Merkle Puzzles are Optimal

Boaz Barak* Mohammad Mahmoody-Ghidary†

January 27, 2008

Abstract

We prove that every key exchange protocol in the random oracle model in which the honest users make at most n queries to the oracle can be broken by an adversary making $O(n^2)$ queries to the oracle. This improves on the previous $\tilde{\Omega}(n^6)$ query attack given by Impagliazzo and Rudich (STOC' 89). Our bound is optimal up to a constant factor since Merkle (CACM '78) gave an n query key exchange protocol in this model that cannot be broken by an adversary making $o(n^2)$ queries.

Our result extends to an $O(n^2)$ query attack in the random permutation model, improving on the pervious $\tilde{\Omega}(n^{12})$ attack of Impagliazzo and Rudich. This bound again is optimal up to a constant factor since Merkle's protocol can be adapted to this model as well.

1 Introduction

In the 1970's Diffie, Hellman, and Merkle began to challenge the accepted wisdom that two parties cannot communicate confidentially over an open channel without first exchanging a secret key using some secure means. The first such protocol (at least in the open scientific community) was given by Merkle in 1974 (although only published in 1978 [Mer]). Merkle's protocol (known as "Merkle Puzzles") allows two parties Alice and Bob to agree on a random number k that will not be known to an eavesdropping adversary Eve. It works in a model where all parties involved have access to a black-box (i.e., oracle) hiding a random function $H : \{0,1\}^\ell \to \{0,1\}^\ell$ (this function is meant to model a cryptographic one-way or hash function). The protocol is quite simple and operates as follows:

1. Alice chooses $10n$ random numbers $x_1, \ldots, x_n$ in $[n^2]$ and sends $a_1, \ldots, a_n$ to Bob where $a_i = H(x_i)$ (we assume $n^2 \ll 2^\ell$ and embed $[n^2]$ in $\{0,1\}^\ell$ in some canonical way).

2. Bob chooses $10n$ random numbers $y_1, \ldots, y_n$ in $[n^2]$ and sends $b_1, \ldots, b_n$ to Alice where $b_j = H(x_j)$.

3. With at least 0.9 probability, there will be at least one "collision" between Alice's and Bob's messages: a pair i, j such that $a_i = b_j$. Alice and Bob choose the lexicographically first such pair, and Alice sets $s_a = x_i$ as her secret, and Bob sets $s_b = y_j$ as his secret. (If no collision occurred they will not choose any secret; also note that assuming $2^\ell \gg n^4$, H will be one to one on $[n^2]$ with very high probability $H(x_i) = H(y_j)$ implies $x_i = y_j$.)

To analyze the protocol one shows that the collision is distributed uniformly in $[n^2]$ and deduces that an adversary Eve that makes $o(n^2)$ queries to the oracle will find the secret with $o(1)$ probability.

One problem with Merkle's protocol is that it's analyzed in the random oracle model which does not necessarily capture security when instantiated with a cryptographic one-way or hash function [CGH]. But

[*] Department of Computer Science, Princeton University, boaz@cs.princeton.edu. Supported by NSF grants CNS-0627526 and CCF-0426582, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

[†] Department of Computer Science, Princeton University, mohammad@cs.princeton.edu. Supported by NSF grant CNS-0627526.

the most serious problem is that it only provides a quadratic gap between the running time of the honest parties and the adversary. Fortunately, not too long after Merkle, Diffie and Hellman [DH] and later Rivest, Shamir, and Adleman [RSA] gave constructions for key exchange protocols that are conjectured to have *super-polynomial* (even subexponential) security. But because these and later protocols are based on certain algebraic computational problems, and hence could perhaps be vulnerable to unforseen attacks using this structure, it remained an interesting question to show whether there exist protocols with superpolynomial security that use only a random oracle.[1] The seminal paper of Impagliazzo and Rudich [IR] answered this question negatively, by showing that every key exchange protocol using n queries in the random oracle model can be broken by an adversary asking $O(n^6 \log n)$ queries.[2] Since a random oracle is in particular a one-way function, this implied that there is no construction of a key-exchange protocol that is based on a one-way function and has a proof of super-polynomial security that is of the standard black-box type (i.e., a proof that transforms an adversary breaking the protocol into an inversion algorithm for the one-way function that only uses the adversary and the function as black boxes).

Still, [IR] left open the question of whether there exist protocols in this model with $\omega(n^2)$ security or in fact Merkle's protocol is optimal. One reason to ask this question is that large polynomial security might suffice for some applications. But perhaps more than that it's just interesting to know whether Merkle's simple 2-message protocol is the best that can be done in this model, or it is possible to take advantage of interaction to get better security. In this work we answer the above question, by showing that every protocol in the random oracle model where Alice and Bob make n oracle queries can be broken with high probability by an adversary making $O(n^2)$ queries. That is, we prove the following:

Theorem 1.1. *Let Π be a two-party protocol in the random oracle model such that when executing Π the two parties Alice and Bob make a total of at most n queries, and their outputs are identical with probability at least ρ. Then, there is an adversary Eve making $O((\frac{n}{\delta})^2)$ queries to the oracle whose output agrees with Bob's output with probability at least $\rho - \delta$.*

We also show the same result in the case that the oracle is a random permutation (for this case [IR] gave an $O(n^{12} \log^2 n)$ query attack). We note that although our adversary makes few queries it does *not* run in polynomial time (otherwise this would rule out the existence of key exchange protocols with super-polynomial security). As is the case in [IR] and other black-box separation results, our adversary can be implemented efficiently in a relativized world where $\mathbf{P} = \mathbf{NP}$, meaning that these results rule out one-way function and one-way permutation based key-exchange protocols with a relativizing proof showing $\omega(n^2)$ security.

2 Our techniques

It is instructive to compare our techniques with the techniques of the previous work by Impagliazzo and Rudich [IR]. Our attack is similar (though not identical) to the attack of [IR], but our analysis is quite different (and arguably simpler). Consider a protocol that consists of n rounds of interaction, where each party makes exactly one oracle query before sending its message. [IR] called such protocols "normal-form protocols" and gave an $\tilde{O}(n^3)$ attack against them (their final result was obtained by transforming every protocol into a normal-form protocol with a quadratic loss of efficiency). For normal-form protocols, the attacker Eve of [IR] operated as follows: after Bob sends a message, Eve samples $O(n \log n)$ executions of Bob conditioned on all the information she has at that moment (communication transcript and previous oracle answers). To sample an execution Eve chooses uniformly random tapes and oracle answers that are consistent

[1] This is not to be confused with some more recent works such as [BR], that combine the random oracle model with assumptions on the intractability of other problems such factoring or the RSA problem.

[2] More accurately, [IR] gave an $O(m^6 \log m)$-query attack where m is the maximum of the number of queries n and the number of communication rounds, though we believe their analysis could be improved to an $O(n^6 \log n)$-query attack. For the sake of simplicity, when discussing [IR]'s results we will assume that $m = n$, though for our result we do not need to make this assumption.

with the information she has, but after sampling an execution she makes all the queries asked during this execution and records the answers. (Generally, the true answers will not be the same answers as the one Eve guessed when sampling the execution.) Similarly, after every message Alice sends, Eve samples $O(n \log n)$ executions of Alice. Overall Eve will sample $\tilde{O}(n^2)$ executions making a total of $\tilde{O}(n^3)$ queries. The bulk of [IR]'s analysis is devoted to showing that with high probability Eve will learn all of the *intersection queries* of Alice and Bob— there will not be a query q that was asked by both Alice and Bob but not by Eve. It is not hard for Eve to find out the secret given this information (see also Theorem 7.1 below).

Our attacker is more frugal than [IR]'s attacker in the queries it asks. As is the case in [IR]'s attack, after Bob sends a message our attacker Eve will compute the same distribution $\mathcal{D}$ on possible executions of Bob conditioned on Eve's information on the oracle and the messages sent so far. But instead of sampling an execution from $\mathcal{D}$ and asking all queries in this execution, Eve will first compute whether there is a *heavy* query q that is asked in $\mathcal{D}$ with probability more than, say, $1/(10n)$. If there is such a query q then Eve will ask q from the oracle and record the answer. Note that if a query q is heavy and we sample $O(n \log n)$ executions from $\mathcal{D}$ then these executions will contain q with high probability. Thus [IR]'s attacker will ask every heavy query with high probability, but it might ask many more queries— this is the reason we say our attacker is more frugal. We note that our attacker may ask more than one query per message - as long as there exists a heavy query q and the total number of queries asked is less than cn^2 for some constant c (specified below) the attacker will make the query q.

Both in our case and [IR], the main difficulty in the analysis is that the communication transcript creates *dependencies* between the executions of Alice and Bob. In fact even conditioned on a particular transcript and the inputs and outputs of all intersection queries, the two executions are still dependent. This is the cause of much of the complications in [IR] and the need for many queries. We tackle this difficulty in a different way. We define an "imaginary experiment", where the oracle doesn't always behave properly but sometimes returns different answers when asked the same question. This imaginary experiment is defined to ensure that conditioned on Eve's knowledge the executions of Alice and Bob are independent of one another. This makes it much easier to show that our attack strategy succeeds in this experiment with high probability. We then conclude the proof by showing that the distributions of the two experiments are statistically close to one another.

3 Setting

A key exchange protocol Π involves Alice and Bob tossing coins r_a and r_b and then run a protocol having access to a random oracle $H : \{0,1\}^* \to \{0,1\}^*$. We assume that the protocol proceeds in some finite number of rounds, and no party asks the same query twice. In round k, if k is odd then Alice makes some number of queries and sends a message to Bob (and then Eve asks some oracle queries), and if k is even then Bob makes some queries and sends a message to Alice (and then Eve asks some oracle queries). At the end of the protocol Alice obtains an output string s_a and bob obtains an output string s_b. We assume that $\Pr[s_a = s_b] \geq \rho$ where the probability is over the coin tosses of Alice and Bob and the randomness of the oracle. We will establish Theorem 1.1 by proving that an attacker can make $O(n^2)$ queries to learn s_b with probability arbitrarily close to 1.

4 Finding intersection queries

We start by showing that an attacker can find the intersection queries of Alice and Bob with high probability. It turns out that this is the main step in showing that an attacker can find the secret with high probability (see Theorem 7.1 below).

Theorem 4.1. *Assume that Π is a key exchange protocol in the random oracle model where Alice and Bob ask at most n oracle queries each (so, together they make at most $2n$ queries), then there is an adversary Eve* *who has access to the messages sent between Alice and Bob and asks at most $2(\frac{n}{\epsilon})^2$ number of queries such that her queries contain all intersection queries of Alice and Bob with probability at least $1 - 3\epsilon$.*

4.1 The attack

After fixing n, let ℓ be the upper bound on the length of the queries that Alice or Bob might ask from the oracle. So we can think of the oracle H as a finite binary string of length $\sum_{1 \le i \le \ell} i2^i$. We assume that in the last round of the protocol Alice sends a special message LAST to Bob indicating the end of of the protocol.

Suppose at some point during the execution of the protocol, $M = [m_1, \ldots, m_k]$ is the sequence of messages sent by Alice and Bob to each other so far, and let $\mathcal{I}$ be the set of oracle query/answer pairs that Eve has learned up to that point. We define the random variable $A(M, \mathcal{I})$ (or just A for brevity) to be a random execution (i.e., random tape, the transcript of all oracle query/answers, and the messages received) of Alice which is produced by uniformly choosing (H, r_a) among all of oracle/random tape pairs which are consistent with $(M, \mathcal{I})$. So A describes r_a and all the oracle query/answer pairs that are used during the computation. We define $Q(A)$ to be the random variable denoting the set of oracle queries asked during the execution A.[3] We can similarly define the distribution $B = B(M, \mathcal{I})$, and then $Q(B)$ will denote the queries asked by Bob described in B. We define $Q = Q(A)$ if k is odd, and $Q = Q(B)$ if k is even.

Remark 4.2. In general, $A(M, \mathcal{I})$ is *not* the same distribution that is obtained by sampling H, r_a, and r_b uniformly conditioned on $(M, \mathcal{I})$ and then taking Alice's part of this execution— fixing $(M, \mathcal{I})$ may introduce *dependencies* between r_a and r_b. For example, think of a protocol where Bob sets x to be either 0^ℓ or 1^ℓ with probability $1/2$, and sends $y = H(x)$ to Alice, and then Alice queries 0^ℓ and gets back $y' = H(0^\ell)$. Fix the transcript $M = [y]$ and the set $\mathcal{I} = \emptyset$. The distribution $A(M, \mathcal{I}) = A(y, \emptyset)$ will choose a random execution of Alice by choosing a random answer y', since any answer is consistent with this transcript. In contrast, if we sample H, r_a, r_b conditioned on $(y, \emptyset)$, the marginal distribution of H will not be a random function but rather a function that has the value y in either 0^ℓ or 1^ℓ. Note that even if $x = 1^\ell$, and so Alice and Bob query disjoint entries, the transcript still creates some dependencies, since with high probability we'll have $y' \neq y$ and so Alice will learn the value of x and of $y = H(x)$.

This example may seem benign but actually examples like this are what make the analysis of attacks on key exchange protocols very subtle, and the main reason for the technical difficulty of the paper [IR].

Heavy and light queries: Let M be a partial transcript of the protocol up to round k, and $\mathcal{I}$ be some set of oracle query/answer pairs. We say that a string $q \in \{0,1\}^*$ is *heavy* with respect to $M, \mathcal{I}$ if $\Pr[q \in Q | M, \mathcal{I}] \ge \frac{\epsilon}{n}$ (where $Q = Q(A(M, \mathcal{I}))$ if k is odd and $Q = Q(B(M, \mathcal{I}))$ otherwise) and say that q is *light* otherwise (so the definition of heaviness considers Alice's possible computations or Bob's depending on whether k is even or odd).

Description of attack (of learning intersection queries). The attack could be described in one line as follows: whenever Eve learns some information that makes a query q heavy, it asks q from the oracle, up to a total of $2(n/\epsilon)^2$ queries. Specifically, we define Eve as follows: after receiving the message m_k sent in the round k, Eve sees if there is any query $q \notin \mathcal{I}$ that is heavy based on her new information (where $\mathcal{I}$ denotes the query/answer pairs Eve knows up to that point). If so, it asks q (if there are more than one, it asks the lexicographically first one). It then updates $\mathcal{I}$ with the new information and repeats the process until there are no more heavy queries. (Then Bob will continue his computation and will send his message and so on.) Eve will stop making any queries after asking $2(n/\epsilon)^2$ many of them, even if it is in the middle of a round. (It might stop making queries even earlier just because there are no more heavy queries.)

[3] We might use the terms distribution and random variable interchangeably, and by a random variable assigned to a distribution we simply mean the output of sampling from that distribution.

5 Analysis

For the sake of analysis, we will consider a modified version of Eve that does not stop making heavy queries even if it means asking more than $2(n/\epsilon)^2$ queries. For i, $1 \le i \le n$, we say the attack has *failed in Bob's i^{th} query*, if Bob's i^{th} query was asked by Alice before but was not in $\mathcal{I}$, where $\mathcal{I}$ is the query/answer pairs known to Eve just before Bob makes this query. We define failure in Alice's i^{th} query similarly. Let Fail be the event that the attack fails in some query (note that in particular if the attack ends with Eve missing an intersection query then Fail has happened), and let Long be the event that Eve asks more than $2(n/\epsilon)^2$ queries. To prove the attack works, it suffices to show that $\Pr[\mathsf{Fail} \vee \mathsf{Long}] \le 3\epsilon$, which will follow from the following two claims:

Claim 5.1. $\Pr[\mathsf{Fail}] \le 2\epsilon$.

Claim 5.2. $\Pr[\neg\mathsf{Fail} \wedge \mathsf{Long}] \le \epsilon$.

The proof of the claims will follow by defining a new imaginary experiment Imag in a way that as long as Fail has not happened, the distribution of executions in Imag is exactly the same as that in Real. It means that the event Fail is also well defined in Imag and we have $\Pr_{\mathsf{Imag}}[\mathsf{Fail}] = \Pr_{\mathsf{Real}}[\mathsf{Fail}]$. Moreover for any event D defined in Real, the event $\neg\mathsf{Fail} \wedge D$ is also well defined in Imag and we have $\Pr_{\mathsf{Imag}}[\neg\mathsf{Fail} \wedge D] = \Pr_{\mathsf{Real}}[\neg\mathsf{Fail} \wedge D]$. Thus, it will suffice to prove claims 5.1 and 5.2 in the probability space of executions of the imaginary experiment Imag.

5.1 The imaginary experiment Imag

Description of Imag: The codes of Alice, Bob, and Eve in Imag is the same as those in Real. The only difference between the experiments is that sometimes we will fool Alice or Bob by not giving the true answer of the oracle to them. In the description of Imag it will be more convenient to think that there is no oracle H used in the experiment, and the answers to the oracle queries are determined by the random tapes of Alice, Bob, and Eve (denoted by r_a, r_b, and r_e respectively, and each assumed to be of length L where L depends on n). So the random tapes of the parties are longer in the experiment $\mathsf{Imag}\dot{\mathrm{T}}$he experiment Imag is defined as follows:

- If Alice asks a query q in the set $\mathcal{I}$ of queries known by Eve, the answer of $\mathcal{I}$ will be used. If Alice asks a query q that is not in the set $\mathcal{I}$, then she gets a uniformly random answer y, no matter whether or not Bob has asked q before. (And hence her answer could possibly be inconsistent with Bob's answer.) We think of the random choice for y as being taken from Alice's random tape r_a (i.e., r_a contains an entry for every possible query that Alice could make, and we use that entry for queries that are not in $\mathcal{I}$).
- Similarly, if Bob makes a query that is not in $\mathcal{I}$, then he chooses a random answer to this query using his random tape.
- In the case Alice makes a query q that is not in $\mathcal{I}$ but was asked before by Bob, we say that the experiment *failed* at this point. Similarly we say the experiment fails at a point where Bob asks a query not in $\mathcal{I}$ but that was previously asked by Alice. Note that even if the experiment fails, we do *not* stop running it and will continue the experiment.
- If Eve asks a query q for the first time, the answer will be randomly chosen using her own randomness r_e. If Eve asks a query q that was previously asked by Alice or Bob, she gets the same answer as they did. If both Alice and Bob previously asked q and received different answers, and then Eve asks q as well then we halt the experiment and the execution of Alice, Bob, and Eve will be stopped. Because there will be no answer for the query q, the set $\mathcal{I}$ will not get updated, and the sequence of the messages

sent will not change anymore. Note that this can only happen if the experiment has already failed at some previous point.

Note that in Imag the execution of Alice is only a function of r_a, $\mathcal{I}$, and the messages she receives (rather than the whole randomness of the system: r_a, r_b, r_e), and similarly the execution of Bob depends only on r_b, $\mathcal{I}$ and the transcript. When we run either Imag or Real, they behave the same until there is a point that Alice or Bob asks a query from the oracle that the other party has asked and Eve has not asked (i.e., a failure). At this moment, if we are in Real, the true oracle answer is returned, and if we are in Imag, the answer will be chosen at random. Therefore as long as Fail does not happen they are the same experiments. (Note that if Fail does not happen then we never have to halt the experiment due to conflicting answers.)

The following lemma implies that $\Pr_{\mathsf{Imag}}[\neg\mathsf{Fail} \wedge \mathsf{Long}] \leq \epsilon$, proving Claim 5.2:

Lemma 5.3. *The expected number of queries asked by Eve in* Imag *is at most* $\frac{2n^2}{\epsilon}$. *So, by Markov bound, the probability that Eve makes more than* $2(\frac{n}{\epsilon})^2$ *queries is less than* ϵ. *That is,* $\Pr_{\mathsf{Imag}}[\mathsf{Long}] \leq \epsilon$.

Proof. Define the random variable Y_j to be 1 if the j^{th} query Eve makes was asked before by Alice or Bob. Clearly $\sum_j Y_j \leq 2n$ since Alice and Bob each make at most n queries, and hence

$$\sum_j \mathbb{E}[Y_j] = \mathbb{E}[\sum_j Y_j] \leq 2n \quad . \tag{1}$$

Claim: Let p_j be the probability that Eve asks the j^{th} query. Then $\mathbb{E}[Y_j] \geq \frac{p_j \epsilon}{n}$.

Note that $\sum_j p_j$ is the expected number of queries asked by Eve, and the claim implies that $\sum_j p_j \leq \frac{n}{\epsilon} \sum \mathbb{E}[Y_j] \leq \frac{2n^2}{\epsilon}$, hence proving the lemma.

Proof of Claim: Define Y_j^q to be 1 if the j^{th} query that Eve asks is q and q was asked before by Alice or Bob. Then, $\mathbb{E}[Y_j] = \sum_q \mathbb{E}[Y_j^q]$. Let $\mathcal{O}_j$ be a random variable that whenever there is a j^{th} query asked by Eve, it denotes the information (i.e., transcript and query/answer pairs) that Eve has up to the point when it makes its j^{th} query. In an execution where Eve makes less than j queries, we define $\mathcal{O}_j = \perp$. Note that the j^{th} query of Eve is determined by $\mathcal{O}_j$ which we denote by $q(\mathcal{O}_j)$ (and we define $q(\perp) = \perp$). Let $\mathcal{W}_j = SUPP(\mathcal{O}_j)$ (including $\perp$), and so we will have:

$$\mathbb{E}[Y_j^q] = \sum_{\substack{L \in \mathcal{W}_j \\ q(L)=q}} \Pr[\mathcal{O}_j = L] \Pr[q \text{ asked before by Alice or Bob} \mid \mathcal{O}_j = L] \quad .$$

But by definition, if $q(L) = q$ we have $\Pr[q \text{ is asked before by Alice or Bob} \mid \mathcal{O}_j = L] \geq \epsilon/n$. Meaning that $\mathbb{E}[Y_j^q] \geq \frac{\epsilon}{n} \sum_{\substack{L \in \mathcal{W}_j \\ q=q(L)}} \Pr[\mathcal{O}_j = L]$, and hence

$$\mathbb{E}[Y_j] \geq \frac{\epsilon}{n} \sum_{q \neq \perp} \sum_{\substack{L \in \mathcal{W}_j \\ q=q(L)}} \Pr[\mathcal{O}_j = L] = \frac{\epsilon}{n} \sum_{L \in \mathcal{W}_j} \Pr[\text{Eve queries some } q \text{ as its } j^{\text{th}} \text{ query} \mid \mathcal{O}_j = L] \Pr[\mathcal{O}_j = L] = \frac{\epsilon}{n} p_j \quad .$$

□

As we said above, what heavy query to be asked by Eve is determined only by $(M, \mathcal{I})$ where $M = [m_1, \ldots, m_k]$ is the sequence of first k messages sent and $\mathcal{I}$ is what Eve knows about the oracle. It means we can check the consistency of $(M, \mathcal{I})$ without knowing r_a or r_b, because the answers to the queries are also contained in $\mathcal{I}$. Roughly speaking, we call the pair $(M, \mathcal{I})$ where $M = [m_1, \ldots, m_k]$ *consistent* if $(M, \mathcal{I})$ self-justifies itself, and describes all that happens to Eve till the end of the k^{th} round. More formally, the following process checks the consistency of $(M, \mathcal{I})$. We start by looking at the first heavy query q determined by the message m_1 (and the empty set $\emptyset$ of query/answre pairs known to Eve), and this query should be in the set $\mathcal{I}$ (i.e., otherwise we reject). We add the pair (q, a) to the empty set where a is the answer determined for q by $\mathcal{I}$. We continue the process of finding the next heavy query, checking if it is in $\mathcal{I}$, and using the answer that $\mathcal{I}$ determines. When there is no next heavy query to learn for round i: if $i < k$, we add the next available message m_{i+1} to our list of messages and continue the process, but if $i = k$, we accept iff we have generated all the queries of $\mathcal{I}$ as a heavy query at some point before (i.e., there is nothing extra left in $\mathcal{I}$).

Now, we call the triple $(M, \mathcal{I}, r_a)$ (where $M = [m_1, \ldots, m_k]$) *consistent*, if:

1. $(M, \mathcal{I})$ is consistent.
2. If we use r_a, M, and $\mathcal{I}$ to run Alice (which are enough for doing so), the computation goes consistently till m_k is sent. Note that because of the halting condition in the definition of the experiment Imag, all the queries of Alice which are also in $\mathcal{I}$ should be answered the same no matter if it is Alice or Eve who asks the query first.

A similar definition of consistency holds for $(M, \mathcal{I}, r_b)$ where r_b is Bob's randomness. Note that we could check the consistency of $(M, \mathcal{I}, r_a)$ without knowing r_b (and vice versa). We can further define the consistency of a quadruple $(M, \mathcal{I}, r_a, r_b)$ as:

1. $(M, \mathcal{I})$ is consistent.
2. If we use r_a, r_b, M, and $\mathcal{I}$ to run Alice and Bob, the computation goes consistently till m_k is sent.

Lemma 5.4. *The quadruple $(M, \mathcal{I}, r_a, r_b)$ is consistent, iff both the triples $(M, \mathcal{I}, r_a)$ and $(M, \mathcal{I}, r_b)$ are consistent.*

Proof. Note that if $(M, \mathcal{I}, r_a, r_b)$ is consistent, then both $(M, \mathcal{I}, r_a)$ and $(M, \mathcal{I}, r_b)$ are consistent by definition. On the other hand, suppose that $(M, \mathcal{I}, r_a)$ and $(M, \mathcal{I}, r_b)$ are consistent. As we said before, it means that if Alice's computation determined by $(M, \mathcal{I}, r_a)$ asks a query which is in $\mathcal{I}$, it should get the answer determined by $\mathcal{I}$. The same is for Bob's computation. So, if they ask the same query which is available in $\mathcal{I}$, they have the same answer for it which is equal to the answer of $\mathcal{I}$, no matter the order by which Alice, Bob, and Eve ask the query. It means that $(M, \mathcal{I}, r_a, r_b)$ is also consistent. □

The following lemma shows that the distribution of $A(M, \mathcal{I})$ defined in the attack of Real is in fact the *true* distribution of Alice in the experiment Imag conditioned on getting $(M, \mathcal{I})$ during the experiment.

Lemma 5.5. *For every odd k and a consistent triple $(M, \mathcal{I}, r_b)$ of: sequence of messages $M = [m_1, \ldots, m_k]$, set of query/answer pairs $\mathcal{I}$ known to Eve, and Bob's random tape r_b, the distribution $A(M, \mathcal{I}, r_b)$ of Alice's executions in* Imag *up to round k conditioned on $(M, \mathcal{I}, r_b)$ is equal to the distribution $A(M, \mathcal{I})$ defined in Section 4.1.*

As we remarked in Section 4.1, Lemma 5.5 is not true for experiment Real.

Proof. Recall that $A(M, \mathcal{I})$ in Real is obtained by uniformly choosing a (r_a, H) conditioned on getting an execution of Alice consistent with $(M, \mathcal{I})$ till m_k is sent. In Imag's terminology, this is the same as uniformly (only) choosing r_a (because it contains the oracle's description) such that $(M, \mathcal{I}, r_a)$ is consistent. On the other hand, $A(M, \mathcal{I}, r_b)$ uniformly chooses r_a conditioned on $(M, \mathcal{I}, r_a, r_b)$ being consistent. But because we know that $(M, \mathcal{I}, r_b)$ is consistent, by lemma 5.4 this is the same as uniformly choosing r_a conditioned on $(M, \mathcal{I}, r_a)$ begin consistent. □

Thus claim 5.1 will follow from the following lemma:

Lemma 5.6. $\Pr_{\mathsf{Imag}}[\mathsf{Fail}] \leq 2\epsilon$.

This lemma, in turn, follows from the next lemma.

Lemma 5.7. *For every $j \in [n]$, the probability that* Imag *fails in Bob's j^{th} query is at most ϵ/n.*

Proof. We prove the lemma in the stronger form that it holds even conditioned on a consistent $(M, \mathcal{I}, r_b)$ (where $M = [m_1, \dots, m_k]$) describing all the information known to Bob and Eve up to the point that Bob makes its j^{th} query in round $k+1$. Since $(M, \mathcal{I}, r_b)$ describes Bob's computation till that moment, we shall choose $(M, \mathcal{I}, r_b)$ in such a way that Bob's j^{th} query is asked in round $k+1$ and (in addition to $(M, \mathcal{I}, r_b)$ begin consistent) Bob's computation goes consistently till the moment that he asks his j^{th} query. Hence by choosing such $(M, \mathcal{I}, r_b)$ Bob's j^{th} query q will be fixed. If $q \in \mathcal{I}$ then certainly there is no failure, and so it suffices to bound the probability that $q \in Q(A(M, \mathcal{I}, r_b))$ when $q \notin \mathcal{I}$ (where $A(M, \mathcal{I}, r_b)$ is the distribution of Alice's computation till the end of round k in experiment Imag condition on $(M, \mathcal{I}, r_b)$ as defined in Lemma 5.5). That is, prove that if that if $q \notin \mathcal{I}$ then

$$\Pr_{r_a}[q \in Q(A(M, \mathcal{I}, r_b))] \leq \tfrac{\epsilon}{n} \tag{2}$$

But by Lemma 5.5, this is the same as $\Pr[q \in Q(A(M, \mathcal{I}))]$ (where $A(M, \mathcal{I})$ is defined in the attack) which we know is at most ϵ/n by the fact that q is light at this point (or else it would have been in $\mathcal{I}$). □

We can prove in a symmetric fashion that the probability that we fail in Alice's j^{th} query is at most ϵ/n, establishing by the union bound that the total probability of the event Fail is at most $2n(\epsilon/n) = 2\epsilon$, thus completing the proof of Lemma 5.6. □

6 Finding the secret

Now, we turn to the question of finding the secret.

Theorem 6.1. *Assume that the total number of queries asked by Alice and Bob is at most n (so, together they make at most $2n$ queries), and their outputs agree with probability at least ρ having access to a random oracle. Then there is an adversary Eve asking at most $18(\frac{n}{\delta})^2$ number of queries such that Eve's output agrees with Bob's output with probability at least $\rho - \delta$.*

Proof. We will show how to find Bob's secret in the experiment Imag by first running the adversary Eve of Theorem 7.1 with $\epsilon = \delta/3$, and then somehow guess Bob's secret with probability at least ρ (just like Alice) as we will see in a moment. By Claims 5.1,5.2 and noting that conditioned on ¬Fail there is no difference between Imag and Real, if we run the same attack (to learn Bob's secret) in Real and if we stop learning heavy queries after asking $2(n/\epsilon)^2 = 18(n/\delta)^2$ steps, our chance of guessing Bob's secret correctly decreases at most by $3\epsilon = \delta$, and we will success with probability at least $\rho - \delta$.

What we have to do after running Eve is very simple. After Alice sends the last message LAST,[4] if M and $\mathcal{I}$ indicate the sequence of messages and the set of oracle query/answers that Eve knows, we sample from $A(M, \mathcal{I})$, and will output whatever it generates as output. Note that by (proof of) Lemma 5.5 conditioned on M, $\mathcal{I}$, and any computation for Bob (which determines his output), the distribution of $A(M, \mathcal{I})$ is just the same as the true distribution of Alice in the experiment Imag. So, our probability of guessing Bob's secret is just the same as that of Alice. □

[4] Actually, there is no need to learn Alice's heavy queries after she sends LAST, although it does not hurt us. That is because as we saw Eve learns the intersection queries *before* they become so.

7 The random permutation model

The attack and its analysis is the random permutation (RP) model is actually very similar to that of RO model, and we can prove the following theorem.

Theorem 7.1. *Assume that the total number of queries asked by Alice and Bob is at most n (so, together they make at most $2n$ queries), and their outputs agree with probability at least ρ having access to a random permutation oracle. Then there is an adversary Eve asking $O(\frac{n}{\delta})^2$ number of queries such that Eve's output agrees with Bob's output with probability at least $\rho - \delta$.*

Difference in attack. There are two differences in the attack:

- We merge all the queries over the domains of size at most $N = 18(n/\epsilon)^2 + n^2/\epsilon + n$, as a single query. Note that the total number of members of such domains is at most $2N = O((n/\epsilon)^2)$. We call this query the "big" query. We pretend that Alice, Bob, and Eve ask the big query whenever they want to ask a query in a domain of size at most N. This query counts only on in the analysis. Then by expanding the big query (if asked) of Eve into real queries, it adds $2N$ more queries to our attack and the total number of them is still at most $3N = O((n/\epsilon)^2)$.
- Another difference is in the definition of $A(M, \mathcal{I})$: Now, it is the uniform distribution over Alice's computation such that:
    - The oracle used by Alice is a permutation oracle over each length of queries.
    - The oracle answers are consistent with $\mathcal{I}$ (as part of the description of the whole RP oracle).
    - The execution of Alice is consistent with M (which like before).

  The new definition of $B(M, \mathcal{I})$ is also similar.

Difference of Analysis. First note that as we said the number of the queries Eve asks is still $O((n/\epsilon)^2)$, and we need to show that Eve's output equals Bob's output with probability at least $\rho - O(\epsilon)$. Then by taking $\delta = \epsilon/c$ for a big enough constant c we will be done. The definition of Imag experiment is like before with the difference that: whenever we were supposed to choose a totally random answer for Alice (resp. Bob), now we choose the answer at random among the answers not used by Alice (resp. Bob) nor Eve for a query of the same length. In addition, whenever Eve asks a query for the first time, the answer will be chosen at random different from all the answers already used by Alice, Bob, or Eve herself for a query of the same length.

Now a point is that even if Fail does not happen, there is another event that distinguishes Real from Imag, and that is when Alice and Bob get the same answer for two different queries of the same length in Imag, which is impossible to happen in Real. So, we define the new event Col to be the mentioned event in Imag. (We will not stop the experiment Imag if Col happens). Note that still, as long as $\mathsf{Col} \vee \mathsf{Fail}$ does not happen in Imag, and Fail does not happen in Real they are the same experiments.

Claim 7.2. $\Pr_{\mathsf{Imag}}[\mathsf{Col}] \leq \epsilon$.

Proof. Note that the collision could happen only when, say, Alice asks a query which neither she nor Eve has asked before, but the answer equals to an answer that Bob has already received. When

we are choosing a random answer for a query for Alice (other than the big query which is a different case), there are at most n answers already used by Alice herself, and $18(n/\epsilon)^2$ answers already used by Eve, and therefore, the answer is chosen at random from a space of size at least n^2/ϵ. So, the probability that the answer to the i^{th} query of Alice collides with the j^{th} query of Bob is at most ϵ/n^2 (regardless of who asks first). Hence, because there are at most n^2 such pairs, the probability of Col is at most ϵ. □

Now we can find the secret similar to what we did in Section 6. Namely, we run the modified attacked descried above in the modified experiment Imag and at the end will sample $A(M, \mathcal{I})$ and output its output. The probability that our output agrees with Bob's in experiment Imag is at least ρ (like Alice), and if we run the same attack in Real and stop making more queries after asking $3N$ many of them, our chance of success decreases at most by $O(\epsilon)$. All we need is a new version of Lemma 5.5 with new definitions of $A(M, \mathcal{I})$ and Imag, but the proof is just similar to the one given above.

Acknowledgements. We thank Russell Impagliazzo for useful discussions.

References

[BR] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the First Annual Conference on Computer and Communications Security*, pages 62–73. ACM, November 1993.

[CGH] R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology, Revisited. In *Proc. 30th STOC*, pages 209–218. ACM, 1998.

[DH] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, Nov. 1976.

[IR] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proc. 21st STOC*, pages 44–61. ACM, 1989.

[Mer] R. C. Merkle. Secure Communications Over Insecure Channels. *Commun. ACM*, 21(4):294–299, 1978.

[RSA] R. L. Rivest, A. Shamir, and L. M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, Feb 1978.

A Some Useful Facts

We state some simple but useful facts and definitions on random variables without proof.

Lemma A.1. *For any pair of events A, B, we have $\Pr[A \vee B] \geq \Pr[A \mid \neg B]$.*

Definition A.2. The statistical difference $\Delta(X, Y)$ of two finite random variables X, Y is defined as:

$$\frac{1}{2} \sum_a |\Pr[X = a] - \Pr[Y = a]| \ .$$

Lemma A.3. *If A, B are random variables, and the event E is defined over $SUPP(A) \cup SUPP(B)$, then $|\Pr[E(A)] - \Pr[E(B)]| \leq \Delta(A, B)$.*

Lemma A.4. *If the random variable A' is a function of random variable A, and the random variable B' is a function of B, then $\Delta(A', B') \leq \Delta(A, B)$.*

Lemma A.5. *If the event E is defined over the random variable A, and the event D is defined over the random variable B, and we have $\Delta(A \mid E, B \mid D) = 0$, then $\Delta(A, B) \leq (\Pr[E] + \Pr[D])/2$.*

Lemma A.6. *Suppose A, B, X, and Y are random variables such that for any $a \in SUPP(A) \cap SUPP(B)$ we have $\Delta(X \mid A = a, Y \mid B = a) = 0$. Then $\Delta((A, X), (B, Y)) \leq \Delta(A, B)$.*