Merkle Puzzles are Optimal — an $O(n^2)$-query attack on key exchange from a random oracle

Boaz Barak* Mohammad Mahmoody-Ghidary†

July 10, 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.

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 designed by Merkle in 1974 (although only published in 1978 [Mer]). Merkle's protocol allows two parties Alice and Bob to agree on a random number k that will not be known to an eavesdropping adversary Eve. It is described in Figure 1.

One problem with Merkle's protocol is that its security was only analyzed in the random oracle model which does not necessarily capture security when instantiated with a cryptographic one-way or hash function [CGH].[1] But the most serious issue with Merkle's protocol 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 so could perhaps be vulnerable to unforseen attacks using this algebraic structure, it remained an important question to show whether there exist key exchange protocols with superpolynomial security that use only a random oracle.[2] 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.[3]

*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.

[1] Recently, Biham, Goren and Ishai [BGI] gave a security analysis for Merkle's protocol under some concrete complexity assumptions, namely existence of exponentially hard one-way functions.

[2] 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.

[3] 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.

Merkle's Key Exchange Protocol

Let n be the security parameter. All parties have access to oracle to a function $H : \{0,1\}^\ell \to \{0,1\}^\ell$ chosen at random, where $\ell \gg \log n$. The protocol 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 $a_i = H(x_i)$ (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. Note that assuming $2^\ell \gg n^4$, H will be one to one on $[n^2]$ with very high probability and hence $H(x_i) = H(y_j)$ 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.

Figure 1: Merkle's key exchange protocol. (Merkle described his protocol using "puzzles" that can be implemented via some ideal cryptographic primitive; we describe the protocol in the case that the puzzles are implemented by a random oracle.)

Since a random oracle is in particular a one-way function (with high probability), this implied that there is no construction of a key exchange protocol based on a one-way function with 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). Indeed, that was the motivation behind their result.

Still, Impagliazzo and Rudich left as an open question [IR, Section 8] whether there exist protocols in the random oracle model with $\omega(n^2)$ security or in fact Merkle's protocol is optimal. One motivation for this question is practical— protocols with sufficiently large polynomial gap could be secure enough in practice (e.g., a key exchange protocol taking 10^9 operations to run and $(10^9)^6 = 10^{54}$ operations to break could be good enough for many applications), and in fact as technology improves, such polynomial gaps only become more useful. Another motivation is theoretical— since Merkle's protocol has very limited interaction (it consists of one round in which both parties simultaneously broadcast a message) it's natural to ask whether more interaction can help achieve some polynomial advantage over this simple protocol.

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 $O(\frac{n^2}{\delta^2})$ queries to the oracle whose output agrees with Bob's output with probability at least $\rho - \delta$.*

As is the case in [IR], our result can be shown to rule out the existence of *black-box* constructions of a key exchange protocol with super-quadratic security from a one-way function, though we omit the details.[4]

[4]To formalize the above statement, one needs to quantify what it means for a black-box reduction of a primitive X to a

To the best of our knowledge, no better bound than [IR] was previously known even in this case, where one does not assume the one-way function is a random oracle (hence making the task of proving a negative result easier). We note that similarly to previous black-box separation results, our adversary can be implemented efficiently in a relativized world where $\mathbf{P} = \mathbf{NP}$, meaning that we also rule out the somewhat larger family of *relativizing* reductions as well.

Correction of error: A previous version of this manuscript [BMG] claimed a different proof of the same result. However, we have found a bug in that proof— see Appendix A. In fact the current proof is quite different from the one claimed in [BMG]. In [BMG] we also claimed an extension of Theorem 1.1 to the case of protocols with an oracle to a *random permutation* (i.e., a random one-to-one function R from $\{0,1\}^*$ to $\{0,1\}^*$ such that $|R(x)| = |x|$ for every $x \in \{0,1\}^*$). We do not know of an extension of the current proof to this model, beyond the observation of [IR] that any m-query attack in the random oracle model translates into an $O(m^2)$-query attack in the random permutation model. Hence our results imply an $O(n^4)$-query attack in the latter model, improving on the previous $\tilde{O}(n^{12})$ attack of [IR].

We also note that shortly after we posted the manuscript [BMG], Sotakova [Sot] posted an independently obtained weaker result, showing that protocols with only one round of interaction (each party sends one message) and non-adaptive queries can achieve at most $O(n^2)$ security. In contrast, as in the work of [IR], in this paper we allow protocols where the parties' choice of queries is adaptive and they can use an arbitrary polynomial number of interaction rounds.[5] The one-round case seems to be simpler, and in particular the bug found in our previous proof does not apply to that case.

2 Our techniques

It is instructive to compare our techniques with the techniques of the previous work by Impagliazzo and Rudich [IR]. In order to do this, we review [IR]'s attack and outline of analysis, and particularly the subtle issue of *dependence* between Alice and Bob that arises in both works. The main novelty of our work is the way we deal with this issue, which is different than the approach of [IR]. We believe that this review of [IR]'s analysis and the way it compares to ours can serve as a useful introduction to our actual analysis. However, no result of this section is used in the later sections, and so the reader should feel free at any time to skip ahead to Sections 3 and 4 that contain our actual attack and its analysis.

Consider a protocol that consists of n rounds of interaction, where each party makes exactly one oracle query before sending its message. [IR] called protocols of this type "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). Although without loss of generality the attacker Eve of a key exchange protocol can defer all of her computation till after the interaction between Alice and Bob is finished, it is conceptually simpler in both [IR]'s case and ours to think of the attacker Eve as running concurrently with Alice and Bob. In particular, the attacker Eve of [IR] performed the following operations after each round i of the protocol:

- If the round i is one in which Bob sent a message, then at this point Eve samples $1000n \log n$ random executions of Bob from the distributions of Bob's executions that are consistent with the information that Eve has at that moment (communication transcript and previous oracle answers). That is, Eve samples a uniformly random tape for Bob and uniformly random query answers subject to being consistent with Eve's information. After each time that she samples an execution, Eve asks the oracle all

primitive Y to show T security. Since there is no meaningful notion of running time for black-box reduction, we believe the correct formalization is that the reduction works for every adversary that makes at most T queries to the primitive Y. Under this formalization, the above statement follows immediately from our results combined with the well-known fact that a random-oracle is a one-way function.

[5] In fact, because we count only the number of *oracle queries* made by the honest parties, we can even allow a super-polynomial number of rounds.

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, if the round i is one in which Alice sent a message then Eve samples $1000n \log n$ executions of Alice and makes the corresponding queries.

Overall Eve will sample $\tilde{O}(n^2)$ executions making a total of $\tilde{O}(n^3)$ queries. It's not hard to see that as long as Eve learns all of the *intersection queries* (queries asked by both Alice and Bob during the execution) then she can recover the shared secret with high probability (see also Theorem 5.1 below). Thus the bulk of [IR]'s analysis was devoted to showing the following statement, denoted below by **(*)**: *With probability at least* 0.9 *Eve never fails, where we say that Eve* fails *at round i if the query made in this round by, say, Alice was asked previously by Bob but not by Eve.*

2.1 The issue of independence

At first look, it may seem that one could easily prove **(*)**. Indeed, **(*)** will follow by showing that at any round i, the probability that Eve fails in round i *for the first time* is at most $1/(10n)$. Now all the communication between Alice and Bob is observed by Eve, and if no failure has yet happened then Eve has also observed all the intersection queries so far. Because the answers for non-intersection queries are completely random and independent from one another it seems that Alice has no more information about Bob than Eve does, and hence if the probability that Alice's query q was asked before by Bob is more than $1/(10n)$ then this query q has probability at least $1/(10n)$ to appear in each one of Eve's sampled executions of Bob. Since Eve makes $1000n \log n$ such samples, the probability that Eve misses q would be bounded by $(1 - \frac{1}{10n})^{1000n \log n} \ll 1/(10n)$.

When trying to make this intuition into a proof, the assumption that Eve has as much information about Bob as Alice does translates to the following statement: conditioned on Eve's information, the distributions of Alice's view and Bob's view are *independent* from one another.[6] Indeed, if this statement was true then the above paragraph could be easily translated into a proof that [IR]'s attacker is successful, and it wouldn't have been hard to optimize this attacker to achieve $O(n^2)$ queries. Alas, this statement is false. Intuitively the reason is the following: even the fact that Eve has not missed any intersection queries is some non-trivial information that Alice and Bob share and creates dependence between them.[7]

Impagliazzo and Rudich [IR] dealt with this issue by a "charging argument", where they showed that the probability of such dependence can be charged in a certain way to one of the executions sampled by Eve, in a way that at most n samples can be charged at each round. The exact details are not crucial to the current work, though this is to some extent the heart of [IR]'s analysis and the cause of most of the technical complications there.

2.2 Our approach

We now describe our approach and how it differs from the previous proof of [IR]. The discussion below is somewhat high level and vague, and glosses over some important details. Again, the reader is welcome to skip ahead at any time to Section 3 that contains the full description of our attack, and does not depend on this section in any way.

Our attacking algorithm follows the same general outline, but has two important differences from the attacker of [IR]:

[6] Readers familiar with the setting of communication complexity may note that this is analogous to the well known fact that conditioning on any transcript of a 2-party communication protocol results in a product distribution (i.e., combinatorial rectangle) over the inputs. However, things are different in the presence of a random oracle.

[7] As a simple example consider a protocol where in the first round Alice chooses x to be either the string 0^n or 1^n at random, queries the oracle H at x and sends $y = H(x)$ to Bob. Now Bob makes the query 1^n and gets $y' = H(1^n)$. Now even if Alice chose $x = 0^n$ and hence Alice and Bob have no intersection queries, Bob can find out the value of x just by observing that $y' \neq y$.

1. One *quantitative* difference is that while our attacker Eve also computes a distribution $\mathcal{D}$ of possible executions of Alice and Bob conditioned on her knowledge, she does *not* sample from $\mathcal{D}$ full executions and then ask the arising queries. Rather, she computes whether there is any *heavy query*— a string $q \in \{0,1\}^*$ that has probability more than, say, $1/(100n)$ of being queried in $\mathcal{D}$— and makes only such heavy queries. Intuitively, since Alice and Bob make at most $2n$ queries, the total number of heavy queries (and hence the query complexity of Eve) is bounded by $O(n^2)$. The actual analysis is more involved since the distribution $\mathcal{D}$ keeps changing as Eve learns more information through the messages she observes and query answers she receives. We omit the details in this high-level overview.

2. The *qualitative* difference between the two attackers is that we do not consider the same distribution $\mathcal{D}$ that was considered by [IR]. Their attacker to some extent "pretended" that the conditional distributions of Alice and Bob are independent from one another, and hence when trying to guess Bob's queries, only sampled consistent executions of Bob. In contrast, we define our distribution $\mathcal{D}$ to be the *real* distribution of Alice and Bob, where there could be dependencies between them. Thus to sample from our distribution $\mathcal{D}$ one would need to sample a *pair* of executions of Alice and Bob (random tapes and oracle answers) that are *jointly consistent*. Another (less important) point is that the distribution $\mathcal{D}$ computed by Eve at each point in time will be conditioned not only on Eve's knowledge so far, but also on the event that she has not failed until this point.

The main challenge in the analysis is to prove that the attack is *successful*, that is that the statement **(*)** above holds, and in particular that the probability of failure at each round (or more generally, at each query of Alice or Bob) is bounded by, say, $1/(10n)$. Once more, things would have been easy if we knew that the distribution $\mathcal{D}$ of the possible executions of Alice and Bob conditioned on Eve's knowledge (and not having failed so far) is a *product distribution*, and hence Alice has no more information on Bob than Eve has. While this is not generally true, we show that in our attack this distribution is *close to being a product distribution*, in a precise sense we define below.

At any point in the execution, fix Eve's current information about the system and define a bipartite graph G whose left-side vertices correspond to possible executions of Alice that are consistent with Eve's information and right-side vertices correspond to possible executions of Bob consistent with Eve's information. We put an edge between two executions A and B if they are consistent with one another and moreover if they do not represent an execution in which Eve *failed* prior to this point (i.e., there is no intersection query that is asked in both executions A and B but not by Eve). The distribution $\mathcal{D}$ that our attacker Eve considers can be thought of as choosing a random edge in the graph G. (Note that the graph G and the distribution $\mathcal{D}$ change at each point that Eve learns some new information about the system.) If G was the complete bipartite clique then $\mathcal{D}$ would be a product distribution. What we show is that G is *dense* in the sense that each vertex is connected to at least half the vertices on the other side. We show that this implies that Alice's probability of hitting a query that Bob asked before is at most twice the probability that Eve does so if she chooses the most likely query based on her knowledge.

The bound on the degree is obtained by showing that G can be represented as a *disjointness graph*, where each vertex u is associated with a set $S(u)$ (from an arbitrarily large universe) and there is an edge between a left-side vertex u and a right-side vertex v if and only if $S(u) \cap S(v) = \emptyset$.[8] We show that this particular graph has the property that $|S(u)| \leq n$ for all vertices u, and also the property that the distribution $S(u) \cup S(v)$ for a random edge $\{u, v\}$ is *light* in the sense that there is no element q in the universe that has probability more than $1/(10n)$ of being contained in a set chosen from this distribution. We then show that these properties together imply that each vertex is connected to at least half of the vertices on the other side.

Comparison with [IR]. One can also phrase the analysis of [IR] in terms of a similar bipartite graph. Their argument involved fixing, say, Alice's execution which corresponds to fixing a left-side vertex u, they

[8] The set $S(u)$ will correspond to the queries that are made in the execution corresponding to u but *not* made by Eve.

then showed that if the degree of u is high (e.g., u is connected to at least half of the right side) then their attacker is likely not to fail at this point. On the other hand, they showed that if the degree of u is low, then by taking a random vertex v on the right side and making all queries in the corresponding execution to v, one is likely to make progress in the sense that we learn a new query made in the execution corresponding to u. Now there are at most n new queries to learn, and hence if we sample $1000n \log n$ executions, then in most of them we're in the high degree case. This potential/charging argument inherently requires sampling *all* queries of the execution, rather than only the heavy ones, hence incurring a cost of at least n^2 queries *per round* or n^3 queries total. It also seems hard to generalize this argument to protocols that are not in normal form, which is the reason their attacker for general protocols required $\tilde{\Omega}(n^6)$ queries.

3 Our attacker

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, that is a random function from $\{0,1\}^\ell$ to $\{0,1\}^\ell$ for some $\ell \in \mathbb{N}$. 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 there is some constant $\rho > 0$ such 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 ρ.

In this section we describe an attack for Eve trying to find a set of size $O(n^2)$ which contains all the queries asked by Alice and Bob in the random oracle model. This attack is analyzed in Section 4 to show that it is successful in finding all intersection queries and is efficient (i.e., will not ask more than $O(n^2)$ many queries). Then this attack is used in Section 5 in order to find the actual secret.

3.1 Attacking algorithm

We start by showing that an attacker can find all the *intersection queries* (those asked by both 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 5.1 below).

Theorem 3.1. *Let Π be a key exchange protocol in the random oracle model in which Alice and Bob ask at most n oracle queries each. Then for every $\delta > 0$ there is an adversary Eve who has access to the messages sent between Alice and Bob and asks at most $\frac{10^4 n^2}{\delta^2}$ number of queries such that Eve's queries contain all the intersection queries of Alice and Bob with probability at least $1 - \delta$.*

To prove Theorem 3.1 we need to show an attacking algorithm Eve that learns the intersection queries between Alice and Bob using at most $O(n^2)$ queries. Letting $\epsilon = \delta/100$, our attack can be described in one sentence as follows:

> *As long as there exists a string q such that conditioned on Eve's current knowledge and assuming that no intersection query was missed so far, the probability that q was asked in the past (by either Alice or Bob) is at least ϵ/n, Eve makes the query q to the oracle.*

To describe the attack more formally, we need to introduce some notation. We fix n to be the number of oracle queries asked by Alice and Bob and assume without loss of generality that all the queries are of length $\ell = \ell(n)$ for some $\ell \in \mathbb{N}$. We will make the simplifying assumption that the protocol is in *normal form*—that is, at every round of the protocol Alice or Bob make exactly one query to the oracle (and hence there are $2n$ rounds). Later in Section 4.3 we will show how our analysis extends to protocols that are not of this

form. Below and throughout the paper, we often identify a distribution $\mathcal{D}$ with a random variable distributed according to $\mathcal{D}$.

Executions and the distribution $\mathcal{EXEC}$. An *execution* of Alice, Bob, and Eve can be described by a tuple $(r_a, h_a, r_b, h_b, \mathcal{I})$ where r_a denotes Alice's random tape, h_a denotes the sequence of answers that Alice gets in response to her oracle queries during the execution, r_b and h_b are defined analogously, and $\mathcal{I}$ denotes the set of all query/answer pairs that Eve learns during the execution. We say that a tuple $(r_a, h_a, r_b, h_b, \mathcal{I})$ is *consistent* if it describes an execution of Alice, Bob and Eve in which whenever two parties make the same query to the oracle they get the same answer. A partial execution is an execution truncated at a certain point in time (that is, the transcripts contain only the oracle answers for queries that are asked up to that point). We denote by $\mathcal{EXEC}$ the distribution over (full) executions that is obtained by running the algorithms for Alice, Bob and Eve with uniformly chosen random tapes and a random oracle.

The distribution $\mathcal{EXEC}(M, \mathcal{I})$. For $M = [m_1, \ldots, m_i]$ a sequence of i messages, and $\mathcal{I}$ a set of query/answer pairs, we denote by $\mathcal{EXEC}(M, \mathcal{I})$ the distribution over partial executions up to the point in the system in which the i^{th} message is sent (by Alice or bob), where the transcript of messages equals M and the set of query/answers that Eve learns equals $\mathcal{I}$. Note that we can verify that $\mathcal{I}$ is consistent with M by simulating Eve's algorithm on the transcript M, checking that whenever Eve makes a query, this query is in $\mathcal{I}$, in which case we feed Eve with the corresponding answer (and verifying at the end that there are no "extra" queries in $\mathcal{I}$ not asked by Eve). Thus for every $(M, \mathcal{I})$ that can be obtained from running the protocol, the distribution $\mathcal{EXEC}(M, \mathcal{I})$ is equal to the distribution obtained by sampling (r_a, h_a, r_b, h_b) at random conditioned on being consistent with one another and $(M, \mathcal{I})$.

The event $\mathsf{Good}(M, \mathcal{I})$ and the distribution $\mathcal{GEXEC}(M, \mathcal{I})$. The event $\mathsf{Good}(M, \mathcal{I})$ is defined as the event over $\mathcal{EXEC}(M, \mathcal{I})$ that all the intersection queries asked by Alice and Bob during the partial execution are in $\mathcal{I}$. More formally let $Q(A)$ (resp. $Q(B)$) be the set of queries asked by Alice (resp. Bob) which are specified by the view of Alice (resp. Bob) consisting of her private randomness, oracle answers, and the messages received till the moment specified by $(M, \mathcal{I})$. Therefore $\mathsf{Good}(M, \mathcal{I})$ is the same as $Q(A) \cap Q(B) \subset Q(\mathcal{I})$ where $Q(\mathcal{I})$ is the set of queries of $\mathcal{I}$ (note that $\mathcal{I}$ is a set of query/answser *pairs*). We define the distribution $\mathcal{GEXEC}(M, \mathcal{I})$ to be the distribution $\mathcal{EXEC}(M, \mathcal{I})$ conditioned on $\mathsf{Good}(M, \mathcal{I})$.

Eve's algorithm. The attacker Eve's algorithm is specified as follows. It is parameterized by some constant $\epsilon > 0$ which we assume is smaller than $1/10$. At any point in the execution, if M is the sequence of messages Eve observed so far and $\mathcal{I}$ is the query/answer pairs she learned so far, Eve computes for every $q \in \{0,1\}^\ell$ the probability p_q that q appears as a query in a random execution in $\mathcal{GEXEC}(M, \mathcal{I})$. If $p_q > \epsilon/n$ then Eve asks q from the oracle and adds q and its answer to $\mathcal{I}$. (If there is more than one such q then Eve asks the lexicographically first one.) Eve continues in this way until there is no additional query she can ask, at which point she waits until she gets new information (i.e., observes a new message sent between Alice and Bob).

Note that Eve's algorithm above may ask much more than n^2 queries. However, we will show that the probability that Eve asks more than n^2/ϵ^2 queries is bounded by $O(\epsilon)$, and hence we can stop Eve after asking this many queries without changing significantly her success probability.

Remark 3.2. The attacking algorithm above is not computationally efficient, as in general computing the probability distribution $\mathcal{GEXEC}(M, \mathcal{I})$ could be a hard problem since it involves "inverting" the algorithms of Alice and Bob to a certain extent. But because computing these probabilities in $\#\mathbf{P}$, we can use known techniques (e.g., [BGP]) to approximate them with arbitrarily good precision using an $\mathbf{NP}$-oracle. In particular this means that our attacker (as was the case in previous works) is computationally efficient in a relativized world in which $\mathbf{P} = \mathbf{NP}$, and hence this result also rules out *relativizing* reductions from one-way functions to key exchange that achieve $\omega(n^2)$ security.

4 Analysis of attack: proof of Theorem 3.1

For $i \in [2n]$, define the event Fail_i to be the event that the query made at the i^{th} round is an intersection query but is not contained in the set $\mathcal{I}$ of query/answer pairs known by Eve, and moreover that this is the first query satisfying this condition. Let the event $\mathsf{Fail} = \bigvee_i \mathsf{Fail}_i$ be the event that at some point an intersection query is missed by Eve, and let the event Long be that Eve makes more than n^2/ϵ^2 queries. By setting $\epsilon = \delta/100$ and stopping Eve after n^2/ϵ^2 queries, Theorem 3.1 immediately follows from the following two lemmas:

Lemma 4.1 (Attack is successful)**.** *For every i, $\Pr_{\mathcal{GEXEC}}[\mathsf{Fail}_i] \leq 10\epsilon/n$. Therefore by union bound we have $\Pr[\mathsf{Fail}] \leq 20\epsilon$.*

Lemma 4.2 (Attack is efficient)**.** $\Pr_{\mathcal{GEXEC}}[\mathsf{Long}] \leq 80\epsilon$.

4.1 Success of attack: proof of Lemma 4.1.

We now turn to proving Lemma 4.1. It will follow from the following stronger result:

Lemma 4.3. *Let i be even and let $B = (r_b, h_b)$ be some fixing of Bob's view in an execution up to the i^{th} message asked by him, and let $M, \mathcal{I}$ be some fixing of the messages exchanged and query/answer pairs learned by Eve in this execution such that $\Pr_{\mathcal{GEXEC}(M,\mathcal{I})}[\mathsf{Good}(M,\mathcal{I}) \mid B] > 0$. Then,*

$$\Pr_{\mathcal{GEXEC}(M,\mathcal{I})}[\mathsf{Fail}_i \mid B] \leq 10\epsilon/n \quad .$$

That is, the probability that Fail_i happens is at most $10\epsilon/n$ conditioning on Eve's information equalling $M, \mathcal{I}$, Bob's view of the execution equalling B and $\mathsf{Good}(M,\mathcal{I})$.

Proof of Lemma 4.1 from Lemma 4.3. Lemma 4.3 implies that in particular for every even i, $\Pr_{\mathcal{GEXEC}}[\mathsf{Fail}_i \mid \mathsf{Good}_i] \leq 10\epsilon/n$, where Good_i denotes the event $\mathsf{Good}(M,\mathcal{I})$ where $M, \mathcal{I}$ are Eve's information just before the i^{th} round. But since Fail_i is the event that Eve fails at round i *for the first time*, Fail_i implies Good_i and hence $\Pr_{\mathcal{GEXEC}}[\mathsf{Fail}_i] \leq \Pr_{\mathcal{GEXEC}}[\mathsf{Fail}_i \mid \mathsf{Good}_i]$, establishing the statement of Lemma 4.1 for every even i. By symmetry, the analog of Lemma 4.3 for odd i also holds with the roles of Alice and Bob reversed, completing the proof for all i. □

Proof outline of Lemma 4.3. Our approach to proving Lemma 4.3 is as follows:

1. We start by observing that the Lemma would be easy if the distribution $\mathcal{GEXEC}(M,\mathcal{I})$ would have been a *product distribution*, with the views of Alice and Bob independent from one another. Roughly speaking this is because in this case Bob has no more information than Eve on the queries Alice made in the past, and hence also from Bob's point of view, no query is more probable than ϵ/n to have been asked by Alice.

2. Unfortunately this is not the case. However, we can show that the distribution $\mathcal{GEXEC}(M,\mathcal{I})$ is equal to the distribution obtained by taking some product distribution $\mathcal{A} \times \mathcal{B}$ and conditioning it on the event $\mathsf{Good}(M,\mathcal{I})$. (A similar observation was made by [IR], see Lemma 6.5 there.)

3. This product characterization implies that we can think of $\mathcal{GEXEC}$ as a distribution over random edges of some bipartite graph G. Using some insights on the way this graph is defined, and the definition of our attacking algorithm, we will show that every vertex in G is connected to at least half of the vertices on the other side. We then show that this implies that Bob's chance of asking a query outside of $\mathcal{I}$ that was asked before by Alice is bounded by $O(\epsilon/n)$.

4.1.1 Product characterization of $\mathcal{GEXEC}(M,\mathcal{I})$

If $M,\mathcal{I}$ denote Eve's information just before the i^{th} round, then we know that conditioned on $M,\mathcal{I}$, no query has more than ϵ/n of being asked before by Alice, where this probability is taken over $\mathcal{GEXEC}(M,\mathcal{I})$. Hence if we could just show that from Bob's point of view Alice's distribution is distributed the same as she is from Eve's, we'd be done. Unfortunately, this is not true - there may be dependencies between Alice and Bob that are not captured by $M,\mathcal{I}$, even if $\mathsf{Good}(M,\mathcal{I})$ holds. Fortunately, we have a weaker statement - $\mathcal{GEXEC}(M,\mathcal{I})$ is equal to a product distribution conditioned on the event $\mathsf{Good}(M,\mathcal{I})$: (Note that the fact that $\mathcal{GEXEC}(M,\mathcal{I})$ is equal to a product distribution conditioned on some event is meaningless— every distribution has this property. Rather we use the fact that this condition is the particular event $\mathsf{Good}(M,\mathcal{I})$.)

Lemma 4.4 (Product characterization)**.** *For every $M,\mathcal{I}$ denoting Eve's information up to just before the i^{th} query, if $\Pr_{\mathcal{EXEC}(M,\mathcal{I})}[\mathsf{Good}(M,\mathcal{I})] > 0$ there exist a distribution $\mathcal{A}$ (resp. $\mathcal{B}$) over Alice's (resp. Bob's) computation up to that point such that*

$$\mathcal{GEXEC}(M,\mathcal{I}) = (\mathcal{A}\times\mathcal{B}) \mid \mathsf{Good}(M,\mathcal{I}) \quad .^{9} \tag{1}$$

Proof. We will show that for every pair of Alice/Bob executions (A,B) that satisfy the event $\mathsf{Good}(M,\mathcal{I})$, $\Pr_{\mathcal{GEXEC}(M,\mathcal{I})}[(A,B)] = c\alpha_A\alpha_B$ where α_A depends only on A, α_B depends only on B and c is a constant depending only on $M,\mathcal{I}$. This means that if we let $\mathcal{A}$ be the distribution such that $\Pr_{\mathcal{A}}[A]$ is proportional to α_A, and $\mathcal{B}$ be the distribution such that $\Pr_{\mathcal{B}}[B]$ is proportional to α_B, then $\mathcal{GEXEC}(M,\mathcal{I})$ is proportional (and hence equal to) the distribution $\mathcal{A}\times\mathcal{B} \mid \mathsf{Good}(M,\mathcal{I})$. (Note that if (A,B) do not satisfy $\mathsf{Good}(M,\mathcal{I})$ then $\Pr_{\mathcal{GEXEC}(M,\mathcal{I})}[(A,B)] = 0$.)

By definition,

$$\Pr_{\mathcal{GEXEC}(M,\mathcal{I})}[(A,B)] = \frac{\Pr_{\mathcal{EXEC}}[(A,B,M,\mathcal{I}) \text{ happen}]}{\Pr_{\mathcal{EXEC}}[(M,\mathcal{I}) \text{ happen } \wedge \mathsf{Good}(M,\mathcal{I})]}$$

The denominator of the righthand side is only dependent on M and $\mathcal{I}$. The numerator is equal to

$$2^{-|r_a|}2^{-|r_b|}2^{-\ell|Q(A)\cup Q(B)\cup Q(\mathcal{I})|}.$$

The reason is that the necessary and sufficient condition for getting $(A,B,M,\mathcal{I})$ in the system is that when we choose (r_a,r_b,H) to run the whole system we shall choose these specific random seeds r_a,r_b and we shall choose the answers specified in $(A,B,\mathcal{I})$ to the queries in $Q(A)\cup Q(B)\cup Q(\mathcal{I})$. The messages in M then will be generated by Alice and Bob correctly. Let $\alpha_A = 2^{-|r_a|}2^{-\ell|Q(A)\setminus Q(\mathcal{I})|}$ and $\beta_B = 2^{-|r_b|}2^{-\ell|Q(B)\setminus Q(\mathcal{I})|}$. Since $(Q(A)\setminus Q(\mathcal{I}))\cap(Q(B)\setminus Q(\mathcal{I})) = \emptyset$, the numerator is equal to $2^{-|r_a|}2^{-|r_b|}2^{-\ell|Q(A)\cup Q(B)\cup Q(\mathcal{I})|} = \alpha_A\beta_B 2^{-\ell|Q(\mathcal{I})|}$. Therefore $\Pr[(A,B) = \mathcal{GEXEC}(M,\mathcal{I})] = c(M,\mathcal{I})\alpha_A\beta_B$ where $c(M,\mathcal{I})$ only depends on $(M,\mathcal{I})$. □

4.1.2 Graph characterization of $\mathcal{GEXEC}(M,\mathcal{I})$

Fixing $M,\mathcal{I}$ that contain Eve's view up to just before the i^{th} round, define a bipartite graph $G = (V_L,V_R,E)$ as follows. Every node $u\in V_L$ will have a corresponding view A_u of Alice that is in the support of the distribution $\mathcal{A}$ obtained from Lemma 4.4; we let the number of nodes corresponding to a view A be proportional to $\Pr_{\mathcal{A}}[A]$, meaning that $\mathcal{A}$ corresponds to the uniform distribution over the left-side vertices V_L. Similarly, every node $v\in V_R$ will have a corresponding view of Bob B_v such that $\mathcal{B}$ corresponds to the uniform distribution over V_R. We define $Q_u = Q(A_u)\setminus Q(\mathcal{I})$ for $u\in V_L$ to be the set of queries *outside of* $\mathcal{I}$ that were asked by Alice in the view A_u, and define $Q_v = Q(B_u)\setminus Q(\mathcal{I})$ similarly. We put an edge in the graph between u and v (denoted by $u\sim v$) if and only if $Q_u\cap Q_v = \emptyset$. Lemma 4.4 implies that the distribution $\mathcal{GEXEC}(M,\mathcal{I})$ is equal to the distribution obtained by letting (u,v) be a random edge of the graph G and choosing (A_u,B_v).

[9] Note that the righthand side of (1) is a distribution over pairs of Alice's and Bob's view, while formally $\mathcal{GEXEC}(M,\mathcal{I})$ is a distribution over full executions that also contain Eve's view. However, since Eve's view in $\mathcal{GEXEC}(M,\mathcal{I})$ is always fixed to $(M,\mathcal{I})$, we can consider $\mathcal{GEXEC}(M,\mathcal{I})$ to be a distribution only over Alice's and Bob's views.

Note that because we assumed $\Pr_{\mathcal{GAECC}(M,\mathcal{I})}[\mathsf{Good}(M,\mathcal{I})] > 0$ this graph is nonempty. It turns out that this graph is *dense*:

Lemma 4.5. *Let $G = (V_L, V_R, E)$ be the graph above. Then for every $u \in V_L, d(u) \geq |V_R|(1-2\epsilon)$ and for every $v \in V_R$, $d(v) \geq |V_L|(1-2\epsilon)$ where $d(w)$ is the degree of the vertex w.*

Proof. We first show that for every $w \in V_L$, $\sum_{v \in V_R, w \not\sim v} d(v) \leq \epsilon|E|$. The reason is that the probability of vertex v being chosen when we choose a random edge is $\frac{d(v)}{|E|}$ and if $\sum_{w \not\sim v} \frac{d(v)}{|E|} > \epsilon$, it means that $\Pr_{(u,v) \leftarrow_{\mathrm{R}} E}[Q_w \cap Q_v \neq \emptyset] \geq \epsilon$. Hence because $|Q_w| \leq n$, by the pigeonhole principle there exists $a \in Q_w$ such that $\Pr_{(u,v) \leftarrow_{\mathrm{R}} E}[a \in Q_v] \geq \epsilon/n$. But this is a contradiction, because then a should be in $\mathcal{I}$ by the definition of the attack and cannot be in Q_w. The same argument shows that for every $w \in V_R$, $\sum_{u \in V_L, u \not\sim w} d(u) \leq \epsilon|E|$. So we proved that for any vertex w we have $|E^{\not\sim}(w) = \{(u,v) \in E \mid u \not\sim w \wedge w \not\sim v\}| \leq \epsilon|E|$, and $d(w) > 0$ for every $w \in V_L \cup V_R$. Now the following claim proves the lemma.

Claim 4.6. *Let $G = (V_L, V_R, E)$ be a nonempty bipartite graph such that for every vertex w, $|E^{\not\sim}(w)| \leq \epsilon|E|$ for $\epsilon \leq 1/2$, then for all $u \in V_L, d(u) \geq |V_R|(1-2\epsilon)$ and for every $v \in V_R$, $d(v) \geq |V_L|(1-2\epsilon)$.*

Proof. Let $d_L = \min\{d(u) \mid u \in V_L\}$ and $d_R = \min\{d(v) \mid v \in V_R\}$. Note we have $d_L > 0$ and $d_R > 0$. By switching the left and right sides if necessary, we may assume without loss of generality that **(*):** $\frac{d_L}{|V_R|} \leq \frac{d_R}{|V_L|}$. Thus it suffices to prove that $1 - 2\epsilon \leq \frac{d_L}{|V_R|}$. Suppose $1 - 2\epsilon > \frac{d_L}{|V_R|}$, and let $u \in V_L$ be the vertex that $d(u) = d_L < (1-2\epsilon)|V_R|$. Because for all $v \in V_R$ we have $d(v) \leq |V_L|$, therefore $|E^{\sim}(u)| \leq d_L|V_L| \leq d_R|V_R|$ (using **(*)**) where $E^{\sim}(u) = E \setminus E^{\not\sim}(u)$. On the other hand since we assumed $|\{v \in V_R \mid u \not\sim v\}| > 2\epsilon|V_R|$, we have $|E^{\not\sim}(u)| > 2\epsilon|V_R|d_R$. So $|E^{\sim}(u)| < |E^{\not\sim}(u)|/(2\epsilon)$, and therefore

$$|E^{\not\sim}(u)| \leq \epsilon\left(|E^{\not\sim}(u)| + |E^{\sim}(u)|\right) < \epsilon|E^{\not\sim}(u)| + |E^{\not\sim}(u)|/2,$$

which is a contradiction for $\epsilon < 1/2$. □

□

4.1.3 Proving Lemma 4.3

Now we can prove Lemma 4.3. Let $B, M, \mathcal{I}$ be as in Lemma 4.3 and q be Bob's query which is fixed now. By Lemma 4.4, the distribution $\mathcal{GAECC}(M, I)$ conditioned on getting B as Bob's view is the same as $(\mathcal{A} \times \mathcal{B})$ conditioned on $\mathsf{Good}(M,\mathcal{I}) \wedge (\mathcal{B} = B)$. By the definition of the bipartite graph $G = (V_L, V_R, E)$ it is the same as choosing a random edge $(u,v) \leftarrow_{\mathrm{R}} E$ conditioned on $B_v = B$ and choosing (A_u, B_v). We prove Lemma 4.3 even conditioned on fixing v such that $B_v = B$. Now the distribution on Alice's view is the same as choosing $u \leftarrow_{\mathrm{R}} N(v)$ to be a random neighbor of v and choosing A_u. Let $S = \{u \in V_L \mid q \in A_u\}$. Then we have:

$$\Pr_{u \leftarrow_{\mathrm{R}} N(v)}[q \in A_u] \leq \frac{|S|}{d(v)} \leq \frac{|S|}{(1-2\epsilon)|V_L|} \leq \frac{|S||V_R|}{(1-2\epsilon)|E|} \leq \frac{\sum_{u \in S} d(u)}{(1-2\epsilon)^2|E|} \leq \frac{\epsilon}{(1-2\epsilon)^2 n} < \frac{10\epsilon}{n}$$

The second and fourth inequalities are because of Lemma 4.5. The third one is because $|E| \leq |V_L||V_R|$. The fifth one is because of the definition of the attack which asks ϵ/n heavy queries, and the sixth one is because $\epsilon < 1/3$. □

4.2 Efficiency of attack: proof of Lemma 4.2

We call an event E defined over partial executions of $\mathcal{GAECC}$ *lasting* if whenever E holds for a partial execution, it holds for all extensions of it (i.e., partial executions that we get by continuing the experiment). For example the events Fail and Long are both lasting. For a lasting event E, let $\mathcal{GAECC}(\overline{E})$, be the same experiment as $\mathcal{GAECC}$ with the difference that we stop the execution as soon as E happens. Note that we have $\Pr_{\mathcal{GAECC}}[E \vee D] = \Pr_{\mathcal{GAECC}(\overline{E})}[E \vee D]$ for any lasting event D.

Proof outline of Lemma 4.2. The proof proceeds by the following two steps:

- If at any point during a partial execution of $\mathcal{EXEC}$, we get $\Pr_{\mathcal{EXEC}(M,\mathcal{I})}[\neg\mathsf{Good}(M,\mathcal{I})] > 1/2$, where $(M,\mathcal{I})$ are the current sequence of messages and Eve's set of query/answer pairs, we say that the event Bad holds for this partial execution and all extensions of this execution. Note that the event Bad is a lasting event. We will first use the success property of the attack $\Pr_{\mathcal{EXEC}}[\mathsf{Fail}] \leq 20\epsilon$ to show that $\Pr_{\mathcal{EXEC}}[\mathsf{Bad}] \leq 40\epsilon$ which means also $\Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Bad}] \leq 40\epsilon$.

- In the experiment $\mathcal{EXEC}(\overline{\mathsf{Bad}})$ whenever Eve asks a query q which is ϵ/n heavy for the distribution $\mathcal{GEXEC}(M,\mathcal{I})$, it is also $\gamma = \frac{\epsilon}{2n}$ heavy for $\mathcal{EXEC}(M,\mathcal{I})$ because $\Pr_{\mathcal{EXEC}(M,\mathcal{I})}[\mathsf{Good}(M,\mathcal{I})] \geq 1/2$. We will use this fact to show that in $\mathcal{EXEC}(\overline{\mathsf{Bad}})$ on average Eve will not ask more than $N = \frac{2n}{\gamma} = \frac{4n^2}{\epsilon}$ number of queries. Since Long is the event that Eve asks more than $\frac{n^2}{\epsilon^2} = \frac{N}{4\epsilon}$ queries, by Markov inequality we have $\Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Long}] \leq 4\epsilon$, and therefore we will have

$$\Pr_{\mathcal{EXEC}}[\mathsf{Long}] \leq \Pr_{\mathcal{EXEC}}[\mathsf{Long} \vee \mathsf{Bad}] = \Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Long} \vee \mathsf{Bad}] \leq \Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Long}] + \Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Bad}] \leq 44\epsilon$$

.

4.2.1 Step 1: Bounding $\Pr_{\mathcal{EXEC}}[\mathsf{Bad}]$

Note that $\neg\mathsf{Good}(M,\mathcal{I})$ implies that Fail_i has already happened for some i, and so $\neg\mathsf{Good}(M,\mathcal{I})$ implies Fail. The following lemma is implied by Lemma 6.4 in [IR], but we give a proof here for sake of completeness.

Lemma 4.7. $\Pr_{\mathcal{EXEC}}[\mathsf{Bad}] \leq 40\epsilon$.

Proof. Let's assume $\Pr_{\mathcal{EXEC}}[\mathsf{Bad}] > 40\epsilon$, and we will prove $\Pr_{\mathcal{EXEC}}[\mathsf{Fail}] > 20\epsilon$ which is a contradiction. When we run the system and the attack, instead of choosing the whole randomness (for Alice, Bob, and the oracle) at the beginning, we can choose some parts of the system first (according to their distribution in the original experiment), and then choose the rest of it from their distribution conditioned on the known parts. The lazy evaluation of the oracle answers is a special case of this general method. Therefore we can do as follows:

1. Run the system till an arbitrary point to get $(M,\mathcal{I})$ as Eve's information about the system. We pretend that till this point, we have just sampled (M,I), and the rest of the system's description is not chosen yet.
2. Choose the "true" view of Alice and Bob till this point from their distribution $\mathcal{EXEC}(M,\mathcal{I})$.
3. Continue running the system conditioned on the views of Alice, Bob, and Eve so far.

The moment that we sample Alice and Bob's views in the second step of the mentioned method is arbitrary, and we can choose this point to be the moment that Bad happens (if it happens at all). In other words we run the game till the moment that Bad happens: $\Pr_{\mathcal{EXEC}(M,\mathcal{I})}[\neg\mathsf{Good}(M,\mathcal{I})] > 1/2$, and then will choose Alice and Bob's computation so far, and then continue the game. But if Bad never happens we sample Alice and Bob's computation just at the end.

Since $\neg\mathsf{Good}(M,\mathcal{I}) \subset \mathsf{Fail}$ when $\Pr_{\mathcal{EXEC}(M,\mathcal{I})}[\neg\mathsf{Good}(M,\mathcal{I})] > 1/2$ happens for the first time in our execution of the system and we choose Alice and Bob's previous computation, Fail will hold for this running of the system with probability at least $1/2$. So, if $\Pr_{\mathcal{EXEC}}[\mathsf{Bad}] > 40\epsilon$, then with probability at least $(40\epsilon)\frac{1}{2} = 20\epsilon$ the event Fail holds in the system which is not possible. □

4.2.2 Step 2: Bounding $\Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Long}]$

Let $\gamma = \frac{\epsilon}{2n}, N = \frac{2n}{\gamma} = \frac{4n^2}{\epsilon}$ be as defined above. The following lemma shows that $\Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Long}] \leq 4\epsilon$ where Long is the event of asking more than $\frac{N}{4\epsilon} = \frac{n^2}{\epsilon^2}$ number of queries.

Lemma 4.8. *The expected number of queries asked by Eve in $\mathcal{EXEC}(\overline{\mathsf{Bad}})$ is at most $N = \frac{2n}{\gamma} = \frac{4n^2}{\epsilon}$.*

Proof. By definition whenever Eve asks a query, it is ϵ/n heavy in the distribution $\mathcal{GEXEC}(M,\mathcal{I})$, and since we always have $\Pr_{\mathcal{EXEC}(M,\mathcal{I})}[\mathsf{Good}(M,\mathcal{I})] > \frac{1}{2}$ in $\mathcal{EXEC}(\overline{\mathsf{Bad}})$ (whenever Eve is asking a query) therefore we have:

$$\Pr_{(A,B)\leftarrow_{\mathrm{R}}\mathcal{EXEC}(M,\mathcal{I})}[q \in Q(A)\cup Q(B)] \geq \Pr_{\mathcal{EXEC}(M,\mathcal{I})}[\mathsf{Good}(M,\mathcal{I})] \Pr_{(A,B)\leftarrow_{\mathrm{R}}\mathcal{GEXEC}(M,\mathcal{I})}[q \in Q(A)\cup Q(B)] \geq \frac{\epsilon}{2n} = \gamma$$

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

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

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

Note that $\sum_j p_j$ is the expected number of queries asked by Eve, and the claim implies that $\sum_j p_j \leq \frac{1}{\gamma}\sum \mathbb{E}[Y_j] \leq \frac{2n}{\gamma}$, 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 $\mathcal{W}_j = SUPP(\mathcal{O}_j) \setminus \{\perp\}$, and so we will have:

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

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

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

□

4.3 Removing the normal form assumption

In this section we show how to get an attack of the same $O(n^2/\delta^2)$ complexity finding all the intersection queries of Alice and Bob for a more general form of protocols. The proof has the following two steps:

1. We extend the result with the same complexity of $O(n^2/\delta^2)$ queries for the attack to the "seminormal" protocols by a bit more careful analysis of the same attack given above. A seminormal protocol is a protocol in which Alice and Bob can ask either zero or one query in each of their rounds. Again Alice and Bob ask at most n queries each, but the number of rounds R can be arbitrary larger than n.

2. Any protocol can be changed into a seminormal protocol without increasing n or loosing the security. Suppose i is a round in the original protocol in which Bob is going to ask $k \leq n$ number of queries (k is not known to Eve or Alice) and then send the message m_i to Alice. In the new protocol, this round will be divided into $2n-1$ sub-rounds. In the j^{th} sub-round (of this round) if j is even, Alice will just send the message $\perp$ to Bob. So let j is an odd number. If $j \leq k$, Bob will ask his j^{th} query which he was going to ask in the i^{th} round of the original protocol, and if $j > k$ he asks no query. If $j < 2n-1$, Bob sends also the message $\perp$ to Alice in the sub-round j, and if $j = 2n-1$ he sends his message m_i to Alice. It is clear that this artificial change only increases the number of rounds and will not give Eve any extra information, and therefore it is as secure as the original protocol. In the actual attack, Eve will pretend that Alice and Bob are sending the extra $\perp$ messages to each other in the sub-rounds and will attack the protocol in the seminormal form, and as we will prove she finds all the intersection queries with probability $1-\delta$ using $O(\frac{n^2}{\delta^2})$ number of queries.

Attack for seminormal protocols. Now we assume that Alice and Bob run a seminormal protocol in which each of them asks at most n number of oracle queries. We prove that the same attack of Section 3 finds all the intersection queries with probability $1-\delta$ using $O(\frac{n^2}{\delta^2}$ number of queries. We only show that the attack is successful in finding all the intersection queries and the same argument as before shows that the efficiency follows from the success property. Let BFail_i be the event that Bob's i^{th} query is the first intersection query out of $\mathcal{I}$, and similarly let AFail_i be the event that Alice's i^{th} query is the first intersection query out of $\mathcal{I}$.

Lemma 4.9. *For every $1 \leq j \leq n$, we have $\Pr[\mathsf{BFail}_i] \leq 10\epsilon/n$ and $\Pr[\mathsf{AFail}_i] \leq 10\epsilon/n$.*

Lemma 4.9 shows that $\Pr[\mathsf{Fail}] = \Pr[\bigvee_i(\mathsf{BFail}_i \vee \mathsf{AFail}_i)] \leq \sum_{1\leq i\leq n} \Pr[\mathsf{BFail}_i] + \Pr[\mathsf{AFail}_i] \leq 20n/\epsilon$. But Lemma 4.9 simply follows from Lemma 4.3 because Lemma 4.3 shows that $\Pr[\mathsf{BFail}_i] \leq 10\epsilon/n$ holds, even conditioned on a specific B describing Bob's view till the moment he is going to ask his i^{th} query. Note that the proof of Lemma 4.3 only used the fact that Alice and Bob ask at most n queries each and did not depend on the number of rounds.

5 Finding the secret

Now, we turn to the question of finding the secret. Theorem 6.2 in [IR] shows that once one finds all the intersection queries, with $O(n^2)$ more queries they can also find

the actual secret. Here we use the properties of our attack to show that we can do so even without asking more queries.

Theorem 5.1. *Assume that the total number of queries asked by Alice and Bob is at most n each, and their outputs agree with probability at least ρ having access to a random oracle. Then there is an adversary Eve asking at most $O(\frac{n^2}{\delta^2})$ number of queries such that Eve's output agrees with Bob's output with probability at least $\rho - \delta$.*

Proof. Let assume that in the last round of the protocol Alice sends a special message LAST to Bob. In order to find the secret Eve runs the attack of Section 3.1 [10] and at the end (when Alice has sent LAST and Eve has asked her queries from the oracle), Eve samples $(A,B) \leftarrow_{\text{R}} \mathcal{GEXEC}(M,\mathcal{I})$ (where $(M,\mathcal{I})$ is Eve's information at the moment) and outputs the secret $s(A)$ determined by Alice's view A. Now we prove that her secret agrees with Bob's secret with probability $\rho - O(\epsilon)$ and the theorem follows by setting $\delta = c\epsilon$ for sufficiently small constant c.

Let the random variables $\mathbf{A}, \mathbf{B}, \mathbf{E}$ be in order the view of Alice, Bob, and Eve at the end of the game. Let $\bar{\mathbf{A}}$ be the random variable generated by Sampling $(A,B) \leftarrow_{\text{R}} \mathcal{GEXEC}(M,\mathcal{I})$ where $M,\mathcal{I}$ are

[10] Again, we are interested in the case that the event $\mathsf{Fail} \vee \mathsf{Long}$ does not happen in the attack, and this is the case with probability at least $1 - O(\epsilon)$.

the information specified in $\mathbf{E}$ and choosing A from it. (So $s(\bar{(}A))$ is Eve's output.) We will prove that $\mathsf{SD}((\mathbf{A},\mathbf{B},\mathbf{E}),(\bar{\mathbf{A}},\mathbf{B},\mathbf{E})) \leq O(\epsilon)$. Then it shows that $|\Pr[s(\mathbf{A}) = s(\mathbf{B})] - \Pr[s(\bar{\mathbf{A}}) = s(\mathbf{B})]| \leq O(\epsilon)$. For $(A,B,E) \in SUPP(\mathbf{A}\times\mathbf{B}\times\mathbf{E}))$ we say the event $\mathsf{Good}(A,B,E)$ holds if A and B do not have any intersection query out of $\mathcal{I}$ where $(M,\mathcal{I}) = E$. The proof follows from the following three claims:

1. $\Pr[\neg\mathsf{Good}(\mathbf{A},\mathbf{B},\mathbf{E})] \leq O(\epsilon)$.
2. $\Pr[\neg\mathsf{Good}(\bar{\mathbf{A}},\mathbf{B},\mathbf{E})] \leq \epsilon$.
3. $\mathsf{SD}((\mathbf{A},\mathbf{B},\mathbf{E}) \mid \mathsf{Good}(\mathbf{A},\mathbf{B},\mathbf{E}),(\bar{\mathbf{A}},\mathbf{B},\mathbf{E}) \mid \mathsf{Good}(\bar{\mathbf{A}},\mathbf{B},\mathbf{E})) \leq 2\epsilon$.

The first claim follows from Theorem 3.1. The second claim is true because after fixing $\mathbf{E} = (M,\mathcal{I})$, the random variable $\bar{\mathbf{A}}$ is independent of $\mathbf{B}$, and if we fix $\mathbf{B} = B$ any query of $Q(B)$ has chace of at most ϵ/n of being in $Q(\bar{\mathbf{A}})$ and there are at most n such queries.

So we only need to prove the third claim which we do even for fixed $\mathbf{B} = B$ and fixed $\mathbf{E} = E = (M,\mathcal{I})$. As we will see, the claim basically follows from Lemma 4.5. Let $G = (V_L, V_R, D)$ be the graph characterization of $\mathcal{GEXEC}(M,\mathcal{I})$. Hence we have:

- The distribution of $\mathbf{A}$ in $(\mathbf{A},B,E) \mid \mathsf{Good}(\mathbf{A},B,E)$ is the same as choosing $v \in V_R$ such that $B_v = B$ (because all the vertices $\{v \mid B_v = B\}$ have the same set of neighbors) and then choosing a random neighbor of it $u \leftarrow_{\mathrm{R}} N(v)$ and getting A_u.
- The distribution of $\bar{\mathbf{A}}$ in $(\bar{\mathbf{A}},B,E) \mid \mathsf{Good}(\bar{\mathbf{A}},B,E)$ is the same as choosing $v \in V_R$ such that $B_v = B$ (because all of the vertices $\{v \mid B_v = B\}$ have the same set of neighbors) and then choosing a random edge $(u,v') \leftarrow_{\mathrm{R}} D$ conditioned on $u \sim v$ and then getting A_u. The last step (of choosing u) is the same as choosing $u \in N(v)$ with probabilities proportional to their degrees.

We show that the two above distributions have statistical distance at most 2ϵ even for a fixed v such that $B_v = B$. The first distribution chooses $u \in N(v)$ uniformly at random, but the second distribution chooses $u \in N(v)$ with probabilities proportional to their degrees. But since for every $u \in V_L$, $(1-2\epsilon)|V_R| \leq d(u) \leq |V_R|$, (and $\epsilon < 1/4$) one can easily show that the statistical distance is bounded by 2ϵ. □

Acknowledgements. We thank Russell Impagliazzo for useful discussions, and also for his warning that attempting to prove an $O(n^2)$ bound for this problem leads naturally to conjecturing (and even conjecturing that you proved) intermediate results that are simply not true. He was much more prescient than we realized at the time.

References

[BMG] B. Barak and M. Mahmoody-Ghidary. Merkle Puzzles are Optimal. *Arxiv preprint arXiv:0801.3669*, 2008. Preliminary version of this paper, contained a bug that is fixed in this version.

[BGP] M. Bellare, O. Goldreich, and E. Petrank. Uniform Generation of NP-Witnesses Using an NP-Oracle. *Inf. Comput*, 163(2):510–526, 2000.

[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.

[BGI] E. Biham, Y. J. Goren, and Y. Ishai. Basing Weak Public-Key Cryptography on Strong One-Way Functions. In R. Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2008.

[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. Merkle. Secure communications over insecure channels. *Communications of the 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.

[Sot] M. Sotakova. Breaking One-Round Key-Agreement Protocols in the Random Oracle Model. Cryptology ePrint Archive, Report 2008/053, 2008. http://eprint.iacr.org/.

A The error in the previous proof

The original paper [BMG] contained a wrong proof for Theorem 1.1. The idea of that proof was to describe the attack for another setting in which Alice and Bob will get independent oracle answers for the *same* query q if they ask it before Eve. The experiment was called Imag and the original experiment (which oracle answers are the same for everyone) was called Real. The point is that if we bound the probability of (the first) missing of an intersection query in either of Imag or Real it bounds that probability in the other setting as well, because before that event the experiments are the same. It was claimed in Lemma 5.5 of [BMG] that conditioned on Eve's information till some point, the distribution of Alice and Bob's computations before that point are independent, but it was not correct. Below we explain why.

Let $E = (M, \mathcal{I})$ be Eve's information and B describes Bob's computation, and we want to sample Alice's computation A (in Imag) conditioned on (E, r_b). Lemma 5.4 of [BMG] claimed correctly that the consistency of (E, A) is necessary and sufficient for the consistency of (E, A, B) (i.e. $SUPP(A \mid E) \times SUPP(B \mid E) = SUPP(A \times B \mid E)$), but it does not mean that r_a can be chosen *uniformly* at random, and the correct distribution actually depends on B.

One way to see why there is such dependency is to compute $\Pr[(A, B, M, \mathcal{I})$ happen in Imag experiment$]$ where Alice and Bob have no private intersection query: $Q(A) \cap Q(B) \subset \mathcal{I}$. Let r_a, r_b be the length of the (original) randomness of Alice and Bob which does not describe their oracle answers (In [BMG] the parties' randomness had the oracle answers as well). Now as opposed to what we had in Lemma 4.4 (of this paper) this probability is equal to: $\Pr[(A, B, M, \mathcal{I})$ happen in Imag experiment$] = 2^{-|r_a|}2^{-|r_b|}2^{-\ell|Q(A)\cup Q(B)\cup\mathcal{I}|}2^{-\ell k}$ where k is the number of intersection queries of Alice and Bob (according to $(A, B, M, \mathcal{I})$) which Eve has asked that query later at some point. The reason is that For such queries we have chosen the answer randomly at *two* points (i.e., when Alice asked it and when Bob asked it), and then when Eve asked it for the last time there was no coin tossing for getting the answer. It is different from the case for other queries which, say, Alice asked it first, Eve asked it second, and Bob asked it at last. In the latter case we only choose the random answer for Alice and no randomness is used later. The term $2^{-\ell k}$ makes the probability to be dependent on which order Alice and Bob ask their queries in the computation described by $(A, B, M, \mathcal{I})$. The positive (misleading) thing about $\Pr[(A, B, M, \mathcal{I})$ happen in Imag experiment$]$ is that when we do not have necessarily $Q(A) \cap Q(B) \subset \mathcal{I}$, a secret intersection query for Alice and Bob contributes $2^{-2\ell}$ to the probability which can be divided into two parts $2^{-\ell} \times 2^{-\ell}$ between Alice and Bob, but as we said for intersection queries that Eve also asks the query later the contribution of that query to the probability depends on the order that parties ask it.

Therefore, when we want to know the distribution of Alice in Imag conditioned on $(M, \mathcal{I})$, if q is asked in Alice's view A and also we have $q \in \mathcal{I}$, then A is less probable in the case Bob has asked q before Eve compared to the case that it is asked after Eve.