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

May 28, 1994

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. We show that, if $P = NP$, no protocol for secret agreement is secure in such a setting. Thus, to prove that a secret key agreement protocol which uses a one-way permutation as a black box is secure is as hard as proving $P \neq NP$. We also obtain, as a corollary, that there is 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 framework 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 protocol is possible. 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. A secret agreement protocol is a way for Alice and Bob, having no secret information in common, to agree on a secret string over a public channel. Such a protocol is secure when no polynomial-time Eve listening to the conversation can determine

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

part of the secret. 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.

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 proof that secure secret agreement is possible in a world with a random permutation oracle would simultaneously prove $P \neq NP$. (Formally, $P = NP$ implies there is no secure secret agreement relative to a random permutation oracle.) We conclude that it is as hard to provably base a secure secret agreement protocol on an arbitrary one-way permutation as it is to prove $P \neq NP$. Furthermore, we can use the above result 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 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. 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.

Our result also has some implications for "black box" reductions between various other cryptographic assumptions and tasks. Instead of formalizing the notion of "black box" reducibility to an assumption or task, which would involve going into the specifics of these assumptions and tasks, we will use the phrase "A is black-box reducible to B" as an abbreviation for "If B holds relative to an oracle O, then A also holds relative to O". 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. We hope the reader not familiar with cryptography will still be able to follow the general idea of the following discussion.

Cryptographic tasks to be discussed here include: coin flipping by telephone ([Blu82]), electronic signatures ([DH76] , [GMR84]), private-key cryptography ([GM84, GGM84, LR86, Rac88]), bit-commitment (both the strong committer version ([GMW87]) and the strong receiver version ([BCC87])), identification ([DH76], [FFS86]), electronic voting ([Ben87]), oblivious transfer ([Blu81, Rab81]), and secret agreement itself ([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 two-to-one collision-free functions. This last is a function f which is easily computable, is two-to-one on strings of length n, and where no polynomial-time algorithm, given n, can find strings x and y of length n with $f(x) = f(y)$.

Many reductions between the various assumptions and tasks listed above are known. In partic-

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 free functions exist	

Table 1:

ular, 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 committer bit commitment is possible ([Yao82, GMW87]), telephone coin flipping is possible ([Blu82]), and electronic signatures are possible ([NY]). 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 committer 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 two-to-one collision-free function, but it is easy to see that for a random permutation p, the function which outputs all but the last bit of p will be such a function. [NY] show that the existence of a two-to-one collision-free function suffices to construct a protocol for strong receiver bit commitment. Thus, neither the existence of two-to-one collision-free function nor that of a strong receiver bit commitment protocol suffices 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, 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, Pseudo-random 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 [FFS86] construction of an identification protocol based on any one-way function 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 $f(x)$ 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. 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). Thus, it is fair to say that the result presented here shows that most of the standard techniques in cryptography cannot be used to construct a secret exchange protocol from a one-way permutation.

2 Notation and definitions.

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 the secret with probability $\alpha(l)/poly(l)$. A protocol is *secure* if no Eve can 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 1-1, onto from n-bit strings to n-bit strings, where the inverse permutation is not computable in polynomial-time. In fact, for cryptography we require that no $PPTM$ can expect to invert the function on more than a $1/poly(n)$ fraction of the inputs of length n.

We will abbreviate probabilistic polynomial-time Turing machine with the notation $PPTM$. 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 the computation is determined by the random tape.) If the machine is an

oracle machine, this 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.) We use the notation *poly* to refer to some polynomial function. Thus, we can use the freewheeling arithmetic $poly * poly = poly$.

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≠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 $PPTM$which, given n and a function oracle from n-bit strings to n-bit strings, outputs either 0 or 1. Let T be a tester. 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 *For every tester T, $D_{T_n} < poly(n)/2^n$*

Proof: Assume T makes $q < poly(n)$ queries. In the case of a random function oracle, the answer to a previously unasked query is a random n-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 less than $q/2^n$. Summing, we conclude that the probability that two queries received the same answer is less than $q^2/2^n$. 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; 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 $D_{T_n} < q^2/2^n$. ■

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 Cryptographic Lower Bounds.

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 it is as hard to prove secure secret key agreement is possible using a common random permutation oracle is it is to prove $P \neq NP$. The result will take the form of the contrapositive: $P = NP$ implies that any secret agreement protocol can broken even when a random permutation oracle is available to all parties.

Summarizing the results of this section: We first show that $P = NP$ implies there is no secret agreement protocol that is secure with measure $1/poly$ of random oracles (random function oracles). Theorem 5.1 will be used to extend the result to random permutation oracles. Further strengthening the result by swapping the quantifiers, we show $P = NP$ implies for measure one of oracles there is no secure secret agreement. A corollary of this result is the existence of an oracle relative to which one-way permutations exist, but secure secret agreement is impossible. We also distinguish between two strong senses of breaking a secret agreement protocol.

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; ...

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 $< l, random_{Alice}, random_{Bob}, random_{Eve}, R >$. l, the input to Alice, Bob, and Eve, is the length of the secret being agreed upon. $random_{Alice}$, $random_{Bob}$, and $random_{Eve}$ are random bit tapes for Alice, Bob, and Eve to use during their computations (the random bit tapes are just long enough that they never get used up). 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.

If it is ambiguous which world situation C_r comes from, we write C_R^w to mean the conversation comes from world situation w.

World situation w *satisfies* C_r (written $w \models C_r$) means that the conversation between the machines in w is identical to C_r for the first r rounds. We will use the $\models$ notation with the other world situation variables as well.

Notice that none of the three polynomial time machines involved will be able to access the oracle past some very large address. Thus, without any loss, we can think of the oracle as finite. This means that the probability space WS_l is finite. Similarly any space we will discuss can be considered finite. This technical point will prevent the reader from suspecting any measure-theoretic fallacy.

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 $P = NP$. S_l is a function of the form $1/poly$ (called a security parameter), which determines Eve's probability of failure. The smaller S_l, the longer Eve must run to break the protocol.

For each of n rounds of communication between Alice and Bob, Eve does $m = \lceil 3(n/S_l)\ln(2n/S_l)\rceil$ segments. Each segment has a simulate 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 real oracle queries. Recall that when simulating a random Alice, we make up the answers to the oracle queries.)

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.

The following variables are also associated with any 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. BPQ stands for Bob's private queries. Note the relation: $BPQ_{r,i} = BPQ_{r,0} - E_{r,i}$.

6.6 Intersection queries and the secret.

Intersection queries are the queries Alice and Bob ask in common during an execution of their protocol. A particular query becomes an intersection query, not when it is first asked by one party, but rather when it is later asked by the other party. For conceptual unity, we can assume without loss of generality that the secret is an intersection query; assume that as their final act Alice and Bob query the oracle at the location addressed by the secret.

The next Theorem will prove that with high probability Eve finds all the intersection queries. Thus, Eve will have a polynomial-length list containing the secret; Eve breaks the protocol.

6.7 The efficacy of Eve's algorithm.

Theorem 6.1 *Suppose Alice and Bob attempt to agree on an l-length secret. The probability that Eve finds all the intersection queries is greater than $1 - S_l$. Formally,*
$PROB_{x \in WS_l}[A_n \bigcap B_n \subseteq E_{n,m}] > 1 - S_l$.

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 $q \in A_r \bigcap B_r$ and $q \notin E_{r-1,m}$. To formalize the event that Eve fails for the first time to anticipate an intersection query in the next round, we write it as the conjunct of three events:

- Eve has, in previous rounds, anticipated all intersection queries about to happen. (Thus, Eve knows all intersection queries to date.)
- q_{r+1}, the query asked in the next round, is an intersection query.
  AND
- Eve fails to find q_{r+1}. ($q_{r+1} \notin E_{r,m}$.)

Lemma 6.2, the technical heart of the proof, will show this event has probability no more than S_l/n by showing that the complementary event has probability greater than $1 - S_l/n$. Thus, for each round the probability of failing for the first time to anticipate an intersection query in the next round is less than S_l/n. Summing the error probability for each round, we get a total error probability bounded by S_l. ■

Lemma 6.2 *The probability that in round r, either*

- *In a previous round, Eve failed to anticipate the intersection query about to happen,*
- *q_{r+1} is not an intersection query,*
  OR
- *Eve finds q_{r+1}*

  is greater than $1 - S_l/n$.

Proof: Without loss of generality assume that Alice spoke in round r. We will prove the *stronger* result that *even when conditioned on any consistent choice of* C_r, $E_{r,0}$, $random_{Bob}$, and $BPQ_{r,0}$, the probability that one of the following occurs is greater than $1 - S_l/n$:

1. $A_r \bigcap BPQ_{r,0} \neq \emptyset$. (In the last round, Eve did not know all the intersection queries for round r: Alice asked a query in Bob's private query set making an intersection query that Eve had not anticipated. This is stronger than Eve failing to anticipate an intersection query in a previous round.)
2. $q_{r+1} \notin A_r$. (For $q_{r+1}$to be an intersection query in round r, Alice must have asked it *previously.*)
   OR
3. $q_{r+1} \in E_{r,m}$. (The definition of Eve finding q_{r+1}.)

Fix any consistent C_r^*, $E_{r,0}^*$, $random_{Bob}^*$, and $BPQ_{r,0}^*$. (The * superscript will later serve to disambiguate sets that have been previously defined from similar sets occurring as free variables in expressions.) 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 ($random_{Bob}$), his input (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.

W will be the set of world situations satisfying these conditions:

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

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

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

$$\exists i \ PROB_{x \in W}[\ x \models (conditions\ 1\ or\ 2) \mid x \models E^w_{r,i}] > 1 - S_l/2n\}$$

(Thus, $w \in P_1 \Rightarrow w \not\models (condition\ 1)$ and $w \not\models (condition\ 2)$, but an observer who only knew the $E_{r,i}$ portion of w would guess otherwise.)

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

$$\forall i \ PROB_{x \in W}[\ x \models (conditions\ 1\ or\ 2) \mid x \models E^w_{r,i}] \leq 1 - S_l/2n\}$$

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

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

Claim 1 : $\frac{|P_1|}{|W|} \leq \frac{S_l}{2n}$

Proof:

For each element $x \in W$, we can associate a number $First_x$: The first i such that $PROB_{y \in W}[\ y \models (conditions\ 1\ or\ 2) \mid y \models E^x_{r,i}] > 1 - S_l/2n$; if there is no such i, let $First_x = \infty$.

Partition W as follows: Place all x such that $First_x = \infty$ in the same block. The remaining x's index the other blocks. $Block_x$ consists of all y such that $y \models E^x_{r,First_x}$ and $Random^x_{Eve} = Random^y_{Eve}$. (We have a partition because x, y in the same block implies that $First_x = First_y$; the Eve's in x and y must behave exactly the same until the $First_x$th segment.)

We will show that each block of the partition has a small intersection with P_1. The block where $First_x = \infty$ has no intersection with P_1. Consider one of the remaining blocks, $Block_x$. The proportion of elements of $Block_x$ not in P_1 is given by:

$$PROB_{w \in W}[\ w \models (conditions\ 1\ or\ 2)$$

$$\mid w \models E^x_{r,First_x} \text{ and } Random^x_{Eve} = Random^w_{Eve}]$$

Conditions 1 and 2 are independent of $Random^w_{Eve}$. Therefore, we rewrite our proportion:

$$PROB_{w \in W}[\ w \models (conditions\ 1\ or\ 2) \mid w \models E^x_{r,First_x}]$$

Which, by definition of $First_x$, is greater than $1 - S_l/2n$. Thus, P_1 makes up less than $S_l/2n$ proportion of each block, and hence of the whole space W. ■

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. We now show 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,

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

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

$$\forall w \in P_2 \, \forall i,$$
$$PROB_{x \in W}[A^x_r \bigcap BPQ^x_{r,0} = \emptyset \ and\ q_{r+1} \in A^x_r \mid x \models E^w_{r,i}\,] > S_l/2n$$

Recall $x, w \in W$. $BPQ^x_{r,0} = BPQ^*_{r,0} = BPQ^w_{r,0}$. Furthermore, $BPQ^w_{r,i} = BPQ^w_{r,0} - E^w_{r,i}$, implying $BPQ^w_{r,i} \subseteq BPQ^x_{r,0}$. Substituting we get:

$$\forall w \in P_2 \, \forall i,$$
$$PROB_{x \in W}[A^x_r \bigcap BPQ^w_{r,i} = \emptyset \ and\ q_{r+1} \in A^x_r \mid x \models E^w_{r,i}\,] > S_l/2n$$

Which implies the weaker statement:

$$\forall w \in P_2 \, \forall i,$$
$$PROB_{x \in W}[q_{r+1} \in A^x_r \mid x \models E^w_{r,i} \ and\ A^x_r \bigcap BPQ^w_{r,i} = \emptyset\,] > S_l/2n$$

The next claim is very important. It relates distribution W to the space $A^{C^*_r}_{E^*_{r,i}}$ from which Eve samples. Let e be a random element of $A^{C^*_r}_{E^*_{r,i}}$; e_r will denote all the oracle queries the Alice in e asks up to and including round r.

Claim 2 *: For any fixed $E^*_{r,i}$ and $BPQ^*_{r,i}$,*

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

Proof: By definition of W,

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

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

$S_1 = \{x \in WS_l :$

$$(x \models E^*_{r,i},\, E^*_{r,0},\, C^*_r,\, random^*_{Bob},\, BPQ^*_{r,0},\, BPQ^*_{r,i})\ and\ \ A^x_r \bigcap BPQ^*_{r,i} = \emptyset\ \}$$

$S_2 = \{e \in AS^{C^*_r}_{E^*_{r,i}} : A^e_r \bigcap BPQ^*_{r,i} = \emptyset\ \}$

Partition S_1 by equating $x, x' \in S_1$ iff they are the same on the oracle and $random_{Alice}$. (They are all the same on $random^*_{Bob}$ and may differ on $random_{Eve}$.)Each block has the same size; if a $random_{Eve}$ is possible with one <oracle, $random_{Alice}$ > pair, it is possible with another. (Simply recall that $E_{r,i}$ and C_r are fixed.) Notice that if $q_{r+1} \in A^x_r$ for one element of a block, then the same holds for all elements of the block. A block is described by a $random_{Alice}$ and an oracle containing $BPQ^*_{r,i} \cup E^*_{r,i}$.

Partition S_2 by equating $e, e' \in S_2$ iff they are the same everywhere except the oracle locations in the set $BPQ^*_{r,i}$. Each block of the partition has the same size ($2^{|BPQ^*_{r,i}|}$); recall that $A^e_r \bigcap BPQ^*_{r,i} = \emptyset$. If $q_{r+1} \in A^e_r$ for one element of a block, then the same holds for all elements of the block. A block is described by a $random_{Alice}$ and an oracle containing $E^*_{r,i}$ in addition to "don't care" as the contents of the addresses contained in $BPQ^*_{r,i}$. Thus, there is a 1-1 correspondence between the blocks of $S1$ and the blocks of $S2$.

These facts yield the desired claim. ■

We conclude:

$$\forall w \in P_2\, \forall i, \qquad (1)$$
$$PROB_{e \in A^{C^*_r}_{E^w_{r,i}}} [q_{r+1} \in e_r \mid e_r \bigcap BPQ^w_{r,i} = \emptyset] > S_l/2n$$

Claim 3 : *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 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. By the above remarks, this can only happen n times. The other $m - n$ segments must therefore satisfy $F_{r,i} \bigcap BPQ_{r,i} = \emptyset$. ■

We can think of a random $w \in W$ as being generated, segment by segment, as follows: Conditioned on everything so far, pick random values for the variables in the next segment. In fact, in each segment, we pick a random $e \in A^{C^*_r}_{E^w_{r,i}}$ (recall $random^*_{Bob}$ is fixed).

Furthermore, we can think of a random $e \in A^{C^*_r}_{E^w_{r,i}}$ as being generated as follows: Let b be the probability that a random $e \in A^{C^*_r}_{E^w_{r,i}}$ has no intersection with $BPQ_{r,i}$. Flip a coin of bias b. If heads, then pick a random $e \in A^{C^*_r}_{E^w_{r,i}}$ given that $e \bigcap BPQ_{r,i} = \emptyset$. If tails, pick a random $e \in A^{C^*_r}_{E^w_{r,i}}$ given that $e \bigcap BPQ_{r,i} \neq \emptyset$.

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 $e \in A^{C_r^*}_{E^w_{r,i}}$ conditioned on $e_{r,i} \bigcap BPQ_{r,i} = \emptyset$. By inequality 1, we conclude that in each $w \in P_2$, there are $m - n$ segments where we have a greater than $S_l/2n$ chance of finding q_{r+1}. The proportion of P_2 where we fail to find $q_{r+1}$is therefore bounded by $(1 - S_l/2n)^{m-n}$.

We now calculate, using the fact that $(1 - 1/x)^x < 1/e$. Also $m - n > 2(n/S_l)\ln(2n/S_l)$.

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

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

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

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

We conclude that a $1 - S_l/2n$ portion of P_2 satisfies condition 3. Recall, P_1 accounts for no more than a $S_l/2n$ portion of W, and all of P_3 satisfies conditions 1 or 2. Therefore, at least a $1 - S_l$ portion of W satisfies conditions $1, 2$, or 3. This yields the desired result. ■

Theorem 6.2 *Theorem 6.1 is true relative to a random permutation oracle: Given any secret agreement protocol and a random permutation oracle, the probability that Eve finds all the intersection queries is greater than* $1 - S_l/2$.

Proof: Assume not. We will construct a tester to distinguish between a random function oracle and a random permutation oracle. We start with a protocol where Eve will find all the intersection queries with probability less than $1 - S_l/2$ if a random permutation oracle is used, and probability greater than $1 - S_l$ if a random function oracle is used. A tester can simulate runs of Alice, Bob, and Eve, counting the fraction of times Eve finds all the intersection queries. The essence of the situation is that the tester is flipping a coin with two possible biases: $1 - S_l/2$ and $1 - S_l$; the tester must guess which. If the tester flips the coin $1/{S_l}^2$ times, even a very weak form of the law of large numbers would tell us that Eve can guess the bias of the coin at least 99% of the time. This very strongly contradicts theorem 5.1. ■

Notice the order of the quantifiers in the above result. We picked the protocol between Alice and Bob, then we picked the oracle (since the protocol is bound by definition to work with a random oracle). Then, we showed Eve can break the protocol. We prove a stronger result which reverses the quantifiers. First, we pick a random oracle; then a protocol for Alice and Bob (this time the protocol need not work properly on other oracles). Then, we show that Eve can break the protocol relative to the chosen oracle.

Theorem 6.3 *$P = NP \Longrightarrow$ relative to a random permutation oracle, any secret key agreement scheme can be broken.*

Proof: First, we argue that for every secret agreement protocol, there are only measure zero of oracles where it can't be broken. Fix a protocol. The $P = NP$ assumption allows us to use Eve's algorithm as before. Choose $S_l = 1/l^{2+\epsilon}$. Theorem 6.1 tells us that in $1 - S_l/2$ of world situations we succeed in breaking the protocol. By the pigeon-hole principle, for each length l, there are $1 - \sqrt{S_l/2}$ oracles relative to which there is a $1 - \sqrt{S_l/2}$ chance of Eve breaking the protocol. Call all such oracles good for length l. The probability that a random oracle fails to be good for length l is $\sqrt{S_l/2}$. $\sum_{l=0}^{\infty} \sqrt{S_l/2}$ converges; by the Borel-Cantelli lemma, measure one of oracles are good on all but finitely many lengths. For measure one of the oracles, past some length, Eve has a $1 - \sqrt{S_l/2}$ chance of breaking the protocol. (We can even non-uniformly boost Eve's ability to break protocols for finitely many lengths.) Thus, there are only measure zero oracles where the protocol can't be broken.

For each of the countably many protocols we throw out the measure zero of oracles where the protocol is secure. We have thrown out measure zero in all. Every protocol can be broken relative to the measure one of remaining oracles. ■

Corollary 6.1 *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.

So far, our sense of breaking a secret key agreement consists of finding a polynomial-sized list with the secret on it somewhere. The strongest sense of breaking secret key agreement is clearly to find the secret itself. We show how to extend Eve to actually find the secret. For the same reasons as before, the argument works equally well with both random oracles and random permutation oracles.

Eve's strategy can be extended as follows: Eve's final round will be her simulation of the $n-1$th round of the protocol. In each segment of her final round, Eve records her last query to the oracle. (Recall that the last query to the oracle should be thought of as the secret.) Of the final queries Eve has recorded, she outputs the one which occurs the majority of the time. (If there is no majority, output "failure".)

Theorem 6.4 *Suppose that Alice and Bob agree on a secret with probability at least $1 - \alpha$ over world situations in WS_l. Then, for every $\delta > 0$, there exists an Eve who can guess the secret with probability at least $1 - \alpha(2 + \delta)$ over world situations in WS_l.*

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. The proof hinges on two inequalities. The first is the formal statement of what we are assuming:

$PROB_{w \in WS_l}$[Alice and Bob agree on a secret] $\geq 1 - \alpha_l$

The second is a consequence of Theorem 6.1; if no intersection queries were missed, then Alice could not make an unanticipated intersection query in her final round.

$PROB_{w \in WS_l}[A^w_{n-1} \bigcap BPQ^w_{n-1,i} \neq \emptyset] \leq S_l$

We abbreviate the expression of two events: $FINDSECRET^w_i$ iff in the ith segment of the last round of Eve's simulation in world situation w, Eve's last query is the secret. $ASKBPQ^w_i$ iff in the ith segment of the last round of Eve's simulation in world situation w, Eve asks a query in $BPQ_{n-1,i}$. $NOTASKBPQ^w_i$ is the complementary event.

Fix a segment i. We will lower bound the following quantity in a four step calculation:

$PROB_{w \in WS_l}[FINDSECRET^w_i$ or $ASKBPQ^w_i]$

$\geq PROB_{w \in WS_l}[FINDSECRET^w_i \mid NOTASKBPQ^w_i]$

$= PROB_{w \in WS_l}$[Alice and Bob agree on a secret $\mid A^w_{n-1} \bigcap BPQ^w_{n-1,i} \neq \emptyset]$

(Partition WS_l according to $random_{BOB}$, C_{n-1}, $E_{n-1,i}$, and $BPQ_{n-1,i}$, and apply claim 2 of lemma 6.2 to each block. Alice and Bob will agree on the secret iff $q_{r+1} \in A^w_{n-1}$.)

$\geq PROB_{w \in WS_l}$[Alice and Bob agree on a secret] $- PROB_{w \in WS_l}[A^w_{n-1} \bigcap BPQ^w_{n-1,i} \neq \emptyset]$

$\geq 1 - \alpha_l - S_l$

Thus, for any segment i, $PROB_{w \in WS_l}[FINDSECRET^w_i$ or $ASKBPQ^w_i] \geq 1 - \alpha_l - S_l$. By the pigeonhole principle, there is a $1 - (\alpha_l + S_l)((2 - \epsilon)$ fraction of world situations where a $1/((2 - \epsilon)$ fraction of the segments of the final round satisfy $FINDSECRET^w_i$ or $ASKBPQ^w_i$. By Claim 3 of lemma 6.2,the fraction of segments of any round of any world situation which satisfy $ASKBPQ^w_i$ is smaller than n/m. As l increases, n/m is smaller than any constant. Therefore, there is a $1 - (\alpha_l + S_l)((2 - \epsilon)$ fraction of world situations where the majority of segments of the final round find the secret. By choosing sufficiently small S_l and ϵ, Eve can guess the secret with probability at least $1 - \alpha_l(2 + \delta)$. ■

7 Related Work and Open Problems

In the work presented here, as in much of theoretical cryptography, we do not go into exactly how much time the adversary will take to break the protocol, as long as this time is polynomial. However, in real life, a protocol taking a large degree polynomial time to break may be almost as good as one secure against any polynomial time adversary. Merkle[Mer78] has suggested a protocol, based on any one-way function, the breaking of which 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 one-way function.) We showed that for a protocol in normal form, an eavesdropper can always break the protocol in time $O(n^3 \log n)$; however, to put the protocol into normal form may square n, so our eavesdropper is actually taking time $O(n^6 \log n)$. 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 reduction, there are an infinity of cryptographic assumptions each having different implications. The picture of black box reduction is not complete. For example, it is open whether a one-way permutation can be based on a one-way function. Rudich[Rud88] has shown that if a certain *unproven* combinatorial conjecture holds then there is no such black box reduction.

The possibility of a non black box reduction from one-way functions to secret key agreement remains. 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 program to perform secret key agreement based on any one-way function (e.g., DES) would have to work in a very unusual way. The program would not only call the subroutine for DES, it would also have to read the actual program which makes up that subroutine.

8 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 pseudo-random 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.

[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.

[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.

[NY] 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.

[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.