [DH76], [GM84]. However, researchers have been frustrated by unsuccessful attempts to base it on the weaker assumption that one-way permutations exist.

This should be of some concern to cryptographers, in that there are very few conjectured trapdoor functions that have withstood serious crypto-analysis ([RSA78] and variants being the exceptions) and there are few secret agreement methods not based on trapdoor functions (basically all variants on [DH76]). Since all current methods are based on number theory, it is possible that a new advance in this subject could eliminate all current methods of secret agreement at once. (For another potential threat, see [Shor84].) Already, improved algorithms have forced key sizes in secret agreement protocols beyond what was believed unbreakable at the time RSA was introduced ([DDLM93]).

We provide strong evidence that it will be difficult to prove that secure secret agreement is possible assuming only that a one-way permutation exists. We model the existence of a one-way permutation by allowing all parties access to a randomly chosen permutation oracle. A random permutation oracle is provably one-way in the strongest possible sense. We show that any secret agreement protocol can be broken by an adversary making only a polynomial number of oracle queries. Furthermore, the adversary's strategy is in the polynomial hierarchy, so any proof that secure secret agreement is possible using such an oracle would simultaneously prove $P \neq NP$.

Every proof to date that a cryptographic tool suffices to perform a cryptographic task gives a reduction between the assumption and the task of the following form. The proof gives a polynomial-time transformation $T(f)$, so that if f is a concrete implementation of the tool, $T(f)$ is an implementation of a protocol for the task. For example, if in a reduction from a one-way permutation to a secret agreement protocol, given any circuit f supposedly computing a one-way function, $T(f)$ would produce a pair of circuits $Alice(f), Bob(f)$ for the sender and receiver in a secret agreement protocol. Then the proof would give a reverse polynomial-time transformation, where given any implementation E of an adversary strategy to break $T(f)$, it would produce an adversary strategy $Adv(E)$ which breaks f with a polynomially related success probability. In our example, E would be some function which given the conversation between $Alice(f)$ and $Bob(f)$ predicts the secret they agree on with some probability ϵ, and $Adv(E)$ would be a strategy to invert f with some polynomially related probability ϵ'. (Hence, if the protocol $T(f)$ is insecure, so is the tool f.) We call such a reduction "black box" if all component functions of $T(f)$ are of the form M^f for some polynomial-time oracle machine M, and if $Adv(E)$ is of the form $N^{E,f}(\epsilon, ..)$ for some oracle machine with two oracles, which runs in time polynomial in its input size and in $1/\epsilon$. In other words, in a "black box" reduction, the protocol uses f's being polynomial-time computable as a "black box", rather than using the knowledge of exactly *how* f is computed; and similarly for the supposed adversary strategy E. Almost all the known reductions between cryptographic tools and tasks are "black box"; we will detail the one counter-example we know of later.

Our results show that no "black box" reduction exists basing secret agreement on a one-way permutation. If such a reduction exists, replacing f with a random permutation, and E with the algorithm to break $T(f)$ that we give here, one would get a procedure $Adv(E)$ to invert f making only polynomially many queries to f. This is well-known to be impossible. (For completeness, we prove this as Theorem 5.2) Secondly, our results show that to give any proof that a "black box" construction of a secret agreement protocol using a one-way permutation $T(f)$ is secure (not just

A	B
One-way permutations exist	Secret agreement is possible
Signature schemes exist	Oblivious transfer is possible
Pseudo-random generators exist	Trapdoor functions exist
Private-key cryptosystems exist	Voting Schemes exist
Telephone coin flipping is possible	
Bit commitment with strong receiver is possible	
Bit commitment with strong sender is possible	
Collision intractable hash functions exist	

Table 1:

via a "black box" argument) is as hard as proving $P \neq NP$.

Finally, we use our results to construct an oracle relative to which one-way permutations exist, but for which secure secret agreement is impossible. (This oracle O is constructed by starting with an oracle for which P=NP, and adding on a random permutation oracle.) This means that any proof, "black box" or not, that the existence of a one-way function implies that of a secure secret agreement protocol cannot relativize (i.e., hold relative to any oracle). Non-relativizing proofs are few and far between not only in cryptography, but in complexity theory as a whole (although admittedly not as few as there were a few years ago [LFKN90],[Sha90]). Since the technique of examining complexity relative to an oracle was introduced in [BGS75], relativization results have been used to provide evidence for the difficulty of resolving questions in complexity theory [BG81]. (We will later briefly discuss the possibility that a non-relativizing proof basing secure secret agreement on a one-way permutation can be found.) Relativized complexity has not been frequently used in cryptography ([Bra, Bra83] is one exception to this rule); we hope the framework developed here will have wide applicability in separating the strengths of cryptographic assumptions.

We also get many similar non-reducibility results between various cryptographic assumptions as corollaries. Since the definitions of the cryptographic tasks and assumptions mentioned here are lengthy, technical, and often not unique, to describe them formally would require a separate paper. (In fact, papers have been written describing various ways of formalizing some of the terms used here; for others, such papers do not presently exist but are greatly needed.) Therefore, rather than attempt to define these terms here, we will give references to papers introducing these concepts and/or papers clarifying them.

Cryptographic tasks to be discussed here include: coin flipping by telephone ([Blu82]), electronic signatures ([DH76], [RSA78], [GMR84]), private-key cryptography ([GM84, GGM84, LR86, Rac88]), bit-commitment (both the strong commiter version ([GMW87]) and the strong receiver version ([BCC87])), identification ([DH76], [FFS86]), electronic voting ([Ben87]), oblivious transfer ([Blu81, Rab81]), and secret agreement ([DH76, Mer78]). General assumptions which have been used in cryptography include the existence of : one-way permutations ([P74]), pseudo-random generators ([BM84, Yao82]), trap-door permutations ([DH76]), and collision-intractable hash functions. This last is a collection of functions of the form $f_x = \lambda y.f(x, y)$ for some easily computable function f which are strictly length-decreasing and for which no feasible algorithm, given a random x, can find strings z and y of length n with $f_x(z) = f_x(y)$ with non-negligible probability.

Many reductions between the various assumptions and tasks listed above are known. In particular, it is known that the existence of a one-way permutation implies the following: pseudo-random generators exist ([Yao82]), private-key encryption is possible ([GM84, GGM84, LR86]), strong commiter bit commitment is possible ([Yao82, GMW87]), strong receiver bit commitment is possible ([NOVY92]), telephone coin flipping is possible ([Blu82]), and electronic signatures are possible ([NY89]). All of the preceding results relativize. We construct an oracle O relative to which one-way permutations exist, but for which no secret agreement protocol is secure. From relativized versions of these results, it follows that O will also have the property that, relative to O, pseudo-random generators exist, strong commiter bit commitment is possible, etc. Thus, none of the preceding assumptions can imply that secure secret agreement is possible in a way which relativizes.

Furthermore, we can add to this list several statements which are not known to follow from the existence of a one-way permutation, but which O can be proved to satisfy because a truly random permutation is used in O's construction. For example, it is unknown whether a one-way permutation can be used to construct a family of collision-intractable hash functions, but it is easy to see that for a random permutation p, and for $|x| = n/2$ the function which outputs the last $n/2 - 1$ bits of $p(x, y)$ will be such a family of functions. Thus, the existence of collision-intractable hash functions does not suffice to construct a secret agreement protocol via a relativizing proof.

Similarly, if an assumption is sufficient to prove the possibility of secret agreement in a relativizing manner, it itself cannot be proven from the existence of a one-way permutation via a "black box" reduction. Examples of such assumptions include oblivious transfer ([Blu81, Rab81]), voting ([Ben87]), and trap-door functions ([DH76, RSA78, GM84]).

To summarize:

There is an oracle O relative to which all A's hold, but all B's do not. (See Table 1.)

Progress has been made in understanding the finer structure of column A. In particular, Pseudorandom generators, Private-key cryptosystems, Telephone coin flipping, Bit commitment, and Signatures have all been shown black-box equivalent to the existence of one-way functions (by relativizing techniques) [HILL91, IL89, Rom90, Naor89].

Some caution is needed in interpreting these results, since at least one non-relativizing construction in cryptography is known. In [I88] it is shown that the theorem proved in [GMW87], "the existence of a one-way permutation implies the existence of zero-knowledge protocols for all languages in NP", fails with respect to a random permutation. Consequently, the construction of an identification protocol based on any one-way function using zero-knowledge proofs of knowledge ([FFS86]) will *not* be possible with just a black box for a random permutation. Their construction is as follows. Every person chooses a random x, and announces publicly $I = f(x)$ as their ID. To prove you are the person with ID I, you give a zero-knowledge proof of knowledge that you know an x with $f(x) = I$. However, to give a zero-knowledge proof as in GMW and FFS that you know such an x, it is necessary to have an actual circuit that computes f, not just a black box which gives the value of f. In fact, if f is a random permutation, no such zero-knowledge proof will be possible. Thus, the [FFS86] scheme does not relativize (although other mechanisms for identification can be based on the existence of a one-way function, for example, using a signature scheme.) The [FFS86] protocol is exceptional even for those constructions involving zero-knowledge proofs. Most applications of zero-knowledge will in fact relativize, although the literal statement of the [GMW87] theorem does not. In a world with a random permutation oracle, it *is* possible to give a zero-knowledge proof for any property *actually* in NP, as opposed to NP relativized to this oracle. It is only applications which attempt to "bootstrap", proving things concerning the values of the same function used to make the protocol zero-knowledge, which fail to relativize.

The above example is the only non-black box construction in cryptography known to the authors for a result in a general form (as opposed to results involving specific crypto-systems). Other non-relativizing techniques, such as arithmetization ([LFKN90, Sha90]) have not yet produced any such reductions between cryptographic assumptions, but only time will tell whether they can be applied to the structural theory of cryptography. Thus, it is fair to say that the result presented here shows that almost all of the standard techniques in cryptography cannot be used to construct a secret exchange protocol from a one-way permutation. Although we believe it to be a difficult question, we feel that searching for a non-relativizing construction is still a worthwhile goal for researchers, in that the potential benefits to cryptography would merit the work and risk. We hope the results here will at least prune out ideas that will not work, and so point towards ideas that will.

2 Notation and definitions.

We use the notation *poly* to represent any function bounded by a polynomial, i.e., $poly = n^{O}(1)$. We abuse notation slightly by not distinguishing between different such functions, e.g., "$poly * poly = poly$".

We will be discussing relationships between many different random variables. Each random variable will be determined as a function of the "world situation" w. For $Event$ a possible event, i.e., subset of world situations, we use the notation $w \models Event$ to represent that $Event(w)$ is true. If V is a random variable which depends on w, we use V^* to represent a possible value for V, and assume that $Prob[V = V*] > 0$. More generally, if $V_1, ..V_k$ are such random variables, if we use the variables $V_1^*, ..V_k^*$ simultaneously, we implicitly assume that $Prob[V_1 = V_1^*, ..V_k = V_k^*] > 0$. We use the notation $w \models V^*$ to abbreviate $w \models V = V^*$. $V[w]$ denotes the value of V in world situation w; since $V[w]$ is always a fixed possible value of V, we can use $V[w]$ in any situation we would use V^*, e.g., "$x \models V[w]$", which is the same as "$V[x] = V[w]$".

A *secret agreement protocol* is a pair of probabilistic polynomial-time Turing machines called Alice and Bob. Each machine has a set of private tapes: a random-bit tape, an input tape, two work tapes, and an output tape (also called the secret tape). In addition, they have a common communication tape that both can read from and write to. A run of the protocol is as follows: Alice and Bob both start with the same integer l written in unary on their input tapes and independent random strings of size polynomial in l written on their random-bit tapes; Alice and Bob run, communicating via the common tape; Alice and Bob both write an l-length string on their secret tape. If this string is the same, Alice and Bob are said to *agree*. The entire history of the writes to the communication tape is called *the conversation*. $\alpha(l)$ will denote the probability that Alice and Bob agree on a secret of length l.

A probabilistic polynomial-time Turing machine Eve *breaks* a secret agreement protocol if Eve, given only the conversation, can guess Bob's secret tape (and hence the secret if there is one) with probability $\alpha(l)/poly(l)$. Eve *strongly breaks* the protocol if her guessing probability is $\alpha(l)(1-o(1))$. A protocol is *secure* if no Eve can break it. A protocol is *weakly secure* if no Eve can strongly break it. One could imagine far more stringent notions of security. For example, we might require that Eve can't even get one bit of the secret. However, in our scenario, we will be breaking the secret agreement protocols in the strong sense defined above, and thus in any weaker sense.

A *one-way permutation* is a polynomial-time computable function which for each n is a 1-1, onto map from n-bit strings to n-bit strings, and where the following holds: no probabilistic polynomial-time algorithm can invert f on a random n bit string with a greater than $1/poly(n)$ probability of success.

We will abbreviate probabilistic polynomial-time Turing machine with the notation $PPTM$, and use $PPOTM$ to stand for probabilistic polynomial time oracle Turing machine. The *computation of a $PPTM$ on a given input* will be a trace of the entire run of the machine given the input. On a fixed input, for a non-oracle machine, the computation is determined by the random tape. If the machine is an oracle machine, the computation would include all the queries and answers received during the computation. (In this case, each computation would be determined by a random tape, and by a finite set of query-answer pairs.)

2.1 The Pigeonhole principle.

We will use the following form of the *pigeonhole principle*:

Let M be a 0-1 matrix with a $1-\alpha$ proportion of 1s. For every $ab = \alpha$, a $1-a$ portion of the columns have at least a $1-b$ portion of 1s. (It suffices to note that the worst case is when the 0's are concentrated in an a by b rectangle.)

3 Uniform Generation.

3.1 Polynomial-time relations.

A relation, R, is polynomial-time if we can decide xRy in time polynomial in $\|x\| + \|y\|$. In this paper, we will only consider relations where the length of y is polynomially related to the length of x. *Is satisfied by* is an example of such a relation: x *is satisfied by* y iff x is a boolean formula and y is one of its satisfying assignments.

3.2 What is uniform generation?

Let R be the "is satisfied by" relation. We can ask two natural questions:

Existence Given x, does there exist a y such that xRy?
(Does a given formula has a satisfying assignment?)

Counting Given x, how many y exist such that xRy?
(How many satisfying assignments does a given formula have?)

The existence question, satisfiability, is NP-complete. The counting question, thought to be harder than satisfiability, is $\#P$-complete. Jerrum, Valiant, and Vazirani[JVV86] introduced a problem of intermediate complexity.

Uniform generation Given x, pick a y uniformly at random such that xRy.
(Given a formula, find a random satisfying assignment.)

More generally, let R be a polynomial-time relation. Let M be a $PPTM$ with a fixed (as opposed to expected) polynomial running time. We say M *uniformly generates* R if given x, M has at least a 50% chance of outputting a uniformly chosen y such that xRy; otherwise, M outputs "try again". If such a y does not exist, M will only output "try again". Notice that re-running the algorithm when it fails to generate a random y will succeed in generating a random y in expected polynomial time.

3.3 $P = NP$ and uniform generation.

Theorem 3.1 (JVV) *For any polynomial-time relation, there exists a PPTM equipped with a* $\sum_2^P$ *oracle that uniformly generates it.*

Theorem 3.2 $P = NP \Longrightarrow$ *for any polynomial-time relation, there exists a PPTM that uniformly generates it.*

Proof: $P = NP \Rightarrow$ the polynomial-time hierarchy collapses[CKS81] $\Rightarrow$
a polynomial-time machine can simulate a $\sum_2^P$ oracle $\Rightarrow$we can use previous theorem to uniformly generate. ■

Let M be a $PPTM$. There are possibly many different computations of M consistent with a given input and output. (Of course, there may be none.) The following corollary shows that if $P = NP$, we can efficiently pick a random element from the finite set of these computations.

Corollary 3.1 $P = NP \Longrightarrow$ *it is possible to generate a random computation for a given PPTM, M, with given input, I, and given output, O, in expected polynomial time.*

Proof: Checking that the trace of a computation is consistent with M, I, and O is a polynomial-time relation. ■

Corollary 3.2 $P = NP \Longrightarrow$ *given a conversation, C, between two PPTMs M and N, we can uniformly generate a possible computation of M.*

Proof: Checking that C is consistent with a given computation of M is possible in polynomial-time. ■

3.4 An application to cryptography.

Public-key cryptography relies on the assumption that **P$\neq$NP**. The formal version of this fact is that $P = NP$ implies secret agreement is not possible. We can use our results on uniform generation to give a particularly simple proof of the optimal result.

Theorem 3.3 $P = NP \Longrightarrow$ *Eve has an expected polynomial time algorithm to break any given secret agreement protocol in the strongest possible sense: Eve will find the secret with exactly the same probability that Alice and Bob agree on one.*

Proof: Fix a computation and resulting secret for Bob. We will show that the probability that Alice agrees with Bob is the same as the probability that Eve agrees with Bob. By corollary 3.2, Eve can generate a random computation of Alice consistent with the conversation. Alice's particular computation is, by definition, a random computation of Alice consistent with the conversation. Thus, Eve and Alice produce secrets with exactly the same probability distribution. They must, therefore, have exactly the same probability of agreeing with Bob. In other words, from Bob's point of view, Alice and Eve think alike; he will fool Eve with exactly the same probability that he will fool (i.e., disagree with) Alice. ■

4 Random Oracles.

4.1 Random function oracles.

Let r be a random real between 0 and 1, chosen with the uniform distribution; express r in binary notation. A *random oracle* is the set induced from r as follows: $\{x :$ the xth binary digit of r is a 1 $\}$.

With each random oracle R, we can associate a function from n-bit strings to n-bit strings. $f(i)$ is defined by its length(i) binary digits; the jth digit is 1 iff $(2i+1)2^j \in R$. (Every natural is uniquely expressed as an odd times a power of 2.) Notice that as we vary over all possible R, we get all possible length-preserving functions, each one occurring with the same frequency. Furthermore, using R as an oracle, f is polynomial-time computable. Thus, a TM with a random oracle also has at its disposal an easy to compute length-preserving random function. The notions of a random oracle and a random function oracle will be used interchangeably. We use the notation $T^f(x)$ to represent the distribution on outputs of probabilistic oracle machine T on input x using function oracle f.

For completeness, we prove the folklore theorem that random functions are one-way in the strongest possible sense.

4.2 Random oracles and one-way functions.

First, for a given machine, we fix the input and the random-bit tape; we show the machine can only invert a very small fraction of functions as we vary over oracles.

Lemma 4.1 *Let T be an oracle PPTM. Then there is a polynomial poly, so that , for any n and any input x of length n, the probability, over a randomly chosen function oracle f and the random tape for T, that $f(T^f(x)) = x$, is less than $poly(n)/2^n$.*

Proof: Fix any random tape for T. If T never queries f at a y such that $f(y) = x$, then the probability that $f(T^f(x)) = x$ is bounded by $1/2^n$. We argue that T never asks such a y with very high probability. Each time T asks a new query y, the probability that $f(y) = x$ is equal to $1/2^n$; T asks only poly(n) many queries; T asks such a y with probability less than $poly(n)/2^n$. The probability that $f(T(x)) = x$ is therefore bounded by $(poly(n) + 1)/2^n$. ∎

Now, for a given machine, we fix the oracle; we show that the machine has a low expectation of inverting as we vary over inputs and random-bit tapes.

Lemma 4.2 *Let T be an oracle PPTM. There exists a $poly(n)$ such that for every length n, there is a $1 - 1/n^2$ measure of oracles f for which the probability that $f(T^f(x)) = x$ on a random input x of length n, is less than $poly(n)/2^n$.*

Proof: By 4.1, there is a polynomial $poly$ so that the probability over random oracles f, inputs x of length n and random tapes, that $f(T^f(x)) = x$ is at most $poly(n)/2^n$. Let P_f be this probability for a fixed function f; the expectation of P_f over random choices of f is at most $poly(n)/2^n$. Thus, by Markov's inequality, the probability over choices of f that $P_f > n^2 poly(n)/2^n$ is at most $1/n^2$. ∎

We say T with f *inverts better than* $< poly(n), n >$, when the expectation that $f(T(x)) = x$, on a random x of length n, is more than $poly(n)/2^n$.

Lemma 4.3 *For every oracle PPTM T, there exists a polynomial poly such that for almost all oracles f, T with f inverts better than* $< poly(n), n >$ *for only finitely many n.*

Proof: Fix T. By lemma 4.2, there exists a poly such that the measure of f's for which T inverts better than $< poly(n), n >$ is bounded by $1/n^2$. $\sum_{n=0}^{\infty} 1/n^2$ converges; by the Borel-Cantelli lemma, measure one of oracles invert better than $< poly(n), n >$ for only finitely many n. ■

Theorem 4.1 *With probability 1 over random oracles, the function associated with the oracle is one-way in the strongest possible sense: For every oracle PPTM, there exists a poly, such that the machine has expectation no more than* $poly(n)/2^n$ *of inverting the inputs of length n.*

Proof: By lemma 4.3, for every oracle $PPTM$, we can throw out the measure zero of oracles where the machines invert well infinitely often (as above). There are only countably many machines; we have thrown out measure zero of oracles in all. Therefore, the remaining measure one of oracles has the property that for every machine, there exists a *poly*, such that there are only finitely many lengths where the machine can invert more than $poly(n)/2^n$ of the inputs. By adding a large constant to *poly*, we can deal with the finite number of lengths on which the machine is able to invert frequently.

This is the strongest possible sense because a machine that samples f at $poly(n)$ random points has a $poly(n)/2^n$ chance of finding the inverse. Such a machine has expectation equal to $poly(n)/2^n$ of inverting a random inputs of length n. ■

In [Bra83, Bra81], Brassard solves what seems to be the more difficult problem of explicitly constructing a strongly one-way function oracle. (He uses no randomness or counting techniques.)

4.3 Random oracles and uniform generation.

Theorem 4.1 implies that uniform generation is impossible in a random world; it is impossible to uniformly generate an inverse to the function associated with the oracle. Our goal is, assuming $P = NP$in the real world, to break secret agreement in a random world. (In theorem 3.3, we saw how to break it in the real world.) Even though we can't hope for uniform generation in a random world (which would make life very easy), we can prove weak analogues of the uniform generation results, which will be helpful.

The idea is not to generate the computation of an oracle $PPTM$, M, with a *particular* random oracle, but rather, with a *random* random oracle; we want a random computation of the machine over all possible oracles. Let $M^{I,O}$ be the finite set of possible computations of M given input I, output O, using some oracle. (These computations are described by the random-bit tape, and the oracle query-answer pairs used during the computation.) A natural probability distribution to put on $M^{I,O}$ is to weight each computation by the probability that it occurs using a random oracle. We want to be able to pick a random element of the space $M^{I,O}$. Note: This time the distribution on the underlying set is not necessarily uniform. The probability of a computation with q queries being chosen is $2^{-q}/2^{-p}$ as likely as a computation with p queries being chosen.

Theorem 4.2 $P = NP \Longrightarrow$ *there exists a PPTM that picks a random element from the probability space* $M^{I,O}$ *in expected polynomial time.*

Proof: ¿From the oracle $PPTM$ M, we construct a $PPTM$ M', such that a uniformly generated computation of M' given input I and output O, when suitably syntactically modified, yields a random element of the probability space $M^{I,O}$. Intuitively, M' is an oracle machine that makes up its own oracle on the fly.

Without loss of generality, assume the computation of M never makes the same oracle query twice; keep track of queries asked in a table, and use the oracle only when the table does not have the answer. Let $t(n)$ be a polynomial bound on the number of oracle queries M asks given an input of length n. M' starts its computation by writing down $t(n)$ random bits on a separate tape, called the *answer tape*. M' then proceeds as M would, except that when M asks the oracle for a query answer, M' answers the simulated query with the first unused bit from the answer tape. By corollary 3.2, we can generate a random computation m' of M', with input I and output O, in expected polynomial time. To make m' look like a random computation of M, strip away the answer tape, pretending that all answers came from an oracle; call the computation that remains m. The probability associated with an m asking q queries is proportional to 2^{-q}. Hence, m is a random element of $M^{I,O}$. ∎

We can strengthen our result slightly by fixing some finite portion of the oracles we wish to consider. Let E be a finite set of oracle addresses and their contents. An oracle is said to be *consistent* with E if the content-address pairs in E are also in the oracle. We define a space similar to $M^{I,O}$: $M_E^{I,O}$ is a finite set of computations of M given I and O, using oracles consistent with E. Each element in $M_E^{I,O}$ is weighed by the probability of it occurring using a random oracle consistent with E. Once again, we wish to pick a random element of the space.

Theorem 4.3 $P = NP \Longrightarrow$ *there exists a PPTM that picks a random element of the probability space* $M_E^{I,O}$ *in expected polynomial time.*

Proof: Same as the proof of the previous theorem with one important modification: Hardwire the answers to oracle queries in E into the finite state control of M'. When M' asks a query in E, *do not* use a bit from the answer tape. ∎

We can now prove the analogue of corollary 3.2 using **oracle** $PPTM$s Alice and Bob. In the case where oracle Alice and oracle Bob have conversation C, and E is a finite set of queries and answers, we define another similar space: A_E^C is the space of possible computations of oracle Alice consistent with the conversation C, where each computation is weighed by its probability of occurring with a random oracle consistent with E. The next theorem will be very important in the results on secret agreement.

Theorem 4.4 $P = NP \Longrightarrow$ *there exists a PPTM that picks a random element of* A_E^C *in expected polynomial time.*

Proof: From Alice's point of view a conversation is a set of inputs and outputs occurring at certain prescribed times during her computation. No further modification of the above proof technique is required. ∎

5 Random Permutation Oracles.

Random permutation oracles are similar to the random function oracles discussed in the previous section, except that the random functions must be 1-1 onto. A *random permutation oracle* Π is a random length-preserving function from the set of finite strings *onto* itself. Again, the function is chosen from the uniform distribution.

From the point of view of oracle $PPTM$s, there is no difference between the two types of oracles. We will formalize this in the spirit of pseudo-randomness.

A *tester* is an oracle machine which, given n and a length-preserving function oracle outputs either 0 or 1. Let T be a n-tester, which makes at most $t(n)$ oracle calls on input n. Let P_n be the probability that T will output a 0, when given n and a random function from n-bit strings to n-bit strings. Let P_n' be the probability that T will output a 0, when given n and a random permutation from n-bit strings to n-bit strings. Let $D_{T_n} = |P_n - P_n'|$. Thus, D_{T_n} measures how well the tester can distinguish between the two types of oracles.

Theorem 5.1 *Let T be a tester that, on input n, makes $t(n)$ queries and queries no string of length less than $l(n)$. Then $D_{T_n} \leq t^2(n)/2^{l(n)}$*

Proof: Consider T's run when the oracle is a random function. Then the answer to a previously unasked query of length k is a random k-bit number, independent of the answers to previously asked queries. Thus, for each query made the probability that it gets the same answer as a previously made query is at most $t(n)/2^k \leq t(n)/2^{l(n)}$. Summing, we conclude that the probability that two queries received the same answer is less than $t(n)^2/2^{l(n}$. Next we observe that the distribution on possible query answers, given that all query answers are different, is the same for random function oracles and random permutation oracles. Thus, the probability that T will output a 0 given that all query answers are different, is the same for the two types of oracles. It follows that $D_{T_n} \leq t^2(n)/2^n$. ∎

The above theorem will allow us to first prove our results relative to a random oracle, and then extend them to a random permutation oracle. A random permutation is also very hard to invert.

Theorem 5.2 *Measure one of random permutation oracles are one-way in the strongest possible sense: For every oracle $PPTM$, there exists a poly, such that the machine has expectation no more than $poly(n)/2^n$ of inverting the inputs of length n.*

Proof: Follows immediately from Theorems 4.1 and 5.1. ∎

6 Breaking Secret Agreement in a Random World

6.1 Introduction.

We will show that the existence of a very strong one-way permutation is not an assumption likely to yield a proof that secure secret key agreement is possible. By theorem 5.2, we know that a random permutation oracle is one-way in the strongest possible sense. Therefore, we will use the availability of a random permutation oracle to model the existence of an ideal one-way permutation. We will show that any secret agreement protocol using such an oracle can be broken by Eve with a polynomial number of queries to the oracle. Furthermore, if $P = NP$, then this Eve is polynomial-time. Thus, to prove secure secret key agreement is possible using a common random permutation oracle is as hard as it is to prove $P \neq NP$. Another way of viewing this is that the security of the protocol must be based on the intractability of some specific problem in NP, rather than merely the one-wayness of the function. For example, it appears likely that the RSA secret agreement method based on factoring is secure ([RSA78]), and so will still be secure if a random one-way permutation is given as a black box (but not actually used by Alice or Bob.) However, this has no bearing on whether secret agreement can be based on an arbitrary one-way permutation! Thus, it is necessary to make some complexity assumption at least as strong as the non-existence of secure secret agreement in the real world, in order to get an analagous impossibility result in a random oracle setting.

6.2 A normal form for secret agreement.

In our analysis, we will assume that the secret agreement protocol has the following normal form: Communication takes place in $n = poly(l)$ rounds (where l is the security parameter and $poly$ is some fixed polynomial). Each round involves one person speaking and computing. Before each round, the party who is to speak asks the oracle a single query, and then does some computation. If Alice speaks first, the protocol would take the following form: Alice queries the oracle; Alice computes; Alice speaks (i.e. writes on the communication tape); Bob queries the oracle; Bob computes; Bob speaks; Alice queries the oracle; Alice computes; Alice speaks; Bob queries the oracle; ... For technical reasons, we also assume that before writing the final secret on the output tape, both Alice and Bob query the oracle at the secret. This only slightly affects the running time, and does not at all affect the security.

Any protocol can be converted to normal form with only a polynomial blow-up in running time. Let Alice and Bob be an arbitrary secret agreement protocol. Let $\Delta(l)$ be a polynomial bounding the number of queries that either Alice or Bob can make in any run of the protocol with security parameter l. Let $r(l)$ be a polynomial bounding the number of communication rounds in such a run, and $t(l)$ be a polynomial bounding the total time taken for computation in such a run.

Lemma 6.1 *There is a protocol Alice', Bob' in normal form taking total time at most $O(r(l)\Delta(l) + t(l))$ so that the probability that Alice' and Bob' agree on a secret is identical to that for Alice and Bob, and so that for any adversary Eve' that runs in time $e(l)$ on a conversation between Alice' and Bob', there is a function Eve that on input a conversation between Alice and Bob, runs in time $O(e(l) + r(l)\Delta(l))$ and has the same probability of producing Alice and Bob's secret as Eve' has of producing the secret for Alice' and Bob'.*

Proof: Alice' and Bob' simulate Alice and Bob, replacing each round with $2\Delta(l) - 1$ rounds. The first $\Delta(l) - 1$ messages for each party are all some fixed message, such as "Please wait". If the round being simulated is one in which Alice speaks, Alice' uses as many rounds as queries Alice makes to make the same queries as Alice, and then makes arbitrary queries subsequently. Bob' just makes arbitrary queries. Alice' sends as her last message the message Alice would have sent in that round. The protocol simulates rounds where Bob speaks analogously. Since runs of the two protocols are in an obvious correspondence, Alice' and Bob' agree with the same probability as do Alice and Bob. Also, any strategy Eve' to break the new protocol can easily be modified to one which given a conversation between Alice and Bob, pads it with "Please wait" messages as appropriate, and then simulates Eve'. ■

6.3 Notation and definitions.

We wish to investigate a random world where Alice and Bob attempt to agree on an l-bit secret. In other words, we vary over runs of Alice, Bob, and Eve; and over oracles. Formally, a *world situation* is a five-tuple $< l, AliceRand, BobRand, EveRand, R >$. l, the input to Alice, Bob, and Eve, is the length of the secret being agreed upon. $AliceRand$, $BobRand$, and $EveRand$ are random bit tapes for Alice, Bob, and Eve to use during their computations. (The length of the tapes is equal to their respective worst-case running times in a protocol for length l.) R is a random oracle. Let WS_l be the set of all world situations where Alice and Bob attempt to agree on an l-length secret (l is the first entry of the five-tuple). We will also think of WS_l as a probability space with the uniform distribution. A world situation determines a random run of the protocol with a random oracle. With each world situation we can associate the following variables:

C_r, the conversation up to and including round r.

q_r, the query asked in round r.

A_r, the query-answer pairs Alice knows up to and including round r. B_r, the query-answer pairs Bob knows up to and including round r.

Notice that none of the three polynomial time machines involved will be able to access the oracle at any address longer than their run-times. Thus, without any loss of generality, we can think of the oracle as finite. This means that the probability space WS_l is finite. (In particular, any event is measurable.)

6.4 Eve's sample space.

We need to define the probability distributions Eve samples from during her algorithm. They have already been described in section 4.3, Theorem 4.4. We define them again here.

Call a random tape for Alice *consistent* with conversation C_r and oracle R if the run of Alice, determined by the random tape and input from Bob's portion of C_r, outputs Alice's portion of C_r. (What she does after round r does not matter.) Let E be a finite set of query-answer pairs.

Let $AS_E^{C_r}$ be the set of <oracle, random tape for Alice> pairs such that E is in the oracle and the random tape for Alice is consistent with C_r and the oracle. Eve will be sampling from the space $A_E^{C_r}$ of computations of Alice consistent with C_r and the query-answer pairs in E. The distribution on $A_E^{C_r}$ is induced from the uniform distribution on $AS_E^{C_r}$; sample a point in $AS_E^{C_r}$, that point corresponds to a computation of Alice: An <oracle, random tape for Alice> pair corresponds to a <finite portion of the oracle used during the computation, random tape for Alice pair>.

6.5 Eve's algorithm.

We now give an algorithm for Eve to break a secret agreement protocol in a random world. This algorithm runs in polynomial time under the assumption that $P = NP$, and in any case makes only polynomially many queries to the function oracle. σ is a *security parameter* which determines Eve's probability of failure and running time.

Assume, without loss of generality, that before writing the secret on their output tape, Alice and Bob both query the oracle at their secret. This increases the number of queries they make by 1, and does not affect the security of the protocol. Then the following algorithm will have the property, that with high probability, if Alice and Bob agree on a secret, then Eve will output a polynomial-size list of l-bit strings that includes their mutual secret. (See Theorem 6.1 for a formal statement.) This suffices for Eve to break the protocol in the weak sense. We later show (Theorem 6.2) how to modify Eve's algorithm so that, if Alice and Bob agree with high probability, then Eve outputs the secret with high probability, thus breaking the protocol in the strong sense.

For each of n rounds of communication between Alice and Bob, Eve computes $m = \lceil 3(n/\sigma)\ln(2n/\sigma)\rceil$ segments. Each segment has a simulation phase and an update phase. We will describe these phases in segment i for round r.

Without loss of generality, assume Alice speaks in round r. Let $E_{r,i-1}$ be the finite set of query-answer pairs that Eve knows about the oracle so far; $< q, a > \in E_{r,i-1}$ iff prior to round r, segment i, Eve has asked if q is in the oracle ($q \in R$?), and received answer a. Recall that C_r is the conversation that has occurred up to this round.

SIMULATION PHASE:

Using the method described in theorem 4.4, Eve picks a random run of Alice from the space $A^{C_r}_{E_{r,i-1}}$. (If Bob speaks in round r, Eve would instead simulate Bob.) Let $F_{r,i}$ be the set of queries that the simulated run of Alice asks her simulated oracle. (Note that so far in this segment, we have not asked any oracle queries. Recall that when simulating a random Alice, we make up the answers to any oracle queries whose answers are not specified by $E_{r,i-1}$.)

UPDATING PHASE:

Eve asks all the queries in $F_{r,i}$ of the actual oracle R. Thus, $E_{r,i}$ equals $E_{r,i-1}$ union the new query-answer pairs Eve learned by asking $F_{r,i}$ of the oracle. Thus, in each updating phase in a round where Alice talks, Since the simulated Alice asks at most one query per round, Eve asks at most $r \leq n$ queries to R in an update phase in the r'th round.

OUTPUT: After the update phase of segment m of round n, Eve outputs the list of queries in $E_{m,n}$. This is the same as the queries Eve has made, so the size of the list is the same as the number of queries, $O(mn^2) = O(n^3/\sigma \ln(2n/\sigma))$.

Given that Eve uses the algorithm above, we can define the following random variables determined by the world situation: $E_{r,i}$, the query-answer pairs Eve knows up to and including the ith segment of her simulation of round r.

$E_{r,0}$, the query-answer pairs Eve knows before she simulates round r. ($E_{r,0} = E_{r-1,m}$.)

$BPQ_{r,i}$, the query-answer pairs Bob knows and Eve does not, up to and including round r, segment i, i.e., $BPQ_{r,i} = B_r - E_{r,i}$. ($BPQ$ stands for "Bob's private queries".)

$EveRand_{r,i}$, the portion of Eve's random tape used up to segment i of round r.

6.6 The efficacy of Eve's algorithm.

Intersection queries are the queries Alice and Bob ask in common during an execution of their protocol. A particular query *becomes an intersection query in round* r if it is asked in round r by one party and has been asked *previously* by the other party. Remember, we have modified the protocol so that, if the parties agree on a secret, the secret is an intersection query. (In a normal form protocol, as their final action, Alice and Bob both query the oracle at the location addressed by their secret outputs.)

The next theorem will prove that with high probability Eve finds all the intersection queries. Thus, Eve will output a polynomial-length list containing the secret and so break the protocol in the weaker sense. We later (Theorem 6.2) show how to extend the algorithm given above to one that guesses the secret with high probability, assuming that Alice and Bob agree on a secret with high probability.

Theorem 6.1 *The probability, over random world situations, that Eve queries all the intersection queries for Alice and Bob is at least* $1-\sigma$*, i.e.,*
$PROB_{x \in WS_l}[x \models A_n \bigcap B_n \subseteq E_{n,m}] \geq 1-\sigma$.

The intuition behind the proof is to consider the first intersection query Alice and Bob make. Before that query, Alice's and Bob's examined parts of the oracle are independent, so until that point, an eavesdropper has the same information about possible runs of Alice that Bob does, namely, consistency with the conversation. Say that there are no intersection queries up to round r but in round $r+1$ Bob speaks, and his query has a non-negligible chance of being an intersection query. Then this query must have been asked at some previous round by a non-negligible fraction of possible Alices. So by sampling such Alices, Eve can probably learn Bob's query. Eve can now ask the intersection query, and hence her information about possible Alices becomes again as good as that of Bob, and the process iterates. The proof of the theorem basically follows this intuition, but great care is needed to handle many subtleties caused by the fact that Eve does not know for which rounds Alice and Bob intersect, or which of her queries were in fact the intersection queries for Alice and Bob.

Proof: We show the stronger result that Eve probably anticipates (asks) a query before it becomes an intersection query. Eve's algorithm has n rounds. If Eve fails to find all intersection queries, there must be a first round where she fails to anticipate an intersection query that occurs in the next round; there exists a first time r when there is a q with $q \in A_{r+1} \bigcap B_{r+1}$ and $q \notin E_{r,m}$. Assume, without loss of generality, that Alice queries and speaks in round r, and Bob in round $r+1$. Then since q_{r+1} is the only new element of A_r or B_r, and $A_{r+1} = A_r$, the only way for r to be the first such round is to have $q_{r+1} \in A_r - E_{r,m}$. Lemma 6.2, the technical heart of the proof, will show this event has probability no more than σ/n by showing that the complementary event has probability greater than $1-\sigma/n$. Summing the error probability for each round, we get a total error probability bounded by σ. ∎

Lemma 6.2 *Without loss of generality, assume that Alice spoke in round* r*. Then the probability that in round* r*, either*

1. $A_r \bigcap BPQ_{r,0} \neq \emptyset$. *(In a previous round, Eve failed to anticipate an intersection query.)*
2. $q_{r+1} \notin A_r$. *(*q_{r+1} *is not an intersection query in round* $r+1$*)*
   OR

3. $q_{r+1} \in E_{r,m}$ *(Eve finds* q_{r+1}*)*

is greater than $1 - \sigma/n$*.*

Proof: We will prove the stronger result that the lemma holds *even when the probability is conditioned on any possible choice of* $C^*{}_r$, $E^*_{r,0}$, $BobRand^*$, $EveRand^*_{r,0}$ and $BPQ^*_{r,0}$.

An important consequence of fixing these variables is to fix q_{r+1}. $q_{r+1}$is to be asked by Bob in round $r+1$; we have fixed his tape ($BobRand^*$), his input from Alice (C^*_r), and his oracle queries and answers (contained in $E^*_{r,0} \cup BPQ^*_{r,0}$); thus, Bob's computation, up until he queries in round $r+1$, has been fixed. $q_{r+1}$is therefore determined.

Let W be the set of world situations given these variables:

$$W = \{w \in WS_l : (w \models C^*_r, E^*_{r,0}, BobRand^*, BPQ^*_{r,0})\ \}$$

We wish to show that a $1 - \sigma/n$ fraction of W satisfies conditions 1, 2, or 3. Partition W into three sets P_1, P_2, and P_3:

$P_1 = \{w \in W:\ w \not\models (condition\ 1),\ w \not\models (condition\ 2)$, and

$$\exists i\ PROB_{x \in W}[\ x \models (conditions\ 1\ or\ 2) \mid x \models E_{r,i}[w], EveRand_{r,i}[w]] > 1 - \sigma/2n\}$$

(Thus, for $w \in P_1$, we have: $w \not\models (condition\ 1)$ and $w \not\models (condition\ 2)$, but an observer who only knew what Eve knows up to segment i would guess otherwise.)

$P_2 = \{w \in W:\ w \not\models (condition\ 1),\ w \not\models (condition\ 2)$, and

$$\forall i\ PROB_{x \in W}[\ x \models (conditions\ 1\ or\ 2) \mid x \models E_{r,i}[w]] \leq 1 - \sigma/2n\}$$

$P_3 = \{w \in W:\ w \models conditions\ 1\ or\ 2\ \}$

Clearly, $|P_1| + |P_2| + |P_3| \ = \ |W|$.

Lemma 6.3 $Prob_{x \in W}[x \in P_1] \leq \sigma/2n$

Proof:

This lemma follows from a general principle. Consider any situation where an observer estimates the probability of an event being true as she learns more about the world being observed. Even if she makes many observations, the probability of the event ever being $1 - p$ probable given all her previous information, but in fact false, is at most p. This is stated precisely in the following lemma:

Lemma 6.4 *Let* $Y_1, ..Y_i...$ *be any sequence of random variables determined by a finite underlying random variable* x*, and let* $Event$ *be any event for the underlying variable. Let* $1 \geq p \geq 0$*. Let* $EventProbable_i[w]$ *be the event that* $Prob_x[Event[x] | x \models Y_1[w], ..Y_i[w]] \geq 1 - p$*. and let* $EventProbable[w]$ *be the event that* $EventProbable_i[w]$ *for some* i*. Then* $Prob[Event | EventProbable] \geq 1 - p$ *and* $Prob[EventProbable \wedge \neg Event] \leq p$*.*

Proof:

Let $First[w]$ be the least i so that $EventProbable_i[w]$ holds if such an i exists, and let $First[w] = \infty$ otherwise. So $EventProbable[w] \iff First[w] < \infty$.

The event whose probability we are trying to bound can then be written as $\neg Event[x] \wedge First[x] < \infty$. Now, for any $i < \infty$, and for any $Y_1^*, ..Y_i^*$, $Prob[First[w] = i | w \models Y_1^*, ..Y_i^*]$ is either 1 or 0. Thus, for any such i and any $Y_1^*, ..Y_i^*$ which are consistent with $First = i$, we have $Prob_x[Event[x] | x \models First[x] = i, Y_1^*, ..Y_i^*] = Prob_x[Event[x] | x \models Y_1^*, ..Y_i^*] \geq 1 - p$. Then for any such i, averaging the above over all possible $Y_1^*, ..Y_i^*$, we have $Prob_x[Event[x] | First[x] = i] \geq 1 - p$. Averaging this over all $i < \infty$, we have $Prob_x[Event[x] | First[x] < \infty] \geq 1 - p$, which proves the first half of the lemma. Then $Prob_x[\neg Event[x] \wedge First[x] < \infty] \leq Prob_x[\neg Event[x] | First[x] < \infty] \leq p$ which is the second half. ∎

Lemma 6.3 then follows from Lemma 6.4 by letting $Event$ be satisfying conditions 1 or 2, $p = \sigma/2n$ and letting Y_i be $E_{r,i}$, $EveRand_{r,i}$, and observing that knowing Y_i determines Eve's computation up to segment i, and hence determines $Y_{i'}$ for all $i' < i$. ∎

Thus, P_1 is small and can be ignored. By definition, all of P_3 satisfies conditions 1 or 2, and hence can only help us. What remains to show is that for most world situations in P_2, Eve finds q_{r+1}.

We give an overview of our strategy: First, we argue that when Eve (in P_2)samples from her space, and happens to produce an Alice whose queries do not intersect $BPQ_{r,i}$, she has a pretty good chance of finding q_{r+1}. Second, we note that when Eve samples an Alice who does intersect $BPQ_{r,i}$, Eve will learn a new query in $BPQ_{r,i}$ in her update phase. (This can only happen size of $BPQ_{r,i}$ times.) Therefore, there will be many segments where Eve has a pretty good chance of finding q_{r+1}.

By definition,

Fact 1

$$\forall w \in P_2 \, \forall i,$$
$$PROB_{x \in W}[\, x \models (conditions\ 1\ or\ 2) \mid x \models E_{r,i}[w], EveRand_{r,i}[w]] \leq 1 - \sigma/2n$$

Equivalently, we can describe the probability of the complementary event:

Fact 2

$$\forall w \in P_2 \, \forall i,$$
$$PROB_{x \in W}[A_r[x] \bigcap BPQ_{r,0}[x] = \emptyset \ and\ q_{r+1} \in A_r[x] \mid x \models E_{r,i}[w], EveRand_{r,i}[w]\,] > \sigma/2n$$

Recall $x, w \in W$. Therefore, $BPQ_{r,0}[x] = BPQ^*_{r,0} = BPQ_{r,0}[w]$. Furthermore, $BPQ^w_{r,i} = BPQ_{r,0}[w] - E_{r,i}[w]$, so $BPQ^w_{r,i} \subseteq BPQ_{r,0}[x]$. From this and 2 we have:

Fact 3

$$\forall w \in P_2 \, \forall i,$$
$$PROB_{x \in W}[A_r[x] \bigcap BPQ_{r,i}[w] = \emptyset \ and\ q_{r+1} \in A_r[x] \mid x \models E_{r,i}[w], EveRand_{r,i}\,] > \sigma/2n$$

Which implies the weaker statement:

Fact 4

$$\forall w \in P_2 \, \forall i, \quad PROB_{x\in W}[q_{r+1} \in A_r[x] \mid x \models E_{r,i}[w], EveRand_{r,i}[w] \text{ and } A_r[x] \bigcap BPQ_{r,i}[w] = \emptyset\,] > \sigma/2n$$

The next lemma is very important. Intuitively, it shows that, conditioned on Alice and Bob having made no private intersection queries prior to r, Eve has the same information about possible runs of Alice in the first r rounds as does Bob. For e an element of $A^{C_r^*}_{E_{r,i}}$, let e_r denote all the oracle queries the simulated Alice in e asks up to and including round r.

Lemma 6.5 *Let $E^*_{r,i}$ and $EveRand^*_{r,i}$ be given, and let $BPQ^*_{r,i} = BPQ^*_{r,0} - E^*_{r,i}$, Then*

$$PROB_{x\in W}[q_{r+1} \in A_r[x] \mid (x \models E^*_{r,i}\, EveRand^*_{r,i},) \text{ and } A_r[x] \bigcap BPQ^*_{r,i} = \emptyset\,] =$$
$$PROB_{e\in A^{C_r^*}_{E^*_{r,i}}}[q_{r+1} \in e_r \mid e_r \bigcap BPQ^*_{r,i} = \emptyset]$$

Proof: By definition of W,

$$PROB_{x\in W}[q_{r+1} \in A_r[x] \mid (x \models E^*_{r,i}\,, EveRand^*_{r,i}) \text{ and } A_r[x] \bigcap BPQ^*_{r,i} = \emptyset\,] =$$
$$PROB_{x\in WS_l}[q_{r+1} \in A_r[x] \mid$$
$$(x \models E^*_{r,i}, EveRand^*_{r,i}\, C^*_r,\ BobRand^*,\ BPQ^*_{r,0},) \text{ and } \ A_r[x] \bigcap BPQ^*_{r,i} = \emptyset\,]$$

We will consider the two relevant probability spaces as sets with the uniform distribution:

$S_1 = \{x \in WS_l :$

$$(x \models E^*_{r,i},\ E^*_{r,0},\ C^*_r,\ BobRand^*,\ BPQ^*_{r,0},\ BPQ^*_{r,i}) \text{ and } \ A_r[x] \cap BPQ^*_{r,i} = \emptyset\ \}$$

$S_2 = \{e \in A^{C_r^*}_{E^*_{r,i}} : e_r \cap BPQ^*_{r,i} = \emptyset\ \}$

For $x \in S_1$ chosen at random, any possible pair $AliceRand[x], R[x]$ is equally likeley, since to extend it to a world situation in S_1 we must pick $BobRand = BobRand^*$ and $EveRand$ to be any extension of $EveRand^*_{r,i}$. Such a pair $AliceRand^*, R^*$ will be possible if and only if R contains $E^*_{r,i}$ and $BPQ^*_{r,i}$, Alice given the pair is consistent with C^*_r, and she queries no element of $BPQ^*_{r,i}$ in the first r rounds. On the other hand, a pair $AliceRand^*, R^*$ is in S_2 if and only if R contains $E^*_{r,i}$, Alice given the pair is consistent with C^*_r, and she queries no element of $BPQ^*_{r,i}$ in the first r rounds. Thus, for each possible pair $AliceRand^*, R$ for S_1, we can associate $2^{|}BPQ_{r,i}|$ such pairs $AliceRand^*, R'$ in S_2 obtained by varying all the oracle answers in $BPQ^*_{r,i}$ arbitrarily. The run of Alice for the first r rounds in the associated pairs will be the same as that in the original, since Alice never asks any question in $BPQ^*_{r,i}$ in the original, and these are the only locations containing a possible difference. Thus, in all the associated pairs, Alice will be consistent with C^*_r and not query an element of $BPQ^*_{r,i}$, and so each associated pair will indeed be in S_2. Conversely, each pair $AliceRand^*, R'$ in S_2 is associated with the unique pair $AliceRand^* R$ where R is obtained from R' by changing the answers to queries asked in $BPQ^*_{r,i}$ to be the answers from $BPQ^*_{r,i}$. Since a unifomly chosen element of S_1 thus is associated with a uniformly distributed element of S_2, and the runs of

Alice are the same for the associated distribution as the original, $A_r[x]$ for $x \in S_1$ chosen uniformly has the same distribution as e_r for $e \in S_2$ chosen uniformly.

This proves Lemma 6.5 . $\blacksquare$

From Fact 4, and Lemma 6.5 using $E^*_{r,i} = E_{r,i}[w]$, $EveRand^*_{r,i} = EveRand_{r,i}[w]$, we conclude:

Fact 5

$$\forall w \in P_2 \, \forall i, \quad PROB_{e \in A^{C^*_r}_{E_{r,i}[w]}}[q_{r+1} \in e_r \mid e_r \bigcap BPQ_{r,i}[w] = \emptyset] > \sigma/2n \tag{1}$$

Claim 1 *: In every world situation, for each round r, there are at least $m-n$ segments where $F_{r,i} \bigcap BPQ_{r,i} = \emptyset$.*

Proof: At the beginning of round r, Bob's private query set has no more than $r \leq n$ elements (there is only one query per round). After each segment, the cardinality of Bob's private query set can only decrease. In a segment where $F_{r,i} \bigcap BPQ_{r,i} \neq \emptyset$, Eve will discover a new query-answer pair in Bob's private query set during the update phase; thus, Eve will decrease Bob's private query set by at least one. Hence, this can happen at most n times. The other $m-n$ segments must therefore satisfy $F_{r,i} \bigcap BPQ_{r,i} = \emptyset$. $\blacksquare$

In each segment i, Eve picks a random $e \in A^{C^*_r}_{E_{r,i}[w]}$, and makes all queries in e_r. We can think of a random $e \in A^{C^*_r}_{E_{r,i}[w]}$ as being generated as follows: Let b_i be the probability that, for a random $e \in A^{C^*_r}_{E_{r,i}[w]}$, e_r has no intersection with $BPQ_{r,i}$. Flip a coin with bias b_i for heads. If heads, then pick a random $e \in A^{C^*_r}_{E_{r,i}[w]}$ given that $e_r \bigcap BPQ_{r,i} = \emptyset$. If tails, pick a random $e \in A^{C^*_r}_{E_{r,i}[w]}$ given that $e_r \bigcap BPQ_{r,i} \neq \emptyset$.

The above claim tells us that there are at least $m-n$ segments where $F_{r,i} \bigcap BPQ_{r,i} = \emptyset$. Therefore, in the generation of each world situation there are $m-n$ segments where the coin comes up heads. In each of these segments, we pick a random $e \in A^{C^*_r}_{E_{r,i}[w]}$ conditioned on $e_{r,i} \bigcap BPQ_{r,i} = \emptyset$. By inequality 5, we conclude that in each $w \in P_2$, there are $m-n$ segments where we have a greater than $\sigma/2n$ chance of finding q_{r+1}. The proportion of P_2 where we fail to find $q_{r+1}$is therefore bounded by $(1-\sigma/2n)^{m-n}$.

Using the fact that $(1-1/x)^x < 1/e$, and that $m-n > 2(n/\sigma)\ln(2n/S_l)$, we can estimate this probability by:

$$(1 - \frac{S_l}{2n})^{2\frac{n}{S_l}\ln\frac{2n}{S_l}}$$

$$= \left((1 - \frac{S_l}{2n})^{\frac{2n}{S_l}}\right)^{\ln\frac{2n}{S_l}}$$

$$< (\frac{1}{e})^{\ln\frac{2n}{S_l}}$$

$$= \frac{\sigma}{2n}$$

We conclude that a $1 - \sigma/2n$ portion of P_2 satisfies condition 3. Recall, P_1 accounts for no more than a $\sigma/2n$ portion of W (lemma 6.3), and all of P_3 satisfies conditions 1 or 2. Therefore, at least a $1 - \sigma$ portion of W satisfies conditions $1, 2$, or 3. This is the desired result for Lemma 6.2. ■

6.7 Finding the secret

In the last subsection, we show how Eve can find a polynomial-sized list with the secret on it somewhere. This is sufficient to guess the secret with at least a polynomial probability. However, in this subsection we show how to extend Eve to actually find the secret with almost the same probability that Alice and Bob agree on one.

Eve will follow the same strategy as in the last section; only her output will change from a list to a single string. Assume that Alice speaks in round $n-1$ of the protocol. In Eve's final round, her simulation of Alice for the entire $n-1$ rounds of the protocol, for each segment Eve records the simulated Alice's last query to the oracle. (Recall that the last query to the oracle for each party is written on the secret tape, and there is no communication in the final round of the protocol.) Of the final queries Eve has recorded, she picks $i \in \{1, ..m\}$ uniformly and outputs the query from the i'th segment.

Theorem 6.2 *Suppose that Alice and Bob agree on a secret with probability at least* $1 - \alpha$. *Then for Eve as above, Eve outputs Bob's secret tape with probability at least* $1 - \alpha - 2\sigma$.

Proof: Without loss of generality, assume Alice asks her final query (the secret) in round $n-1$; then Bob asks his final query (the secret) in round n. We can also assume that the secret has a different form from pervious oracle queries; e.g., the secret is odd, and all previous queries are even, so possible secrets are only asked in the last rounds. Then q_{n-1} is Alice's secret, and q_n is Bob's secret. We are assuming:

$$PROB_{w \in WS_l}[\text{Alice and Bob agree on a secret }] = Prob_{w \in WS_l}[q_{n-1} = q_n] =\geq 1 - \alpha_l$$

Also we know from Theorem 6.1 that with probability $1 - \sigma$, no intersection queries were missed, so Alice could not make an unanticipated intersection query in her final round with any reasonable probability i.e.,

$$PROB_{w \in WS_l}[A_{n-1}[w] \bigcap BPQ_{n-1,i}[w] \neq \emptyset] \leq \sigma$$

Let fixed values $C^*, BobRand^*, EveRand^*_{n-1,i}, E^*_{n-1,i}$, and $BPQ^*_{n-1,0}$ be given, and let W be the set of world situations consistent with these values. These determine q_n, Bob's secret. In segment i, Eve samples a random $e \in A^{C^*}_{E_{n-1,i}}$ and finds the secret in that segment if and only if $q_n \in e_r$. (since the secret has a distinguishing form). By Lemma 6.5

$$PROB_{e \in A^{C^*}_{E_{n-1,i}}}[q_n \in e_r \vee e_r \bigcap BPQ_{n-1,i} \neq \emptyset]$$

$$\geq PROB_{e \in A^{C^*}_{E_{n-1,i}}}[q_n \in e_r | e_r \bigcap BPQ_{n-1,i} = \emptyset]$$

$=$

$$Prob_{w \in W}[q_n \in A_{n-1} | A_{n-1} \bigcap BPQ_{n-1,i} = \emptyset]$$

$$\geq Prob_{w \in W}[q_n \in A_{n-1} \wedge A_{n-1} \bigcap BPQ_{n-1,i} = \emptyset]$$

.

This last is the conditional probability that Alice and Bob agree on a secret and Eve finds all the intersection queries, given all the random variables listed above. Thus, averaging over all values of the aforementioned variables, we have that the overall probability of Eve either finding an element of $BPQ_{n-1,i}$ or of q_n being the final query of the simulated Alice in segment i is at least the probability that Alice and Bob agree on a secret and Eve finds all the intersection queries. This latter probability is at least $1-\alpha-\sigma$. Since this is true for every segment i, it is true for the randomly chosen segment Eve uses for her output. On the other hand, by Claim 1, Eve only finds an element of $BPQ_{n-1,i}$ in at most n of the m segments. Thus, the probability that Eve finds an element in the random segment used for her output is at most n/m. Thus, the overall probability that the simulated Alice in a random segment i shares Bob's secret is at least $1-\alpha-\sigma-n/m < 1-\alpha-2\sigma$.

■

6.8 Permutation oracles

For technical ease, we proved our results so far using a random oracle rather than a random permutation oracle. However, analogs of the above results for random permutation oracles follow as corollaries.

Let $t = (m+2)n$, which is an upper bound on the total number of queries made by Alice, Bob, and Eve. Consider the strategy for Eve which first makes all of the at most t^2/σ queries to the permutation oracle of size at most $l_min = 2\log t - \log \sigma$, and then performs the breaking protocol described before, substituting for Alice and Bob, the protocol *Alice'*, *Bob'* that simulate queries to strings of size l_min or smaller by looking them up rather than querying the oracle.

The following theorem is the analog of Theorem 6.1 for permutation oracles:

Theorem 6.3 *The probability, over random permutation oracles, random tapes for Alice, random tapes for Bob, and random tapes for Eve, that the above algorithm for Eve finds all intersection queries of Alice and Bob, is at least* $1-2\sigma$

Proof: Assume not. We will construct a tester that queries strings of length at least l_min to distinguish between a random function oracle and a random permutation oracle. The tester simulates the above protocol between Alice' and Bob' and then simulates the above described Eve. It outputs 1 if and only if Eve finds all the intersection queries. The number of queries made is at most t (n each to simulate Alice' and Bob', mn to simulate Eve.)

If the above tester is run using a random permutation oracle, then by assumption, the probability of outputting a 1 is less than $1-2\sigma$. By Theorem 6.2, the probability of outputting a 1 using a random function oracle is at least $1-\sigma$. Thus, for the above tester T, $D_T > \sigma = t^2/2^{l_min}$ a contradiction to Theorem 5.1. ■

Likewise,

Theorem 6.4 *Given any secret agreement protocol and a random permutation oracle, the probability that the above Eve outputs Bob's secret is at least* $1 - \alpha - 4\sigma$.

7 Results relative to a fixed oracle

We have shown that if $P = NP$, for any fixed protocol, the protocol is insecure for most permutation oracles. In this section, we reverse the order and show that for almost all permutation oracles, no protocol is secure. We also show that, whether or not $P = NP$, this implies the existence of a fixed oracle relative to which cryptographic one-way permutations exist, but no secret exchange protocol is secure. This shows that no relativizing technique can prove the implication, "If one-way permutations exist, then there is a secure secret exchange protocol".

Theorem 7.1 *Assume* $P = NP$. *Then for measure 1 of random permutation oracles, any polynomial-time secret key agreement protocol where the parties agree on a secret with any* $1/poly$ *probability can be broken with polynomial probabilty in polynomial time.*

Proof: First, we argue that for every secret agreement protocol, there are only measure zero of oracles where it cannot be broken. Fix a protocol and a polynomial p. Consider the oracle algorithm Eve as in Theorem 6.3, where we choose $\sigma = 1/(2l^2 p(l))$. Under the $P = NP$ assumption, this is a polynomial-time breaking strategy relative to any fixed oracle. Theorem 6.3 tells us that, with overall probability at least $1 - \sigma$, either Eve outputs a polynomial size list containing the secret or Alice and Bob do not agree on a secret. By the pigeonhole principle, then, with probability at least $1 - 1/l^2$ over choice of the permutation oracle P, this last event occurs with conditional probability at least $1 - l^2\sigma = 1 - 1/(2p(l))$. Relative to such an oracle, either Alice and Bob agree on a secret with probability less than $1/p(l)$, or Eve has the secret on her list with probability at least $1 - 1/(2p(l)) - (1 - 1/p(l)) = 1/(2p(l)$. Thus, the probability over oracles that Alice and Bob agree with probability $1/p(l)$ and Eve does not have Bob's secret on her list with probability at least $1/2p(l)$ is at most $1/l^2$. Call such an oracle *bad for length l.*

Since $\sum_{l=0}^{\infty} 1/l^2$ converges; by the Borel-Cantelli lemma, measure one of oracles are bad on only finitely many lengths. Thus, for measure one of the oracles, past some length, either Eve has a $1/2p(l)$ chance of breaking the protocol, or Alice and Bob agree on a secret with less than $1/p(l)$ probability. Thus, there are only measure zero oracles where the protocol can't be broken and succesfully agrees on a secret with $1/p(l)$ probability.

For each of the countably many protocols and polynomials, we throw out the measure zero of oracles where the protocol works and is secure. We have thrown out measure zero in all. Every protocol that reaches agreement a polynomial fraction of the time can be broken with at least polynomial probability relative to the measure one of remaining oracles. ■

Corollary 7.1 *There exists an oracle relative to which a strongly one-way permutation exists, but secure secret agreement is impossible.*

Proof: Consider any oracle world where $P = NP$. Add a random permutation oracle to this world. Because all the techniques in our theorem relativize, we can conclude that secure secret agreement is not possible in the resulting world.

Construct an example of such an oracle as follows: The even numbers form an oracle for PSPACE (a PSPACE-complete problem), the odd numbers form a random permutation oracle. $P = NP$ relative to a PSPACE-complete oracle. We know the random permutation is one-way in the strongest possible sense. ■

The only other relativized result that we know in cryptography is Brassard[Bra83, Bra]. He explicitly constructs an oracle where secret agreement is possible.

8 Related Work and Open Problems

Although we, following the tradition of complexity-theoretic cryptography, define security as that versus any polynomial adversary, in practice, security versus an adversary whose time is a large polynomial might be almost as good. Merkle[Mer78] has suggested a secret-agreement protocol, based on a random function, the breaking of which if the function were random would require an eavesdropper to take time quadratic in the time taken by the participants. (Here, time is measured as the number of calls to a black box for the function.) There is no complexity-theoretic characterization of the functions for which Merkle's protocol is quadratically secure. One important problem is to show that Merkle's protocol is actually quadratically secure for some specific function.

Another question is to provide tighter upper and lower bounds for secret agreement in Merkle's model. We showed that for a protocol in normal form, an eavesdropper can always break the protocol with $O(n^3 \log n)$ queries; however, to put the protocol into normal form may square n, so our eavesdropper is actually taking time $O(n^6 \log n)$. This number is again squared in our argument if the function is guaranteed to be a permutation. This leaves open Merkle's question of whether his scheme is optimal.

Another general question brought up by this research is whether similar statements can be proved for other cryptographic applications. We have previously given a list of applications at least as strong as secret key agreement; that these are unlikely to be a consequences of the existence of a one-way permutation follows from the result here. Rudich[Rud91] has shown that, from the point of view of black box reducibility, it is not possible to base a k-round secret key agreement protocol on a secret key agreement protocol requiring $k+1$ rounds. Thus, in terms of black box reductions, there is an infinity hierarchy of non-equivalent cryptographic assumptions related to secret agreement. In particular, this shows that secure secret agreement and the existence of trapdoor functions are not likely to be shown equivalent.

Other issues in the structural theory of cryptography remain open. For example, it is open whether a one-way permutation can be based on an arbitrary one-way function. Rudich[Rud88] has shown that if a certain unproven combinatorial conjecture holds then there is no such black box reduction.

The existence of a non-relativizing reduction from one-way functions to secret key agreement remains possible. The discovery of such a reduction would constitute a major advance in cryptography. It should be appreciated that if this were to happen then the method to perform secret key agreement based on any one-way function would have to work in a very unusual way. The protocol would not only call a subroutine for the one-way function, it would also have to use the actual program which makes up that subroutine.

9 Acknowledgements

We are especially grateful to Manuel Blum and Amos Fiat, who asked us the question of whether a one-way permutation suffices for secret agreement, and presented a model in which it might be disproved. Gilles Brassard asked us about the power of blobs. We would also like to thank Noam Nisan, Charlie Rackoff, and Umesh Vazirani.

References

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=NP question. SIAM J. Comp., 4 (1975) pp. 431-442.

[BG81] C. H. Bennett and J. Gill. Relative to a random oracle A, $P^A \neq NP^A \neq Co-NP^A$ with probability 1. SIAM J. Comp. 10 (1981)

[BCC87] G. Brassard, D. Chaum, and C. Crepeau. Minimum disclosure proofs of knowledge. Technical Report PM-R8710, Centre for Mathematics and Computer Science, Amsterdam, The Netherlands, 1987.

[Ben87] J. Cohen Benaloh. *Verifiable Secret-Ballot Elections.* PhD thesis, Yale University, Sept 1987. YALEU/DCS/TR-561.

[Blu81] M. Blum. Three applications of the oblivious transfer: Part i: Coin flipping by telephone; part ii: How to exchange secrets; part iii: How to send certified electronic mail. Department of EECS, University of California, Berkeley, CA, 1981.

[Blu82] M. Blum. Coin flipping by telephone: A protocol for solving impossible problems. In *Proceedings of the 24th IEEE Computer Conference (CompCon)*, pages 133–137, 1982. reprinted in *SIGACT News*, vol. 15, no. 1, 1983, pp. 23–27.

[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. SIAM J. Comp. 13 (1984) pp. 850–864

[Bra] G. Brassard. An optimally secure relativized cryptosystem. *Advances in Cryptography, a Report on CRYPTO 81*, Technical Report no. 82-04, Department of ECE, University of California, Santa Barbara, CA, 1982, pp. 54–58; reprinted in *SIGACT News* vol. 15, no. 1, 1983, pp. 28–33.

[Bra83] G. Brassard. Relativized cryptography. *IEEE Transactions on Information Theory*, IT-19:877–894, 1983.

[Bra81] G. Brassard. A time-luck tradeoff in relativized cryptography. *Journal of Computer and System Sciences*, 22:280–311, 1981.

[CKS81] A.K. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *JACM*, 28:114–133, 1981.

[DDLM93] J. Denny, B. Dodson, A. K. Lenstra, M. S. Manasse, On the factroization of RSA-120. In *Proceedings of Advances in Cryptography.* CRYPTO, 1993, pp. 166-74.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, 1976.

[FFS86] U. Feige, A. Fiat and A. Shamir. Zero-knowledge proofs of identity. STOC, 1987.

[GGM84] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *Proceedings of the 25th Annual Foundations of Computer Science*. ACM, 1984.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for proto cols with honest majority. In *Proceedings of the 19th Annual Symposium on Theory of Computing*. ACM, 1987.

[GM84] S. Goldwasser and S. Micali. Probabalistic Encryption. *JCSS*, 28:270–299, 1984.

[GMR84] S. Goldwasser, S. Micali, and R. Rivest. A "paradoxical" solution to the signature problem. In *Proceedings of the 25th Annual Foundations of Computer Science*. ACM, 1984.

[HILL91] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby. Pseudo-random number generation from any one-way function. In preparation.

[I88] R. Impagliazzo Proofs that relativize, and proofs that do not. Unpublished manuscript, 1988.

[IL89] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Foundations of Computer Science*. ACM, 1989.

[IY87] R. Impagliazzo and M. Yung. Direct minimum-knowledge computations. In *Proceedings of Advances in Cryptography*. CRYPTO, 1987.

[JVV86] Mark Jerrum, Leslie Valiant, and Vijay Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[LFKN90] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. In *STOC90*, pages 2–10, 1990.

[LR86] M. Luby and C. Rackoff. How to construct pseudo-random permutations from pseudo-random functions. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, 1986.

[Mer78] R. C. Merkle. Secure communications over insecure channels. *CACM*, 21(4):294–299, April 1978.

[Naor89] Bit commitment using pseudo-randomness. In *Proceedings of Advances in Cryptography*. CRYPTO, 1989.

[NOVY92] M. Naor, R. Ostrovsky, R. Venkatesan, M. Yung, *"Perfect Zero-Knowledge Arguments for NP Can BE Based on Genral Complexity Assumption"* Crypto-92.

[NY89] M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, 1989.

[P74] G. P. Purdy A high security log-in procedure. *CACM*, 17:442–445, 1974.

[Rab81] M. O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard University, 1981.

[Rac88] C. Rackoff. A basic theory of public and private cryptosystems. In *Proceedings of Advances in Cryptography.* CRYPTO, 1988

[Rud88] S. Rudich. Limits on the Provable Consequences of One-way Functions. Ph.D. thesis. Berkeley technical report No. UCB/CSD 88/468 (Dec. 1988).

[Rud91] S. Rudich. The Number of Rounds of Interaction in Secret-key Agreement Protocols. In *Proceedings of Advances in Cryptography.* CRYPTO, 1991

[Rom90] J. Rompel One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, 1990.

[RSA78] R. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signature and Public Key Cryptosystems*, Comm. of ACM, 21 (1978), pp 120-126.

[Sha90] A. Shamir. IP=PSPACE. In *Proc. 22nd ACM Symp. on Theory of Computing*, pages 11–15, 1990.

[Shor84] P. Shor, Algorithms for Quantum Computation: Discrete Logarithms and Factoring In *Proc. 35th Annual Symposium on Foundations of Computer Science*, pages 124-134, IEEE, 1994.

[Yao82] A.C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 80–91. IEEE, 1982.