Merkle Puzzles Are Optimal — An $O(n^2)$-Query Attack on Any Key Exchange from a Random Oracle

Boaz Barak* and Mohammad Mahmoody-Ghidary**

Department of Computer Science, Princeton University
{boaz,mohammad}@cs.princeton.edu

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), and answers an open question posed by them. Our bound is optimal up to a constant factor since Merkle (CACM '78) gave a key exchange protocol that can easily be implemented in this model with n queries and 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 [1]). 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 Fig. 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 [4]. Recently, Biham, Goren and Ishai [3] took a step towards resolving this issue by providing a security analysis for Merkle's protocol under the concrete complexity assumption of existence of exponentially hard one-way functions. In particular, they proved that assuming there exist a one-way function that cannot be inverted with probability more than $2^{-\alpha n}$ by adversaries running in time $2^{\alpha n}$ for $\alpha \geq 1/2 - \delta$, there is a key exchange protocol in which Alice and Bob run in time n but any adversary whose running time is at most $n^{2-10\delta}$ has $o(1)$ chance of finding the

* Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

** Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797.

S. Halevi (Ed.): CRYPTO 2009, LNCS 5677, pp. 374–390, 2009.

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 $x_1, \ldots, x_{10n}$ in $[n^2]$ and sends $a_1, \ldots, a_{10n}$ 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 $y_1, \ldots, y_{10n}$ in $[n^2]$ and sends $b_1, \ldots, b_{10n}$ 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.

Fig. 1. Merkle's key exchange protocol [1][1]

secret. 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's work, Diffie and Hellman [5] and later Rivest, Shamir, and Adleman [6] 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 [8] 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

[1] 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. We remark that in Merkle's original protocol Bob will try different random queries $y_1, y_2, \ldots$ without sending them to Alice until he finds y_j such that $f(y_j) \in \{a_1, \ldots, a_{10n}\}$ and send j — the index of the "puzzle" a_j — to Alice. The Protocol of Fig. 1 is a *symmetric* version of Merkle's protocol, and is similar to the protocol of [2] in the bounded storage model; see also discussion in [3].

[2] This is not to be confused with some more recent works such as [7], that combine the random oracle model with assumptions on the intractability of other problems such as factoring or the RSA problem to obtain more efficient cryptographic constructions.

$O(n^6 \log n)$ queries.[3] 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.

Question and Motivation. Impagliazzo and Rudich [8, Sect. 8] mention as an open question (which they attribute to Merkle) to find out whether their attack can be improved to $O(n^2)$ queries (hence showing the optimality of Merkle's protocol in the random oracle model) or there exist key exchange protocols in the random oracle model with $\omega(n^2)$ security. Beyond just being a natural question, it also has some practical and theoretical motivations. The practical motivation is that 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.[4] In fact, as was argued by [3], as technology improves and honest users can afford to run more operations, such polynomial gaps only become more useful. Thus if known algebraic key exchange protocols were broken, one might look to polynomial-security protocol such as Merkle's for an alternative. Another motivation is theoretical— Merkle's protocol has very limited interaction (consisting of one round in which both parties simultaneously broadcast a message) and in particular it implies a public key encryption scheme. It is natural to ask whether more interaction can help achieve some polynomial advantage over this simple protocol. A third, less direct motivation comes from quantum computing. In one scenario in which some algebraic key exchange protocols will be broken— the construction of practical quantum computers— Merkle's protocol will also fail to offer non-trivial security due to Grover's search algorithm [9]. Our results below suggest (though do not prove) that Merkle's protocol may be optimal in this setting also, and so there may not exist a fully classical key-exchange protocol based on a one-way function with a black-box proof of super-linear security for quantum adversaries. We note that using quantum communication there is an *information theoretically* secure key-exchange protocol [10], and moreover, very recently Brassard and Salvail [11] (independently observed by [3]) gave a quantum version of Merkle's protocol, showing that if Alice and Bob can use quantum computation (but classical communication), to obtain a key-exchange

[3] More accurately, [8] 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 [8]'s results we will assume that $m = n$, though for our result we do not need this assumption.

[4] Of course, these numbers are just an example and in practical applications the constant terms will make an important difference. We note though that the above constants are not ruled out by [8]'s attack, but are ruled out by our attack (taking number of operations to mean the number of calls to the oracle).

protocol with super-linear (i.e., $n^{3/2}$) security in the random oracle model against quantum adversaries.

Our Result. In this work we answer the above question of [8], 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. *Let Π be a two-party protocol in the random oracle model such that when executing Π the two parties Alice and Bob make at most n queries each, and their outputs are identical with probability at least ρ. Then for every $0 < \delta < 1$, there is an adversary Eve making $(\frac{16n}{\delta})^2$ queries to the oracle whose output agrees with Bob's output with probability at least $\rho - \delta$.*

To the best of our knowledge, no better bound than the $\tilde{O}(n^6)$-query attack of [8] was previously known even in the 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 $\mathbf{P} = \mathbf{NP}$.

Correction of Error: A previous version of this manuscript [12] posted on the Arxiv claimed a different proof of the same result. However, we have found a bug in that proof— see the full version of this paper for more details. In fact the current proof is quite different from the one claimed in [12]. In [12] we also claimed an extension of Theorem 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 [8] 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 [8].

We also note that shortly after we posted the manuscript [12], Sotakova [13] 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 [8], 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

The main technical challenge in proving such a result is the issue of *dependence* between the executions of the two parties Alice and Bob in a key exchange

[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.

protocol. The presence of the random oracle allows Alice and Bob to correlate their executions even without communicating (which is indeed the reason that Merkle's protocol achieves non-trivial security). Dealing with such correlations is the cause of the technical complexity in both our work and the previous work of Impagliazzo and Rudich [8]. We handle this issue in a different way than [8]. On a very vague high level our approach can be viewed as using more information about the structure of these correlations than [8] did. This allows us to analyze a more efficient attacking algorithm, that is more frugal with the number of queries it uses than the attacker of [8]. Below we provide a more detailed (though still high level) exposition of our technique and its relation to [8]'s technique.

2.1 Comparison with [8]

We now review [8]'s attack and outline of analysis, and particularly the subtle issue of *dependence* between Alice and Bob that arises in both their work and ours. The main novelty of our work is the way we deal with this issue, which is different from the approach of [8]. We believe that this review of [8]'s analysis and the way it compares to ours can serve as a useful introduction to our actual proof. 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 Sect. 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. [8] 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). Even though 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 [8]'s case and ours to think of the attacker Eve as running concurrently with Alice and Bob. In particular, the attacker Eve of [8] 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 distribution $\mathcal{D}$ 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 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. Thus the bulk of [8]'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.*

2.2 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 [8]'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 [8] dealt with this issue by a "charging argument" (see also Remark 2 below), where they showed that 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 (and the rest of Eve's

[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 for such dependence 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. Bob then 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 $y' \neq y$. Still, an attacker must ask a non-intersection query such as 1^n to know if $x = 0^n$ or $x = 1^n$.

samples are distributed correctly as if the independence assumption was true). This argument inherently required sampling at least n executions (each of n queries) per round, hence resulting in an $\Omega(n^3)$ attack.

2.3 Our Approach

We now describe our approach and how it differs from the previous proof of [8]. 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 Sect. 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 [8]:

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 expected 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 [8]. Their attacker to some extent "pretended" that the conditional distributions of Alice and Bob are independent from one another. 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* with one another and Eve's current knowledge. 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 again, 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 most of 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] The definition of the graph G implies that $|S(u)| \leq n$ for all vertices u. The definition of our attacking algorithm implies that the distribution obtained by picking a random edge $\{u, v\}$ and outputting $S(u) \cup S(v)$ 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 show that these properties together imply that each vertex is connected to most of the vertices on the other side, and so G is close to being a complete bipartite graph.

Remark 2 (Comparison with [8]). One can also phrase the analysis of [8] in terms of a similar bipartite graph. Their argument involved fixing, say, Alice's execution which corresponds to fixing a left-side vertex u. As we noted above, if the degree of u is high (e.g., u is connected to most of the right side) then independence approximately holds and hence the probability that [8]'s attacker fails at this point is less than $1/(10n)$. The crucial component of [8]'s analysis was their observation 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 much more than n queries 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.

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

3 Our Attacker

We consider a key exchange protocol Π in which Alice and Bob first toss coins r_A and r_B and then run Π using 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 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 attacking algorithm that allows Eve to find a set of size $O(n^2)$ that contains all the queries asked by Alice and Bob in the random oracle model. This attack is analyzed in Sect. 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). As was shown by Impagliazzo and Rudich, it not hard to use this set to obtain 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.

Theorem 3. *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 $0 < \delta < 1$ there is an adversary Eve who has access to the messages sent between Alice and Bob and asks at most $(\frac{13n}{\delta})^2$ number of queries such that Eve's queries contain all the intersection queries of Alice and Bob with probability at least $1 - \delta$.*

Letting $\epsilon = \delta/13$, 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 5 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 $\mathcal{EXEC}$. A (full) *execution* of Alice, Bob, and Eve can be described by a tuple (r_A, r_B, H) where r_A denotes Alice's random tape, r_B denotes Bob's random tape, and H is the random oracle (note that Eve is deterministic). 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. 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). For any partial execution we denote by M the sequence of messages sent between Alice and Bob till that moment, and denote by I the set of oracle query/answer pairs known to Eve. We define *Alice's view* in the execution to be the tuple $A = (r_A, H_A, M)$ where r_A are Alice's coins and H_A is the concatenation of oracle answers to Alice's queries. Similarly Bob's view is the tuple $B = (r_B, H_B, M)$. Below we will only consider Alice's and Bob's view conditioned on a fixed value of M and hence we drop M from these tuples and let $A = (r_A, H_A)$ and $B = (r_B, H_B)$.

The Distribution $\mathcal{EXEC}(M, I)$. For $M = [m_1, \ldots, m_i]$ a sequence of i messages, and I a set of query/answer pairs, we denote by $\mathcal{EXEC}(M, I)$ the distribution over the views (A, B) of Alice and Bob in 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/answer pairs that Eve learns equals I. For every (M, I) that have nonzero probability to occur in the protocol, the distribution $\mathcal{EXEC}(M, I)$ can be sampled by first sampling (r_A, r_B, H) at random conditioned on being consistent with (M, I) and then deriving from this tuple Alice's and Bob's views: $A = (r_A, H_A)$ and $B = (r_B, H_B)$.[9]

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

Eve's Algorithm. The attacker Eve's algorithm is specified as follows. It is parameterized by some constant $0 < \epsilon < 1/10$. At any point in the execution, if

[9] Note that we can verify that the pair (M, I) has nonzero probability to occur in the protocol by simulating Eve's algorithm on the transcript M, checking that whenever Eve makes a query, this query is in I, in which case we feed Eve with the corresponding answer (and verifying at the end that there are no "extra" queries in I not asked by Eve). However in our attack the pair (M, I) will always be generated by running the actual protocol and so we won't need to run such checks.

M is the sequence of messages Eve observed so far and 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 $\mathcal{GEXEC}(M,I)$. If $p_q > \epsilon/n$ then Eve asks q from the oracle and adds q and its answer to 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.

4 Analysis of Attack: Proof of Theorem 3

We now go over the proof of Theorem 3. 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 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 $\epsilon = \delta/13$ and stopping Eve after n^2/ϵ^2 queries, Theorem 3 immediately follows from the following two lemmas:

Lemma 4 (Attack is successful). *For every i, $\Pr_{\mathcal{EXEC}}[\mathsf{Fail}_i] \leq \frac{3\epsilon}{2n}$. Therefore by the union bound, $\Pr[\mathsf{Fail}] \leq 3\epsilon$.*

Lemma 5 (Attack is efficient). $\Pr_{\mathcal{EXEC}}[\mathsf{Long}] \leq 10\epsilon$*.*

4.1 Success of Attack: Proof of Lemma 4

Lemma 4 follows from the following stronger result which is the main technical lemma of our paper:

Lemma 6. *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 sent by him, and let M, I be some fixing of the messages exchanged and query/answer pairs learned by Eve in this execution such that $\Pr_{\mathcal{EXEC}(M,I)}[\mathsf{Good}(M,I) \mid B] > 0$. Then it holds that $\Pr_{\mathcal{GEXEC}(M,I)}[\mathsf{Fail}_i \mid B] \leq \frac{3\epsilon}{2n}$. That is, the probability that Fail_i happens is at most $\frac{3\epsilon}{2n}$ conditioning on Eve's information equalling (M,I), Bob's view of the execution equalling B and $\mathsf{Good}(M,I)$.*

Proof (of Lemma 4 from Lemma 6.). Lemma 6 implies that in particular for every even i, $\Pr_{\mathcal{EXEC}}[\mathsf{Fail}_i \mid \mathsf{Good}_i] \leq \frac{3\epsilon}{2n}$, where Good_i denotes the event $\mathsf{Good}(M,I)$ where M, 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

$\Pr_{\mathcal{EXEC}}[\mathsf{Fail}_i] \le \Pr_{\mathcal{EXEC}}[\mathsf{Fail}_i \mid \mathsf{Good}_i]$, establishing the statement of Lemma 4 for every even i. By symmetry, the analog of Lemma 6 for odd i also holds with the roles of Alice and Bob reversed, completing the proof for all i.

Proof of Lemma 6

Product Characterization. Lemma 6 would be easy if the distribution $\mathcal{GEXEC}(M,I)$ 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. Unfortunately this is not the case. However, we can show that the distribution $\mathcal{GEXEC}(M,I)$ is equal to the distribution obtained by taking some product distribution $\mathcal{A} \times \mathcal{B}$ and conditioning it on the event $\mathsf{Good}(M,I)$. [10]

Lemma 7 (Product characterization). *For every M, I denoting Eve's information up to just before the i^{th} query, if $\Pr_{\mathcal{EXEC}(M,I)}[\mathsf{Good}(M,I)] > 0$ there exist a distribution $\mathcal{A}$ (resp. $\mathcal{B}$) over Alice's (resp. Bob's) view up to that point such that the distribution $\mathcal{GEXEC}(M,I)$ is the same as the product distribution $(\mathcal{A}\times\mathcal{B})$ conditioned on the event $\mathsf{Good}(M,I)$: $\mathcal{GEXEC}(M,I) = (\mathcal{A} \times \mathcal{B}) \mid \mathsf{Good}(M,I)$.*

Proof. We will show that for every pair of Alice/Bob views (A,B) in the probability space $\mathcal{EXEC}(M,I)$ that satisfy the event $\mathsf{Good}(M,I)$, $\Pr_{\mathcal{GEXEC}(M,I)}[(A,B)] = c(M,I)\alpha_A\alpha_B$ where α_A depends only on A, α_B depends only on B and $c(M,I)$ depends only on M, 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 $\mathcal{GEXEC}(M,I)$ is proportional (and hence equal to) the distribution $\mathcal{A} \times \mathcal{B} \mid \mathsf{Good}(M,I)$.

Because $(A,B) \in \mathrm{SUPP}(\mathcal{GEXEC}(M,I))$, if (A,B) happens, it makes the event $\mathsf{Good}(M,I)$ hold, and so we have

$$\begin{aligned}\Pr_{\mathcal{EXEC}(M,I)}[(A,B)] &= \Pr_{\mathcal{EXEC}(M,I)}[(A,B) \wedge \mathsf{Good}(M,I)] \\ &= \Pr_{\mathcal{EXEC}(M,I)}[\mathsf{Good}(M,I)] \Pr_{\mathcal{GEXEC}(M,I)}[(A,B)] \ .\end{aligned}$$

On the other hand, by definition we have $\Pr_{\mathcal{EXEC}(M,I)}[(A,B)] = \frac{\Pr_{\mathcal{EXEC}}[(A,B,M,I)]}{\Pr_{\mathcal{EXEC}}[(M,I)]}$, therefore it holds that $\Pr_{\mathcal{GEXEC}(M,I)}[(A,B)] = \frac{\Pr_{\mathcal{EXEC}}[(A,B,M,I)]}{\Pr_{\mathcal{EXEC}}[(M,I)]\Pr_{\mathcal{EXEC}(M,I)}[\mathsf{Good}(M,I)]}$. The denominator of the righthand side is only dependent on M and I. The numerator is equal to $2^{-|r_A|}2^{-|r_B|}2^{-\ell|Q(A)\cup Q(B)\cup Q(I)|}$. The reason is that the necessary and sufficient condition that $(A = (r_A, H_A), B = (r_B, H_B), M, I)$ happens is that when we choose an execution (r'_A, r'_B, H') then $r'_A = r_A$, $r'_B = r_B$ and H is consistent on the queries in $Q(A) \cup Q(B) \cup Q(I)$ with the answers

[10] A similar observation was made by [8], see Lemma 6.5 there.

specified in H_A, H_B, I. Note that this will ensure that Alice and Bob will indeed produce the transcript M. Let $\alpha_A = 2^{-|r_A|}2^{-\ell|Q(A)\setminus Q(I)|}$ and $\beta_B = 2^{-|r_B|}2^{-\ell|Q(B)\setminus Q(I)|}$. Since $(Q(A) \setminus Q(I)) \cap (Q(B) \setminus Q(I)) = \emptyset$, the numerator is equal to $2^{-|r_A|}2^{-|r_B|}2^{-\ell|Q(A)\cup Q(B)\cup Q(I)|} = \alpha_A\beta_B 2^{-\ell|Q(I)|}$. Thus indeed $\Pr_{\mathcal{GEXEC}(M,I)}[(A,B)] = c(M,I)\alpha_A\beta_B$ where $c(M,I)$ only depends on (M,I).

Graph Characterization. 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 most of the vertices on the other side. We then show that this implies that Bob's chance of asking a query outside of I that was asked before by Alice is bounded by $O(\epsilon/n)$.

More precisely, fixing M, 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 7; 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 $Q_u = Q(A_u) \setminus Q(I)$ for $u \in V_L$ to be the set of queries *outside of I* that were asked by Alice in the view A_u, and define $Q_v = Q(B_u) \setminus Q(I)$ 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 7 implies that the distribution $\mathcal{GEXEC}(M,I)$ is equal to the distribution obtained by letting (u,v) be a random edge of the graph G and choosing (A_u, B_v).

It turns out that this graph is *dense* (i.e., every vertex is connected to almost all other vertices in the other side). The proof has two steps. The first one is to show that such graphs are "highly connected" in the sense that removing any vertex v and its neighbors from the graph, remains a small fraction of the edges in the graph. The reason is that otherwise, there is a member of Q_v which is heavy and Eve should have asked that query. The second step is to show that this notion of connectivity would imply that the graph dense (whenever the graph is bipartite). More formally, we prove the following lemma:

Lemma 8. *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 $\sum_{v\in V_R, w\not\sim v}\frac{d(v)}{|E|} > \epsilon$, it means that $\Pr_{(u,v)\in_{\mathrm{R}} E}[Q_w \cap Q_v \neq \emptyset] \geq \epsilon$. Hence because $|Q_w| \leq n$, by the pigeonhole principle there exists $q \in Q_w$ such that $\Pr_{(u,v)\in_{\mathrm{R}} E}[q \in Q_v] \geq \epsilon/n$. But this is a contradiction, because then q should be in I by the definition of the attack and hence cannot be in Q_w. The same argument shows that for every $w \in V_R$, $\sum_{u\in V_L, u\not\sim w} d(u) \leq \epsilon|E|$. Thus for every vertex $w \in V_L \cup V_R$, $|E^{\not\sim}(w)| \leq \epsilon |E|$ where $E^{\not\sim}(w)$ denotes the set of edges that are not adjacent to any neighbor of w (i.e., $E^{\not\sim}(w) = \{(u,v) \in E \mid u \not\sim w \wedge w \not\sim v\}$). Now the following claim proves the lemma.

Claim. 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 $d_R = \min\{d(v) \mid v \in V_R\}$. 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 $d(v) \leq |V_L|$, thus using **(*)** we see that $|E^{\sim}(u)| \leq d_L|V_L| \leq d_R|V_R|$ where $E^{\sim}(u) = E \backslash E^{\not\sim}(u)$. On the other hand since we assumed that $d(u) < (1-2\epsilon)|V_R|$, there are more than $2\epsilon|V_R|d_R$ edges in $E^{\not\sim}(u)$, meaning that $|E^{\sim}(u)| < |E^{\not\sim}(u)|/(2\epsilon)$. But this implies

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

which is a contradiction for $\epsilon < 1/2$ because the graph G is nonempty.

Proof of Lemma 6 from Lemmas 7 and 8. Let B, M, I be as in Lemma 6 and q be Bob's query which is fixed now. By Lemma 7, the distribution $\mathcal{GEXEC}(M, I)$ conditioned on getting B as Bob's view is the same as $(\mathcal{A} \times \mathcal{B})$ conditioned on $\mathsf{Good}(M, I) \wedge (\mathcal{B} = B)$. By the definition of the bipartite graph $G = (V_L, V_R, E)$ it is the same as choosing a random edge $(u, v) \in_{\mathrm{R}} E$ conditioned on $B_v = B$ and choosing (A_u, B_v). We prove Lemma 6 even conditioned on fixing v such that $B_v = B$. Now the distribution on Alice's view is the same as choosing $u \in_{\mathrm{R}} N(v)$ to be a random neighbor of v and choosing A_u. Let $S = \{u \in V_L \mid q \in A_u\}$. Then it holds that

$$\Pr_{u \in_{\mathrm{R}} N(v)}[q \in A_u] \leq \frac{|S|}{d(v)} \leq \frac{|S|}{(1-2\epsilon)|V_L|} \leq \frac{|S||V_R|}{(1-2\epsilon)|E|} \leq \frac{\sum_{u \in S} d(u)}{(1-2\epsilon)^2|E|} \leq \frac{\epsilon}{(1-2\epsilon)^2 n} < \frac{3\epsilon}{2n} \ .$$

The second and fourth inequalities are because of Lemma 8. 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 $\epsilon = \delta/13 < 1/13$. □

4.2 Efficiency of Attack: Proof of Lemma 5

The proof of attack's efficiency (i.e. Lemma 5) crucially uses the fact that the attack is *successful*, and uses the following lemma from [8].

Lemma 9 (Lemma 6.4 of [8]). *Let* $Z_1, \ldots, Z_i, \ldots$ *be any sequence of random variables determined by a finite underlying random variable* X*, let* F *be any event for random variable* X*, and let* $0 \leq p \leq 1$*. Let* B_j *be the event that* $\Pr_X[F(X) \mid Z_1, \ldots, Z_j] \geq p$*, and let* $B = \bigvee_j B_j$*. Then it holds that* $\Pr_X[F[X] \mid B] \geq p$*.*

We say that a member of the probability space $X \in \mathcal{EXEC}$ is in the event Bad_j, if at the moment that Eve is go to ask her j^{th} query from the oracle, we have $\Pr_{\mathcal{EXEC}(M,I)}[\neg\mathsf{Good}(M,I)] > 1/2$, where (M,I) are the sequence of messages and Eve's set of query/answer pairs at that moment. Let $\mathsf{Bad} = \bigvee_j \mathsf{Bad}_j$. We define the probability space $\mathcal{EXEC}(\overline{\mathsf{Bad}})$ to be the same as $\mathcal{EXEC}$ with the difference that Eve stops asking more queries whenever Bad_j happens for some j.

The proof of efficiency consists of the following two steps.

Step 1. We first use the success property of the attack (i.e., $\Pr_{\mathcal{EXEC}}[\mathsf{Fail}] \leq 3\epsilon/n$) to show that $\Pr_{\mathcal{EXEC}}[\mathsf{Bad}] \leq 6\epsilon$ (Lemma 10 below) which also means that $\Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Bad}] = \Pr_{\mathcal{EXEC}}[\mathsf{Bad}] \leq 6\epsilon$. Note that $\neg\mathsf{Good}(M,I)$ implies that Fail_i has already happened for some i, and so $\neg\mathsf{Good}(M,I)$ implies Fail.

Step 2. We then show that in $\mathcal{EXEC}(\overline{\mathsf{Bad}})$ on average Eve will not ask more than $N = \frac{4n^2}{\epsilon}$ number of queries (see Lemma 11 below). 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

$$\begin{aligned}\Pr_{\mathcal{EXEC}}[\mathsf{Long}] \leq \Pr_{\mathcal{EXEC}}[\mathsf{Long} \vee \mathsf{Bad}] &= \Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Long} \vee \mathsf{Bad}] \\ &\leq \Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Long}] + \Pr_{\mathcal{EXEC}(\overline{\mathsf{Bad}})}[\mathsf{Bad}] \leq 10\epsilon \ .\end{aligned}$$

Now we prove the needed lemmas.

Lemma 10. $\Pr_{\mathcal{EXEC}}[\mathsf{Bad}] \leq 6\epsilon$.

Proof. We use Lemma 9 as follows. Let the underlying random variable be $X = \mathcal{EXEC}$, and the event $F = \mathsf{Fail}$. Let the random variable Z_j be the information that Eve learns about X after asking her $(j-1)^{\text{th}}$ query, before she asks her j^{th} query. Namely $(Z_1, \ldots, Z_j)$ is equal to (M,I) of the moment she wants to ask her j^{th} query. Let $p = 1/2$, which means B_j is the event that $\Pr[\mathsf{Fail} \mid Z_1, \ldots, Z_j] \geq 1/2$. Lemma 9 implies that $\Pr[\mathsf{Fail} \mid B] \geq 1/2$.

Note that $\neg\mathsf{Good}(M,I)$ at any moment implies that Fail has already happened, so Bad_j implies B_j and therefore Bad implies B. Now if $\Pr_{\mathcal{EXEC}}[\mathsf{Bad}] \geq 6\epsilon$, we would have $\Pr[\mathsf{Fail}] \geq \Pr[B \wedge \mathsf{Fail}] = \Pr[B]\Pr[\mathsf{Fail} \mid B] \geq \Pr[\mathsf{Bad}](\frac{1}{2}) \geq 3\epsilon$ which contradicts Lemma 4.

Lemma 11. *Let $\gamma = \frac{\epsilon}{2n}$, and $X = \mathcal{EXEC}(\overline{\mathsf{Bad}})$. If I denotes the set of query/answer pairs that Eve learns by the end of protocol in X, then $\mathsf{E}_X[|I|] \leq \frac{2n}{\gamma} = \frac{4n^2}{\epsilon}$.*

Proof. For a fixed query $q \in \{0,1\}^\ell$, let E_q (resp. F_q) be the event (over X) that Eve (resp. Alice or Bob) asks q. By linearity of expectation we have $\mathsf{E}[|I|] = \sum_q \Pr[E_q]$ and $\sum_q \Pr[F_q] \leq 2n$. We claim that $\Pr[E_q]\gamma \leq \Pr[F_q]$ which would imply the lemma $\mathsf{E}[|I|] = \sum_q \Pr[E_q] \leq \frac{1}{\gamma}\sum_q \Pr[F_q] \leq \frac{2n}{\gamma}$.

To prove $\Pr[E_q]\gamma \leq \Pr[F_q]$, we use Lemma 9 as follows. The underlying random variable $X = \mathcal{EXEC}(\overline{\mathsf{Bad}})$ (as here), the event $F = F_q$, and the random variable Z_j is as defined in the proof of Lemma 10. Let $p = \gamma$ which means B_j is the event that $\Pr[F_q \mid Z_1, \ldots, Z_j] \geq \gamma$. Lemma 9 implies that $\Pr[F_q \mid B] \geq \gamma$.

Note that if Eve asks q from the oracle when she knows $(M, I) = Z_1, \ldots, Z_j$ about X, q has at least ϵ/n probability to be asked by Alice or Bob conditioned on $\mathsf{Good}(M, I)$. But $\Pr[\mathsf{Good}(M, I)] \geq 1/2$ holds in X whenever Eve wants to ask a query, and it means that q is asked by Alice or Bob with probability at least $\frac{\epsilon}{2n} = \gamma$ before. In other words when Eve asks q it holds that $\Pr[F_q \mid Z_1, \ldots, Z_j] \geq \gamma$ which means that the event E_q implies B.

Therefore it holds that $\Pr[F_q] \geq \Pr[F_q \wedge B] = \Pr[B]\Pr[F_q \mid B] \geq \Pr[E_q]\gamma$.

5 Completing the Proof

To complete the proof, we need to **(a)** show how to handle protocols that are not necessarily in normal form and **(b)** show how Eve can recover the secret once she knows the intersection queries. Task **(b)** was already achieved by [8, Theorem 6.2] (although it can be shown that our attack does not need to ask any more queries to find the secret). [8] also showed how one can achieve task **(a)** using a general "compiler" that transforms general protocols to normal form. However that transformation has a quadratic blowup in efficiency that we cannot afford. We now sketch how our attack can be extended to handle general protocols without incurring this cost. (See the full version for the remaining details.)

In order to get an attack of the same $(\frac{13n}{\delta})^2$ complexity finding all the intersection queries of Alice and Bob for general form of protocols we do the following.

Attack for Seminormal Protocol. We first extend the result with the same complexity of $(\frac{13n}{\delta})^2$ 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 can be arbitrary larger than n.

Roughly speaking, the reason that the same attack as above works for seminormal protocols is that although there are $\Omega(n^2)$ number of rounds in the new seminormal protocol, we only need to bound the probability that Eve misses an intersection query for the first time whenever Alice or Bob does ask a query in their turn (and there are only $2n$ such queries). Assuming that it is Bob's turn, if we fix the query he asks we can still bound the probability that the query is the first missing intersection query using Lemma 6. That is because in Lemma 6 the statement holds even conditioned on Bob's computation (and his query in particular) being fixed.

Compiling into Seminormal Form. Any protocol can be simply changed into a seminormal protocol without increasing n or loosing the security. To see why, suppose it is Bob's turn and he is going to ask $k \leq n$ queries from the oracle before sending a message to Alice. Alice and Bob can blow this single round into $2n - 1$ number of rounds in which Alice does nothing other than sending $\perp$ to Bob and it lets Bob to ask his queries in different rounds. He sends the actual message in the last round among the $2n - 1$ new rounds. The total number of rounds will be $O(n^2)$, but the number of queries that Alice and Bob together ask will still remain $2n$ as before.

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

1. Merkle, R.: Secure communications over insecure channels. Communications of the ACM 21(4), 294–299 (1978)
2. Cachin, M.: Unconditional security against memory-bounded adversaries. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 292–306. Springer, Heidelberg (1997)
3. Biham, E., Goren, Y.J., Ishai, Y.: Basing weak public-key cryptography on strong one-way functions. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 55–72. Springer, Heidelberg (2008)
4. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: Proc. 30th STOC, pp. 209–218. ACM, New York (1998)
5. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory IT-22(6), 644–654 (1976)
6. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
7. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the First Annual Conference on Computer and Communications Security, November 1993, pp. 62–73. ACM, New York (1993)
8. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Proc. 21st STOC, pp. 44–61. ACM, New York (1989); Full version available from Russell Impagliazzo's home page
9. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Annual Symposium on Theory of Computing, May 22–24, pp. 212–219 (1996)
10. Bennett, C.H., Brassard, G., Ekert, A.: Quantum cryptography. SCIAM: Scientific American 267 (1992)
11. Brassard, G., Salvail, L.: Quantum merkle puzzles. In: International Conference on Quantum, Nano and Micro Technologies (ICQNM), pp. 76–79. IEEE Computer Society, Los Alamitos (2008)
12. Barak, B., Mahmoody-Ghidary, M.: Merkle Puzzles are Optimal. Arxiv preprint arXiv:0801.3669v1 (2008); Preliminary version of this paper. Version 1 contained a bug that is fixed in this version
13. Sotakova, M.: Breaking one-round key-agreement protocols in the random oracle model. Cryptology ePrint Archive, Report 2008/053 (2008), `http://eprint.iacr.org/`