Distributed Merkle's Puzzles

Itai Dinur* and Ben Hasson

Department of Computer Science, Ben-Gurion University, Israel

Abstract. Merkle's puzzles were proposed in 1974 by Ralph Merkle as a key agreement protocol between two players based on symmetric-key primitives. In order to agree on a secret key, each player makes T queries to a random function (oracle), while any eavesdropping adversary has to make $\Omega(T^2)$ queries to the random oracle in order to recover the key with high probability. The quadratic gap between the query complexity of the honest players and the eavesdropper was shown to be optimal by Barak and Mahmoody [CRYPTO'09].

We consider Merkle's puzzles in a distributed setting, where the goal is to allow *all* pairs among M honest players with access to a random oracle to agree on secret keys. We devise a protocol in this setting, where each player makes T queries to the random oracle and communicates at most T bits, while any adversary has to make $\Omega(M \cdot T^2)$ queries to the random oracle (up to logarithmic factors) in order to recover *any one* of the keys with high probability. Therefore, the amortized (per-player) complexity of achieving secure communication (for a fixed security level) decreases with the size of the network.

Finally, we prove that the gap of $T \cdot M$ between the query complexity of each honest player and the eavesdropper is optimal.

1 Introduction

In 1974 Merkle proposed a protocol that allows a pair of players to agree on a shared secret key without any secret shared in advance (the work was published in 1978 [17]). We describe an idealized variant of the protocol, assuming that player 1 (Alice), player 2 (Bob) and the adversary have access to a cryptographic hash function $H : [N] \rightarrow [N']$ (where $[N] = \{1, \dots, N\}$) that is hard to invert, modeled as a random function (oracle). Alice begins by selecting $\sqrt{N}$ elements in $[N]$ independently and uniformly at random $(x_1, \dots, x_{\sqrt{N}})$, and sends $(H(x_1), \dots, H(x_{\sqrt{N}}))$ to Bob. Then, Bob attempts to invert one of the elements by selecting $\sqrt{N}$ elements in $[N]$ independently and uniformly at random $(y_1, \dots, y_{\sqrt{N}})$, computing $(H(y_1), \dots, H(y_{\sqrt{N}}))$, and comparing with the hashed elements received from Alice. By a birthday paradox-like argument, with high probability, the query sets $\{x_1, \dots, x_{\sqrt{N}}\}$ and $\{y_1, \dots, y_{\sqrt{N}}\}$ intersect, namely, there exist i, j such that $x_i = y_j$. Thus, Bob sends i to Alice and the players

* This research was supported by the Israel Science Foundation (grants no. 573/16 and 1903/20). The first author was additionally supported by the European Research Council under the ERC starting grant agreement no. 757731 (LightCrypt).

agree on x_i as the shared secret key. The properties of H should guarantee that collisions (i.e., different inputs that hash to the same output) are unlikely inside query sets of this size, and thus the players agree on the same key with high probability. In terms of security, as H is a random oracle, an eavesdropping adversary has to query it on essentially the entire domain $[N]$ in order to recover x_i with high probability.

The quadratic gap between the query complexity of the honest players and the eavesdropper was shown to be optimal by Barak and Mahmoody [2,3] (tightening the previous bound of Impagliazzo and Rudich [13]), assuming the symmetric-key primitive is used as a black box. This stands in contrast to various key-agreement protocols (notably, the Diffie–Hellman protocol [7]) that achieve a super-polynomial gap between the complexity of the honest players and the eavesdropper, based on stronger assumptions which imply that public-key encryption schemes exist (refer to [1] for more details about such protocols). Clearly, the security of Merkle's puzzles is far from the ideal exponential security. However, Biham, Goren and Ishai [4] pointed out that it is not completely unacceptable, since the ratio between the work of the honest players and the adversary grows as technology advances and the honest players can afford more computation.

Key agreement protocols based on black-box use of symmetric-key primitives are still subject to active research. For example, the recent work [12] by Haitner et al. studied the communication complexity of such protocols. In this work we propose a distributed model for Merkle's puzzles and show that in this model the gap in query complexity between each honest player and the eavesdropper can be super-quadratic.

1.1 Distributed Key Agreement Based on Symmetric-Key Primitives

We study key agreement protocols in a generalized (distributed) model in which there are M honest players $p_1, \ldots, p_M$ that form a fully connected network.[1] The goal is to allow *all* pairs of players to agree on secret keys. We assume that all honest players and the eavesdropping adversary have access to a random oracle H. We measure the query and communication complexity of the players and the query complexity of the adversary. The problem can be easily solved if the players already have secure communication channels with a trusted party, which can use the channels to distribute all keys. However, in this work we do not assume any pre-existing secure channels.

Motivation. We do not expect our protocol to be used in practice for the purpose of key agreement, largely due to the small gap between the complexity of the honest players and the eavesdropper. However, we believe that the distributed

[1] Our protocol can also be made to work with small overhead in a sparse, but well-connected network such as the hypercube or the butterfly networks [18, Chapter 4.5].

model is a natural generalization of the basic problem of pairwise key agreement using symmetric-key primitives, and is worth studying. Moreover, techniques used in the protocol could potentially be useful in other settings as well. For example, they may be used to optimize key pre-distribution schemes in highly connected networks (see Section 1.4 for details about these schemes).

Basic protocol. In the most straightforward distributed protocol, each of the $\binom{M}{2}$ pairs of players independently carry out the standard 2-player Merkle's puzzles protocol. However, a closer examination reveals that this is wasteful and it is sufficient to form $O(M)$ secure links or edges (i.e., shared keys between player pairs) such that the secure communication graph is connected. Thus, in order for an arbitrary pair of players p_i, p_j to agree on a key, p_i chooses a key $k_{i,j}$ and sends it encrypted on a path to p_j in the secure link graph. Namely, if (ℓ, m) is a secure link in the graph, then p_ℓ sends $k_{i,j}$ to p_m encrypted with the key shared by p_ℓ and p_m. Player p_m decrypts $k_{i,j}$ and then sends it encrypted on the next secure link.

This protocol has the disadvantage that $k_{i,j}$ is not kept private from the other players (and is thus insecure in a model which does not assume all players are perfectly honest). It can be (partially) mitigated by p_i splitting $k_{i,j}$ into different secret shares, and sending the shares to p_j on non-intersecting paths.

In this improved protocol, it is sufficient for each player to agree on secret keys with $O(1)$ other players via standard Merkle's puzzles. Thus, every player makes $O(T)$ queries to H and an eavesdropping adversary has to make $\Omega(T^2)$ queries to recover any particular key with high probability. However, a key is now used to encrypt (shares of) other keys, and thus if the adversary is able to recover a few keys, the security of the entire network may collapse. Thus, we would like security guarantees against recovering *any one* of the keys with high probability. In order to achieve this, we can split the domain of H (assuming it is sufficiently large) among the different executions of Merkle's puzzles, such that they are completely independent.

The main question we consider in this work is whether the quadratic gap in query complexity in the distributed model (obtained by the basic protocol above) between the honest players and the eavesdropping adversary is optimal.

1.2 Our Results

We show that the quadratic gap obtained by the basic protocol in the distributed model is suboptimal.

Theorem 1 (informal). *For parameters M and T such that $T = \tilde{\Omega}(M)$,*[2] *there is a key agreement protocol based on symmetric-key primitives in the distributed model, where each honest player makes T queries to the random oracle*

[2] Throughout this paper, the notation $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ hide poly-logarithmic factors in T.

and communicates at most $\tilde{O}(T)$ bits, while any adversary has to make $\tilde{\Omega}(M \cdot T^2)$ queries to the random oracle in order to recover any one of the keys with high probability.

We further note that the computational complexity (in the standard word RAM model) of each honest player in our protocol is $\widetilde{O}(T)$. Consequently, up to small factors, a group of about 2^{20} players can communicate with 100-bit security after each player performs 2^{40} work. The complexity of the basic protocol above is 2^{50}, which is much higher.

More generally, if we fix the number of queries of the adversary (i.e., the security level of the protocol) to T_A, then the query and communication complexity of each player in our protocol is about $\sqrt{T_A/M}$. This gives the following (informal) property of our protocol.

Property 1. The complexity per player for securely connecting a network decreases with the size of the network.

This property may seem counterintuitive, as the number of targets (secure links) available to the adversary increases with the size of the network, so one may be tempted to conclude that each player must work at least as hard.

We also show that the gap of $T \cdot M$ obtained in our protocol between the query complexity of each player and the adversary is optimal (up to logarithmic factors). In fact, we show that this gap is the best possible even if we set a presumably weaker goal of establishing a single key between p_1 (or any other fixed player) and *any* other player p_j for $j \in [M] \backslash \{1\}$. In other words, we obtain the following property of the distributed model.

Property 2. The complexity per player for securely connecting p_1 to any one of the other players is essentially the same as for securely connecting the entire network.

Property 1 and Property 2 are due to a combination of the birthday paradox and properties of random graphs, as described next.

1.3 Overview of the Protocol and its Analysis

Setup protocol. Instead of trying to create pre-fixed secure links between pairs of players (as in the basic protocol described above), we start by creating arbitrary secure links based on a setup protocol via a distributed variant of Merkle's puzzles. Fixing the parameters T and M, every player selects T elements uniformly at random from $[N]$ (the domain of $H : [N] \to [N']$) and queries H to obtain the corresponding T images. If we choose $N \approx M \cdot T^2$, a birthday paradox-like argument shows that with high probability, the T elements chosen by any player p_i intersect the $(M-1) \cdot T \approx M \cdot T$ elements chosen by the other players. As in standard Merkle's puzzles, two players with intersecting query sets can agree on a shared key. However, it is not yet clear how the players can detect such intersections with limited communication.

One way to detect intersections is to have each player send its T query images to p_1 (or any designated player) that acts as an intermediate and informs all player pairs about the matches. However, this requires that p_1 communicates $\Omega(M \cdot T)$ bits. In order to get around this problem, we distribute the role of the intermediate among the different players: for each query $x \in [N]$, $H(x)$ is sent to player number $H(x) \bmod M$. This guarantees that each player receives about T images (with high probability), and can detect matches among them and then inform the corresponding players.

Choosing $N \approx M \cdot T^2 / \log M$, ensures that the secure network formed by the setup protocol is connected with high probability. However, in terms of security, an adversary may invert any one of the $\Omega(M)$ images (i.e., recover any one of the secret keys) and can succeed with high probability in doing so after making about $N/M \approx T^2/\log M$ queries. Therefore, we have not yet improved upon the basic protocol.

Amplification. In order to strengthen the security of the protocol, we perform amplification. The goal is to connect the network via "strong links" (keys) that the adversary has negligible probability (e.g., less than $2/N$) of recovering unless making (about) $N/2$ queries. For this purpose, for a (small) parameter L, we perform L independent executions of the setup protocol (with independent random oracles that can be derived by splitting the domain of H). Assume we wish to connect p_i and p_j by a strong link. Then, p_i selects $k_{i,j}$ (from a sufficiently large space), computes an L-out-of-L secret sharing of $k_{i,j}$ and sends the ℓ'th share on a path to p_j, encrypted using the keys of the ℓ'th execution. In terms of security, in order to recover $k_{i,j}$, the adversary has to recover one setup key on each of the L paths. For a fixed number of queries, the probability of the adversary to recover a setup key on a path depends on its length (which defines the number of targets). If the paths are too long then we need to select a large value of L to achieve the required security level, resulting in an inefficient protocol (in terms of both query and communication complexity).

Fortunately, the secure link graph formed by an execution of the setup protocol has diameter (i.e., maximal distance between two nodes) of $O(\log M)$ with high probability, and thus the paths are short. A similar phenomenon occurs in the $G(n,p)$ graph model [6] (in which each edge in the n-node graph is present independently with probability $p \approx \frac{\log n}{n}$). We note, however, that the edges of the secure link graph formed by the setup protocol are not independent.

Extension to the semi-honest model. Our basic protocol assumes that all players are perfectly honest. However, using similar techniques used for amplification, the protocol can be extended with logarithmic overhead to the semi-honest model (in which some players are honest but curious), where an adversary controls a fraction of $O(1/\log M)$ of the players.

Analysis. The main contribution of this work is proposing a distributed key agreement model based on symmetric-key primitives and devising a protocol in this model. On the other hand, the analysis of the protocol is elementary and mainly consists of basic concentration inequalities (it is easy to check that the protocol "works on average"). The proof of optimality follows by reduction from a 2-player protocol and is based on the result of Barak and Mahmoody [3]. Throughout the paper we aim for simplicity and make little effort to optimize low-order terms. In particular, it seems that a logarithmic improvement can be obtained by running the setup protocol only once with appropriate parameters, such that it is possible to select sufficiently many short disjoint paths in the secure link graph for the purpose of amplification. However, the analysis of such a protocol is substantially more complicated.

We chose to analyze our protocol in an idealized (information-theoretic) model as it simplifies the protocol and its analysis, and emphasizes its most important differences compared to previous works. An idealized model is also necessary for the proof of optimality. Alternatively, we could have investigated the minimal complexity-theoretic assumptions under which our protocol could be proven secure. Based on the analysis of [4] for 2-player protocols, it seems that we similarly need a one-way function of exponential strength and a "dream version" of Yao's XOR lemma [11]. We leave the formal treatment of this subject to future work.

1.4 Previous Work

Since Merkle's seminal work [17], various aspects of key agreement protocols based on symmetric-key primitives have been studied (c.f., [2,3,4,12,13]).

Key agreement protocols among a group of players have been investigated in numerous previous works, many of which make use of asymmetric-key primitives (c.f., [14]).

Various works also investigated the problem of key agreement among a group of players without using asymmetric-key primitives in models that are fundamentally different from ours. Among these we mention [16] by Leighton and Micali, that studied the problem in a model where keys are pre-assigned to players by a trusted dealer. Another example is [10] by Fischer and Wright, where it is assumed that the players have access to a particular type of correlated randomness (specifically, each player is given a secret set of cards that are not given to any other player).

The key agreement problem among a group of players is also related to secure message transmission (c.f., [8]), but our adversarial model is completely different and the relation is mostly indirect.

To the best of our knowledge, key agreement protocols among a group of players based on symmetric-key primitives have not been previously investigated in our (i.e., Merkle's) model, perhaps because it is not obvious that they offer any advantage compared to 2-player protocols. Below we elaborate on the line of work that seems to be the closest to ours (and is also related to [16]).

Random key pre-distribution schemes. In random key pre-distribution schemes each player (node) is initialized with a set of symmetric keys (chosen randomly from a group of keys, unknown to the adversary) prior to the key agreement protocol in order to bootstrap it. This model has been mostly studied in the context of sensor networks which have limited computational power (c.f., [5,9] and many followup works).

The random key pre-distribution model is related to ours, as our goal is also to connect a network via secure links using symmetric-key cryptography. However, there are important differences between the models, as in random key pre-distribution schemes, there is no random oracle (keys are pre-distributed) and the adversarial model allows the attacker to compromise nodes and discover their keys (but not to break cryptography). On the other hand, in our model the adversary may break the cryptography by querying the random oracle after eavesdropping. In addition, the network topology assumed in key pre-distribution scheme is different than ours and it has a substantial effect on the protocols.

To demonstrate the effect of the different models, note that in key pre-distribution schemes we can trivially establish a pre-shared key between any (fixed) pair of nodes, and the difficulty is in deploying a large-scale system with pre-shared keys where the adversary can compromise some of the nodes. Hence, properties 1 and 2 do not hold for these schemes. On the other hand, in our case, a larger network allows us to make use of its collective power to agree on keys with reduced amortized complexity, resulting in Property 1 (and indirectly, in Property 2).

Despite the different models and analysis, there are similarities between key pre-distribution protocols and our protocol. In particular, our setup protocol is analogous to the initial phase in key pre-distribution protocols, where each node discovers its neighbors by communicating identifiers of keys that it holds. However, the setup protocol of [5,9] is similar to the basic (undistributed) protocol we considered in which suboptimal parameters are selected (each pair of nodes share a common key with high probability). On the other hand, our advantage comes from the distributed variant of Merkle's puzzles in which each player shares a key only with a few other players not selected in advance. This allows to increase the key space (and the complexity of exhaustive search) by a factor of about M. Additionally, unlike [5,9], we match player couples (i.e., discover immediate neighbors in the secure link graph) via intermediate players in order to minimize communication.

The amplification we use is similar to the multipath-reinforcement protocol of [5] that strengthens the security of a link between two nodes by leveraging other secure links. However, we use paths of length about $\log M$, while [5] mainly uses paths of length 2, which are unlikely to exist in our case.

Open problems. An interesting open problem deals with an extended security model in which the goal of the adversary is to recover κ of the keys (where $\kappa \geq 1$ is an integer parameter). In our protocol, the adversary has to query the random oracle about $M \cdot T^2$ times in order to recover one key with high probability, yet roughly the same number of queries suffice for recovering all keys. We conjecture that this is essentially optimal, namely, in any protocol where the players agree on $\Omega(M)$ pairwise keys, the adversary can recover a constant fraction of them with $O(M \cdot T^2)$ queries.

Structure of paper. Next, we describe some preliminaries in Section 2 and then formally define our model in Section 3. Our setup and main protocols are described and analyzed in sections 4 and 5, respectively. In Section 6 we prove the optimality of our protocol with respect to query complexity. Finally, we discuss the extension to the semi-honest model and a communication-security tradeoff in Section 7.

2 Preliminaries

For numbers x and b, we denote by $\log x$, $\log_b x$ and $\ln x$ the logarithm of x with basis 2, b and e, respectively.

Given positive integers n, t, denote $[n] = \{1, \ldots, n\}$ and $[n]^t = \underbrace{[n] \times [n] \times \ldots \times [n]}_{t}$.

We will use the following inequalities. For every positive integer n, $n! > \left(\frac{n}{e}\right)^n$, while for every positive integers n, t (such that $n \geq t$),

$$\binom{n}{t} \leq \frac{n^t}{t!} < \left(\frac{e \cdot n}{t}\right)^t .$$

2.1 Graphs

Let $G = (V, E)$ be an undirected graph. The *distance* between two vertices $v, u \in V$ in G is the length of the shortest path between them. The *diameter* of G is the maximal distance between any two vertices of G.

The vertex v is a *neighbor* of u if $(v, u) \in E$. Let $U \subseteq V$. We define the *neighborhood* of U as $N_G(U) \triangleq \{v \in V \backslash U \mid v \text{ has neighbor in } U\}$.

We will use the notion of (vertex) expander graphs.

Definition 1 (Expander graphs). *Let $G = (V, E)$ be an undirected graph with n vertices and let $\delta > 0$. The graph G is a δ-expander if $|N_G(U)| \geq \delta \cdot |U|$ for every vertex subset $U \subset V$ with $|U| \leq n/2$.*

The following result is considered folklore (c.f., [15, Corollary 3.2]).

Proposition 1 (Diameter of expander graphs). *Let $G = (V, E)$ be an undirected graph with n vertices that is a δ-expander. Then, diam$(G) \leq 2\lceil \log_{1+\delta}(n/2) \rceil + 1 = O_\delta(\log n)$.*

Proof. Let $v \in V$. For an integer $t \geq 0$, denote by $B_t(v)$ the set of vertices within distance t from v in G. We prove by induction on t that

$$|B_t(v)| \geq \min(n/2, (1+\delta)^t).$$

For $t = 0$, we have $B_t(v) = \{v\}$ and $|B_t(v)| = 1$. For the induction step, assume that $|B_{t-1}(v)| \geq \min(n/2, (1+\delta)^{t-1})$ and note that $B_{t-1}(v) \subseteq B_t(v)$. If $|B_{t-1}(v)| \geq n/2$, we are done. Otherwise, $|B_{t-1}(v)| \geq (1+\delta)^{t-1}$ and $|B_{t-1}(v)| < n/2$. Denote $U = B_{t-1}(v)$. We have $B_t(v) = U \cup N_G(U)$ and $|N_G(U)| \geq \delta \cdot |U|$ since G is a δ-expander. Therefore, $|B_t(v)| \geq (1+\delta)|U| \geq (1+\delta)^t$ as claimed.

In particular, for $t = \lceil \log_{1+\delta}(n/2) \rceil$, for any $v, u \in V$ we have $B_t(v) \geq n/2$ and $B_t(u) \geq n/2$. Thus, $B_{t+1}(v) > n/2$ intersects $B_t(u)$, proving the result. ■

2.2 Random Functions and Encryption

A random function (oracle) can be thought of as an idealization of a cryptographic hash function. For positive integers N, N', a random function $H : [N] \to [N']$ is random variable, where for each $x \in [N]$, $H(x)$ is selected independently uniformly at random from $[N']$.

We also make use of an idealization of an encryption scheme using a random function. There are various ways to implement such an encryption scheme and we choose the following one that resembles the counter mode-of-operation: let $F : [N] \to [N']$ be a random function such that $N = N_1 \times N_2$ (i.e., we can write $F : [N_1] \times [N_2] \to [N']$). Given a key $k \in [N_1]$ and a counter $ct \in [N_2]$, a message $m \in [N']$ is encrypted as $F(k, ct) + m \bmod N'$. Decryption is performed by computing $F(k, ct)$ and subtracting it modulo N' from the ciphertext.

Assuming a pair (k, ct) is not reused to encrypt different messages and the adversary does not query F with the key k, then the scheme essentially acts as a one-time pad and no information is revealed about the encrypted messages from the ciphertexts and the values of F queried to the adversary.

3 Distributed Key Agreement Protocols Based on Random Oracles

We consider a complete network with M players $p_1, \ldots, p_M$ that have access to a random oracle H. The players run a protocol whose the goal is to establish keys between a fixed set of pairs of players $E_s \subseteq [M] \times [M]$. We do not assume a broadcast channel, and thus broadcasting a bit requires M bits of communication. We note that if a broadcast channel is assumed, then the communication restrictions in the protocols we devise are essentially trivial to satisfy.

All probabilities are computed with respect to the random oracle and the coin tosses of the players and adversary (whenever relevant).

Definition 2 (Distributed key agreement protocol). *A (M, α, T, β)-DKAP is a protocol between M players $p_1, \ldots, p_M$ with access to a random oracle H. Each player receives as input the same set of edges $E_s \subseteq [M] \times [M]$. For $i \in [M]$, denote the total number of queries of player p_i to H by T_i and the total communication of p_i by C_i. The protocol satisfies the following properties:*

- *For each $(i, j) \in E_s$, player p_i outputs $k_{i,j}$ and player p_j outputs $k_{j,i}$ such that $\Pr[\forall (i, j) \in E_s : k_{i,j} = k_{j,i}] \geq \alpha$.*
- *$\Pr[\forall i \in [M] : T_i \leq T] = 1$, and $\Pr[\forall i \in [M] : C_i \leq T] \geq \beta$.*

A variant of this definition places a worst-case upper bound on the communication complexity of each player. For sufficiently large α and β this variant is essentially equivalent to the one above, since a (M, α, T, β)-DKAP can easily be converted into a $(M, \alpha + \beta - 1, T, 1)$-DKAP: a player that exceeds the communication bound simply aborts and outputs a random value.

Another potential variant also places a bound of $\tilde{O}(T)$ on the total computation performed by each player (in some standard computational model). Our protocol satisfies this additional constraint.

As in standard Merkle's puzzles, security is defined with respect to a passive adversary that has access to the complete transcript of the protocol. The adversary makes a bounded number of queries to H and outputs a string of the form $((i, j), k)$. The adversary wins if $(i, j) \in E_s$ and $k = k_{i,j}$.

Definition 3 (Security of a distributed key agreement protocol). *A (M, α, T, β)-DKAP is (T_A, α_A)-secure if for any adversary $\mathcal{A}$ with access to the communication (transcript) of the protocol Λ that makes at most T_A queries to H, $\Pr[(i, j) \in E_s \wedge k = k_{i,j} \mid \mathcal{A}^H(\Lambda) \to ((i, j), k)] \leq \alpha_A$.*

The security definition does not restrict the keys on which the players agree. In particular, a protocol in which all players agree on the same key can potentially satisfy the definition. However, in our specific protocol the players agree on independent keys. This allows to easily extend it to the semi-honest model, as described in Section 7.

Supporting $\boldsymbol{E_s = [M] \times [M]}$. In general, the parameters of a key agreement protocol may depend on (be a function of) E_s. Ultimately, we would like to design a protocol that allows all pairs of players to exchange keys, namely, $E_s = [M] \times [M]$. However, as we outline below, a protocol for $E_s = [M] \times [M]$ can be easily obtained (with a small loss in parameters) from a protocol in which E_s is much sparser.

Specifically, assume we have a protocol that supports inputs E_s where $G = (V, E_s)$ is a sparse network with $|E_s| = \tilde{O}(M)$ for which there exist routing protocols with small congestion (such as the hypercube or the butterfly networks [18, Chapter 4.5]). Then, we can extend it to allow all $\binom{M}{2}$ pairs of players to agree on keys such that each player performs $\tilde{O}(M)$ additional encryptions (i.e. oracle queries) and communicates additional $\tilde{O}(M)$ bits almost surely: for each $(i, j) \in [M] \times [M]$ such that $i < j$, p_i picks a key $k'_{i,j}$ uniformly at random and sends it encrypted to p_j along a short path in (V, E_s).[3] If the exchanged keys are in a sufficiently large space (of size $\tilde{\Omega}(M \cdot T^2)$ in our case) and perfect encryption with domain separation is used (as described in Section 2), then recovering any $k'_{i,j}$ requires recovering at least one key in E_s and hence the advantage of an adversary (with a fixed upper bound on the number of queries) does not increase due to the additional key agreements. Therefore, we may restrict ourselves to designing distributed key agreement protocols in which $|E_s| = \tilde{O}(M)$.

4 The Setup Protocol

Algorithm 1 describes the setup protocol for player p_i (for any $i \in [M]$), assuming the M players have access to a random oracle $H : [N] \to [N']$. The protocol first establishes keys between various pairs of players and then propagates the information about which players share keys.

Parameter Selection. We assume for simplicity that M divides N'. We choose $N = \lfloor \frac{T^2 \cdot M}{25 \ln M} \rfloor$, $N' = T^6$ and $D = 4 \log M$. We further denote $R \triangleq \frac{T^2 \cdot M}{N}$.

We assume that $M \geq 64, T \geq 20000$ and note that $25 \ln M \leq R \leq 26 \ln M$. Moreover, we assume $M \leq T$, which is reasonable as otherwise, iterating over the list of players requires more than T time (and broadcasting a bit has communication complexity of M bits).

We now analyze the setup protocol with respect to correctness, query and communication complexity, connectivity of the secure link graph G and security.

4.1 Correctness

Proposition 2. *Assume that for all $(i, j) \in E_s$, p_i outputs $k_{i,j}$ player p_j outputs $k_{j,i}$. Then,* $\Pr[\forall (i, j) \in E_s : k_{i,j} = k_{j,i}] \geq 1 - T^{-2}$.

[3] This exposes $k'_{i,j}$ to the players along the path, and is therefore insecure in the semi-honest model. However, the protocol can be patched by secret sharing $k'_{i,j}$ and sending multiple shares encrypted along disjoint paths. We use a somewhat similar protocol for the purpose of amplification.

Algorithm 1: Setup protocol (p_i's algorithm)

```
Parameters: $M, T, N, N', D$
For all $j \in [M] \backslash \{i\}$, set $k_{i,j} = \bot$
Choose $(x_1, \ldots, x_T) \in [N]^T$ uniformly at random (with replacement)
Compute $(H(x_1), \ldots, H(x_T)) \in [N']^T$ and store the $T$ pairs $(x_j, H(x_j))$ in a
  table $\mathcal{T}_1$, sorted by the second column
For each $j \in [T]$, send $(i, H(x_j))$ to player number $H(x_j) \bmod M \in [M]$
Receive messages from other players: $(u_1, y_1), (u_2, y_2), \ldots$ and store them in a
  table $\mathcal{T}_2$, sorted by the second column
forall collisions in $\mathcal{T}_2$: $\{(u_j, y_j), (u_\ell, y_\ell) \mid y_j = y_\ell \wedge u_j \neq u_\ell\}$ do
    send $(u_\ell, y_j)$ to player number $u_j$
    send $(u_j, y_j)$ to player number $u_\ell$
Receive messages from other players: $(v_1, z_1), (v_2, z_2), \ldots$
For each message $(v_j, z_j)$, search for $z_j$ in $\mathcal{T}_1$. If there exists an entry
  $(x_\ell, H(x_\ell))$ in $\mathcal{T}_1$ such that $z_j = H(x_\ell)$, set
  $k_{i,v_j} = \begin{cases} x_\ell & \text{if } k_{i,v_j} = \bot \text{ or } H(k_{i,v_j}) < H(x_\ell), \\ k_{i,v_j} & \text{otherwise} \end{cases}$
▷ Distribute secure link graph
Broadcast the elements of the set $\{(i,j) \mid k_{i,j} \neq \bot \wedge i < j\}$
Receive and store messages $(f_1, g_1), (f_2, g_2), \ldots$ from other players
Construct a graph $G = (V, E)$, where $V = [M]$, $E = \{(f_1, g_1), (f_2, g_2), \ldots\}$
Run breadth-first search on $G$ from node $i$ and calculate the minimal distance
  to each $j \in [V]$. If there exists $j \in [V]$ whose distance from $i$ is larger than $D$,
  broadcast "fail" and output $\bot$.
If a "fail" message is received, then output $\bot$. Otherwise, output $G$ and
  $\{(j, k_{i,j}) \mid k_{i,j} \neq \bot\}$
```

Proof. Note that if the players output $\bot$ the protocol is still formally correct. Therefore, the only event that may cause a pair of players to output non-matching keys is that their joint query set contains a collision in H, namely a pair of elements $q_i, q_j \in [N]$ such that $H(q_i) = H(q_j)$ but $q_i \neq q_j$.

Based on the randomness of H, a pair of different queries collide with probability $1/N'$. By a union bound over all query pairs, the probability of a collision in the $M \cdot T$ queries made by the players is bounded by $\frac{(T \cdot M)^2}{N'} \leq \frac{T^4}{N'} = T^{-2}$. ∎

4.2 Query and Communication Complexity

Proposition 3. *Each player makes at most T queries to H and communicates $\tilde{O}(T)$ bits, except with probability at most $M \cdot 2^{-T} + (36 \log T \cdot T)^{-1} + T^{-2}$.*

Clearly, each player makes T queries to H. It remains to bound the communication complexity by $\tilde{O}(T)$. First, all the messages are in a space of size polynomial in T, hence the length of each message is $\tilde{O}(1)$ bits. Propositions 4 and 5 below bound the number of messages sent and received by each player. Given that G contains $\tilde{O}(T)$ edges (which is guaranteed with high probability by Proposition 5), then the communication of all players for propagating the edges is bounded by $\tilde{O}(T)$. Therefore, it remains to prove propositions 4 and 5 in order to complete the proof of Proposition 3.

Proposition 4. *In lines 4-5 of the setup protocol, each player communicates at most $8T$ messages, except with probability at most $M \cdot 2^{-T}$.*

Proposition 5. *In lines 6-9 of the setup protocol, all players (collectively) communicate at most $130 \log T \cdot T$ messages, except with probability at most $(36 \log T \cdot T)^{-1} + T^{-2}$.*

The probability bound in Proposition 5 is rather loose, but it is sufficient for our purpose.

Proof (of Proposition 4). In Line 4, each player sends at most T messages. It remains to analyze the number of messages each player receives in Line 5.

The number of received messages by p_i is determined by the number of images of H computed by the M players that are equal to i modulo M. As we assume that M divides N' and each image of H is uniform in $[N']$, the probability that each query to H results in a message sent to p_i is $1/M$.

Overall, the players make $M \cdot T$ queries to H, each is uniform in $[N]$. We order them arbitrarily and denote them by $q_1, \ldots, q_{M \cdot T}$. The query q_ℓ results in a message to p_i if $H(q_{j_\ell}) \bmod M = i$ and we bound the probability that this happens for many queries below.

Claim. Consider any ordered subset of $7T$ queries $q_{j_1}, \ldots, q_{j_{7T}}$. Then,

$$\Pr[\forall \ell \in [7T] : H(q_{j_\ell}) \bmod M = i] < \left(\tfrac{2}{M}\right)^{7T}.$$

Proof. For some positive integer $r < 7T$, assume that $H(q_{j_r}) \bmod M = i$ for all $\ell \in [r]$. Then, $H(q_{j_{r+1}}) \bmod M = i$ holds if either $q_{j_{r+1}} \in \{q_{j_1}, \ldots, q_{j_r}\}$ (which occurs with probability at most r/N), or $q_{j_{r+1}} \notin \{q_{j_1}, \ldots, q_{j_r}\}$ and $H(q_{j_{r+1}}) \bmod M = i$ (which occurs with probability at most $1/M$). Therefore $H(q_{j_{r+1}}) \bmod M = i$ holds with probability at most

$$\tfrac{1}{M} + \tfrac{r}{N} \leq \tfrac{1}{M} + \tfrac{7T}{N} < \tfrac{1}{M} + \tfrac{1}{M} = \tfrac{2}{M},$$

as $N = \frac{M \cdot T^2}{R} \geq 7M \cdot T$ (given that $T \geq 20000$). The claim follows by induction on r. □

There are

$$\tbinom{M \cdot T}{7T} \leq \left(\tfrac{e \cdot M \cdot T}{7T}\right)^{7T} = \left(\tfrac{e \cdot M}{7}\right)^{7T}$$

different query subsets of size $7T$. By a union bound over all of them, the probability that at least $7T$ messages are sent to p_i in Line 5 is at most

$$\left(\tfrac{e\cdot M}{7}\right)^{7T} \cdot \left(\tfrac{2}{M}\right)^{7T} < 2^{-T}.$$

The results follows by a union bound over all M players. ■

Proof (of Proposition 5). The number of messages sent in lines 6-8 and received in Line 9 is upper bounded by twice the number of collisions in the tables of the players. Consider the queries made by the players in arbitrarily order $q_1, \ldots, q_{M\cdot T}$. We will make a distinction between two types of collisions. A collision in H was shown in Proposition 3 to occur with probability at most T^{-2}. We assume such a collusion does not occur and use a union bound to obtain the final result. A query collision occurs if $q_j = q_\ell$ for $j \neq \ell$ and it results in a shared key (assuming the queries are issued by different players).

We denote the total number of query collisions by Col and bound $\Pr\left[Col \geq 65 \log T \cdot T\right]$ to finish the proof.

For all $j, \ell \in [M \cdot T]$ such that $j \neq \ell$, define an indicator random variable $C_{j,\ell}$ that is equal to 1 if $q_j = q_\ell$. We have

$$\begin{aligned}&\mathrm{E}[C_{j,\ell}] = \Pr[C_{j,\ell} = 1] = N^{-1}, \text{ and}\\ &\mathrm{Var}[C_{j,\ell}] = \mathrm{E}[(C_{j,\ell})^2] - (\mathrm{E}[C_{j,\ell}])^2 = N^{-1} - N^{-2} < N^{-1}.\end{aligned}$$

Hence,

$$\mathrm{E}[Col] = \mathrm{E}\left[\sum_{j,\ell} C_{j,\ell}\right] = \sum_{j,\ell} \mathrm{E}[C_{j,\ell}] < \tfrac{(M\cdot T)^2}{N} = R \cdot M.$$

Note that the random variables $\{C_{j,\ell}\}$ are pairwise independent. Hence,

$$\mathrm{Var}[Col] = \mathrm{Var}\left[\sum_{j,\ell} C_{j,\ell}\right] = \sum_{j,\ell} \mathrm{Var}[C_{j,\ell}] < \tfrac{(M\cdot T)^2}{N} = R \cdot M.$$

For a parameter $c > 0$, Chebyshev's inequality gives

$$\Pr\left[Col - \mathrm{E}[Col] \geq c \cdot \sqrt{\mathrm{Var}[Col]}\right] \leq c^{-2}.$$

Recalling that $T \geq M$ and $25 \log M \leq R \leq 26 \log M$, we obtain

$$\begin{aligned}&\Pr\left[Col \geq 65 \log T \cdot T\right] \leq \Pr\left[Col - R \cdot M \geq 39 \log T \cdot T\right] \leq\\ &\Pr\left[Col - R \cdot M \geq 6\sqrt{\log T \cdot T} \cdot \sqrt{R \cdot M}\right] \leq (36 \log T \cdot T)^{-1},\end{aligned}$$

as required. ■

4.3 Connectivity

We prove that the secure link graph formed by the setup protocol is a good expander with high probability, and therefore it has small diameter.

Let U be a group of players of size $k > 0$. We call U *useful* if the players in U make at least $T \cdot k/2$ distinct queries to H.

Proposition 6. *Any group of players is useful, except with probability at most* 2^{-2T}.

Proof. Fix a group U of size k. There are $k \cdot T$ queries made by the players in U. Consider them in some order. We call the j'th query useful if it does not collide with the previous $j-1$ queries (and not useful otherwise). For each $j \in [k \cdot T]$, the probability that query number j is not useful is at most $\frac{k \cdot T}{N}$.

Consider an arbitrary subset of $\frac{k \cdot T}{2}$ queries made by players in U. The probability that they are all not useful is at most $\left(\frac{k \cdot T}{N}\right)^{(k \cdot T)/2}$. Taking a union bound over all such sets (whose number is less than $2^{k \cdot T}$), the probability that there is a set of size $k \cdot T/2$ of non-useful queries is at most $2^{k \cdot T} \cdot \left(\frac{k \cdot T}{N}\right)^{k \cdot T/2} = \left(\frac{4 \cdot k \cdot T}{N}\right)^{k \cdot T/2} \leq 2^{-2T}$, given that $N = \frac{M \cdot T^2}{R} \geq 64M \cdot T$ (as $T \geq 20000$). Hence U is useful, except with probability at most 2^{-2T}.

■

Proposition 7. *Consider the secure link graph $G = (V, E)$ formed by the setup protocol. Let $U \subset V$ be a set of size k for $1 \leq k \leq M/2$. Then,*

$$\Pr[|N_G(U)| \leq \tfrac{k}{2}] \leq e^{-R \cdot k/12} + 2^{-2T}.$$

Proof. We first prove that

$$\Pr[|N_G(U)| \leq \tfrac{k}{2} \mid U \text{ is useful}] \leq e^{-R \cdot k/12} \tag{1}$$

Combined with Proposition 6, this implies

$$\Pr[|N_G(U)| \leq \tfrac{k}{2}] \leq \\ \Pr[|N_G(U)| \leq \tfrac{k}{2} \mid U \text{ is useful}] + \Pr[U \text{ is not useful}] \leq e^{-R \cdot k/12} + 2^{-2T},$$

as required.

We now prove (1). Given that U is useful, we fix a set Q of $T \cdot k/2$ distinct queries made by the players in this group.

Note that if $|N_G(U)| \leq \frac{k}{2}$ then there exists a set $V' \subseteq V \backslash U$ of size at least $M - k - \frac{k}{2} = M - 3k/2 \geq M/4$ such that $V' \cap N_G(U) = \emptyset$. Hence the intersection of the queries of the players in V' (whose number is at least $T \cdot M/4$) with Q is empty. The probability of a query hitting Q is $|Q|/N$. Since all the $T \cdot (M - 3k/2)$ queries are independent, the probability none of them hits Q is at most

$$\left(1 - \tfrac{|Q|}{N}\right)^{T \cdot M/4} \leq e^{-|Q| \cdot T \cdot M/4N} = e^{-T^2 \cdot k \cdot M/8N} = e^{-R \cdot k/8}.$$

where for the inequality we have used in inequality $1 - x \leq e^{-x}$ (which holds for any real x).

The number of sets $V' \subseteq V \backslash U$ of size $M - 3k/2$ is

$$\tbinom{M-k}{M-3k/2} = \tbinom{M-k}{k/2} \leq \tbinom{M}{k/2} \leq \left(\tfrac{2eM}{k}\right)^{k/2} = e^{k(1+\ln 2+\ln M-\ln k)/2}.$$

Taking a union bound over all of them, we conclude

$$\Pr[|N_G(U)| \leq \tfrac{k}{2} \mid U \text{ is useful}] \leq e^{-R \cdot k/8 + k(1+\ln 2+\ln M-\ln k)/2} \leq$$
$$e^{-k(R/8-1-\ln M/2)} \leq e^{-R \cdot k/12},$$

where the last inequality follows since $R \geq 25 \ln M$. ■

Proposition 8. *The secure link graph $G = (V, E)$ formed by the setup protocol satisfies* $\Pr[diam(G) > 4 \log M] \leq 2e \cdot M^{-1}$.

Proof. We show that G is a δ-expander for $\delta = 1/2$, except with probability at most $2e \cdot M^{-1}$. Then, by Proposition 1, $diam(G) \leq 2\lceil \log_{3/2}(M/2) \rceil + 1 \leq 2 \log_{3/2}(M/2) + 3 \leq 3.42 \log_2 M + 3 \leq 4 \log M$ (as $M \geq 64$).

Let $U \subset V$ be of size $k \leq M/2$. By Proposition 7, $|N_G(U)| > k/2$ except with probability at most $e^{-R \cdot k/12} + 2^{-2T}$. Taking union bound over all subsets of size k, whose number is $\binom{M}{k} \leq \left(\frac{eM}{k}\right)^k = e^{k(\ln M+1-\ln k)}$, we conclude that for all of them $|N_G(U)| > k/2$, except with probability at most

$$\begin{aligned} e^{k(\ln M+1-\ln k-R/12)} + \tbinom{M}{k} 2^{-2T} &\leq \\ e^{k(\ln M+1-\ln k-24 \ln M/12)} + \tbinom{M}{k} 2^{-2T} &\leq \\ M^{-k} \cdot e^{k(-\ln k+1)} + \tbinom{M}{k} 2^{-2T} &\leq \\ e \cdot M^{-k} + \tbinom{M}{k} 2^{-2T}&, \end{aligned}$$

where we have used the inequality $R \geq 25 \ln M \geq 24 \ln M$.

Taking a union bound over all $k \in [M/2]$, we conclude that all groups U of size at most $M/2$ satisfy $|N_G(U)| > k/2$, except with probability at most $2^M \cdot 2^{-2T} + \sum_{k=1}^{M/2} e \cdot M^{-k} \leq 2e \cdot M^{-1}$, since $M \geq 64$ and $T \geq M$. ■

4.4 Security

Proposition 9. *Fix any pair of players (p_i, p_j) for which $k_{i,j} \neq \bot$. Then, any adversary (with access to the full transcript of the protocol) that makes at most T_A queries to H, makes the query $k_{i,j}$ with probability at most $\frac{T_A}{N}$.*

The security proof is essentially identical to the proof for standard Merkle's puzzles.

Proof. Let $\varLambda$ be a random variable for the transcript of the protocol, which includes $H(k_{i,j})$, as well as other images. The query sets of the players are uniform, and H is a random function for which images do not give any information about their preimages.[4] Consequently, $k_{i,j} \mid \varLambda = \lambda$ is uniformly distributed in $[N]$ for any λ (for which the images of H computed by p_i and p_j intersect).

Fix an adversary for Algorithm 1 that receives $\varLambda$ as input. Let $\varGamma_t$ be a random variable for the first t (adaptive) queries of the adversary and their answers. Since H is a random function, any query $q \neq k_{i,j}$ to H may only give the information that $q \neq k_{i,j}$ (in case $H(q) \neq H(k_{i,j})$). Thus, by induction on the number of queries t, they either hit $k_{i,j}$ with probability at most t/N, or $k_{i,j} \mid \varLambda = \lambda, \varGamma_t = \gamma$ remains uniformly distributed in a set which contains (at least) the remaining $N - t$ inputs to H. Setting $t = T_A$ gives the result. ■

5 The Distributed Key Agreement Protocol

We describe our key agreement protocol in Algorithm 2, where every player receives as input the same set of edges E_s. We set $L = \lceil 16 \log T \rceil$ (the other parameters are set as in the setup protocol).

```
Algorithm 2: Distributed key agreement protocol
  Parameters: $M, T, N, N', D, L$
  Input: $E_s$
  Run the setup protocol (with parameters $M, T, N, N', D$) $2L$ times with
    independent random oracles (derived from $H$)
  If more than $L$ executions fail (i.e., output $\perp$), then each player outputs an
    independent and uniform value in $[N]$. Otherwise, for the first $L$ successful
    executions, denote the corresponding random oracles and secure graphs by
    $H^{(1)}, \dots, H^{(L)}$ and $G^{(1)}, \dots, G^{(L)}$
  For each $(i, j) \in E_s$, run the strong secure link protocol (Algorithm 3), after
    which $p_i$ outputs $k_{i,j}$ and $p_j$ outputs $k_{j,i}$
```

It remains to describe the strong secure link protocol. We assume that the players have access to a perfect encryption scheme: for $\ell \in [L]$ given access to an (independent) random function $F^{(\ell)} : [N] \times [M] \times [M] \times [T^2] \to [N]$, players f, g that share a key $k^{(\ell)}_{f,g} \in [N]$, encrypt the ct'th message $m \in [N]$ as

[4] The transcript reveals information about the equalities (and inequalities) among different queries made by the players, yet any individual query remains uniform in $[N]$.

$F^{(\ell)}(k^{(\ell)}_{i,j}, f, g, ct) + m \bmod N$. We embed f and g into the input of F in order to make sure that it is not invoked twice on the same input.

In the protocol, p_i chooses L independent and uniform values $r_1, \ldots, r_L \in [N]$ and computes $k_{i,j} = \sum_{\ell=1}^{L} r_\ell \bmod N$ (i.e, $k_{i,j}$ is split into L shares using a standard additive L-out-of-L secret sharing scheme). Then, p_i sends the ℓ'th share r_ℓ on a short path to p_j, encrypted with the keys of $G^{(\ell)}$. Specifically, for each edge (f, g) on the selected path, p_f encrypts r_ℓ with counter ct as $F^{(\ell)}(k^{(\ell)}_{f,g}, f, g, ct) + r_\ell \bmod N$ (p_f and p_g then increment the counter). Player g decrypts the message (by subtracting $F^{(\ell)}(k^{(\ell)}_{f,g}, f, g, ct)$ modulo N from the encryption) and encrypts it using the next key on the path. Finally, p_j receives the (encrypted) values $r_1, \ldots, r_L$ and computes $k_{j,i}$ (which should equal $k_{i,j}$) by decrypting and summing the values mod N.

The algorithm of p_i is given below.

Algorithm 3: Strong secure link protocol (p_i's algorithm)

```
Parameters: $M, N, L$
Input: $j$ such that $(i, j) \in E_s$
Select $L$ uniform and independent values $r_1, \ldots, r_L \in [N]$ and define
  $k_{i,j} = \sum_{\ell=1}^{L} r_\ell \bmod N$.
forall $\ell \in [L]$ do
    Find the shortest path between $i$ and $j$ in $G^{(\ell)}$ via breadth-first search,
      and send $r_\ell$ on that path (encrypted with the corresponding keys of $G^{(\ell)}$)
```

5.1 Security Analysis

Proposition 10. *Fix an adversary $\mathcal{A}$ that makes $T_A \leq N/4$ queries to $H^{(1)}, \ldots, H^{(L)}$ and $F^{(1)}, \ldots, F^{(L)}$. Then, given the view of the adversary* $\mathrm{view}_{\mathcal{A}}$ *(the transcript of the protocol of Algorithm 2 and the oracle queries and answers), each $k_{i,j}$ for $(i, j) \in E_s$ is uniformly distributed in $[N]$, except with probability at most T^{-6}. Namely,* $\Pr[(\forall (i,j) \in E_s : k_{i,j}$ *is uniformly distributed in* $[N]) \mid \mathrm{view}_{\mathcal{A}}] \leq T^{-6}$.

Proof. Let Λ be a random variable for the transcript of Algorithm 2. The adversary for Algorithm 2 receives Λ as input. Let Γ be a random variable for the (adaptive) queries of the adversary to $H^{(1)}, \ldots, H^{(L)}$, and $F^{(1)}, \ldots, F^{(L)}$ and their answers.

Fix $(i, j) \in E_s$. For $\ell \in [L]$, let $K^{(\ell)}$ be the set of keys under which r_ℓ is encrypted in Algorithm 3. Namely, $K^{(\ell)}$ contains $k^{(\ell)}_{f,g}$ for all edges (f, g) on the path in $G^{(\ell)}$ selected by p_i. Define the random variable $\mathcal{E}_\ell$ as an indicator for the event that Γ contains a query $F^{(\ell)}(k^{(\ell)}_{f,g}, f, g, ct)$ for some $k^{(\ell)}_{f,g} \in K^{(\ell)}$ and counter ct.

Claim. For any values λ, γ (that occur with positive probability),

$$k_{i,j} \mid \Lambda = \lambda, \Gamma = \gamma, \wedge_{\ell \in [L]} \mathcal{E}_\ell = 0$$

is distributed uniformly in $[N]$.

In other words, if $\wedge_{\ell \in [L]} \mathcal{E}_\ell = 0$ occurs, then $k_{i,j}$ is distributed uniformly in $[N]$ given the view of the adversary.

Proof. Given that $\wedge_{\ell \in [L]} \mathcal{E}_\ell = 0$, then there exists $\ell \in [L]$ such that $\mathcal{E}_\ell = 0$. We fix any such ℓ.

For each $k_{f,g}^{(\ell)} \in K^{(\ell)}$, denote by $c_{f,g}^{(\ell)} = F^{(\ell)}(k_{f,g}^{(\ell)}, f, g, ct) + r_\ell \bmod N$ the encryption (ciphertext) of r_ℓ, and denote $C^{(\ell)} = \{c_{f,g}^{(\ell)} \mid k_{f,g}^{(\ell)} \in K^{(\ell)}\}$. Since we assume the adversary did not query $F^{(\ell)}(k_{f,g}^{(\ell)}, f, g, ct)$ for any $k_{f,g}^{(\ell)} \in K^{(\ell)}$, and since $F^{(\ell)}$ is a random function, then $r_\ell \mid C^{(\ell)}, \mathcal{E}_\ell = 0$ remains uniformly distributed in $[N]$. As the additional values in the adversary's view are independent of r_ℓ (and of all $F^{(\ell)}(k_{f,g}^{(\ell)}, f, g, ct)$), then

$$r_\ell \mid \Lambda = \lambda, \Gamma = \gamma, \mathcal{E}_\ell = 0$$

is also uniform in $[N]$. Recall that $k_{i,j} = \sum_{\ell=1}^{L} r_\ell \bmod N$, where each share is selected independently and uniformly at random from $[N]$. Since r_ℓ is uniform in $[N]$ given the view of the adversary, then $k_{i,j}$ is uniform in $[N]$ given the view of the adversary regardless of the other shares. □

It remains to upper bound $\Pr[\wedge_{\ell \in [L]} \mathcal{E}_\ell = 1]$. A bound on this quantity in the information theoretic model essentially follows from Proposition 9.

Recall that for each $\ell \in [L]$, the path length in $G^{(\ell)}$ between i and j is at most $D = 4 \log M$, and hence $|K^{(\ell)}| \le 4 \log M$.

Assume without loss of generality that the adversary makes exactly T_A queries to $H^{(\ell)}$ (and $F^{(\ell)}$) for each $\ell \in [L]$. Recall that the L executions of the setup protocol are independent. Therefore, as in the proof of Proposition 9, by induction on the number of queries to $H^{(\ell)}$ (and $F^{(\ell)}$) (denoted by t), they either hit $K^{(\ell)}$ with probability at most $|K^{(\ell)}| \cdot t/N \le 4 \log M \cdot t/N$, or only give the information that $K^{(\ell)}$ does not intersect these queries. Hence, $\Pr[\mathcal{E}_\ell = 1] \le \left(\frac{4 \log M \cdot T_A}{N}\right)$ holds for each $\ell \in L$ independently of all $\mathcal{E}_f$ for $f \neq \ell$. Thus,

$$\Pr[\wedge_{\ell \in [L]} \mathcal{E}_\ell = 1] = \prod_{\ell \in [L]} \Pr[\mathcal{E}_\ell = 1 \mid \wedge_{f \in [\ell-1]} \mathcal{E}_f = 1] \le \left(\frac{4 \log M \cdot T_A}{N}\right)^L . \qquad (2)$$

Remark 1. We also need to condition on the success of the protocol that constructs $G^{(\ell)}$, i.e., on the event that the graph $G^{(\ell)}$ is of diameter at most $D = 4 \log M$. However, the diameter of the graph (or its structure in general) does not reveal any information about the individual queries of the players to $H^{(\ell)}$ (each one remains uniformly distributed). Hence the event is independent of the success probability of the adversary.

We can obtain a slightly better bound as follows. Consider a restricted adversary that before making any query to the oracles, fixes some subset $L' \subseteq L$ of size $L/2$ and makes at most $2T_A/L$ queries to $H^{(\ell)}$ (and $F^{(\ell)}$) for each $\ell \in L'$. For such an adversary,

$$\Pr[\wedge_{\ell \in [L]} \bar{\mathcal{E}}_\ell = 1] \leq \Pr[\wedge_{\ell \in [L']} \bar{\mathcal{E}}_\ell = 1] \leq \left(\tfrac{8 \log M \cdot T_A}{L \cdot N}\right)^{L/2}$$

similarly to (2) (where $\bar{\mathcal{E}}_\ell$ are random variables associated with the restricted adversary). For an arbitrary adversary that makes a total of at most T_A queries, there is always such a subset $L' \subseteq L$ of size $L/2$, but L' may depend on the oracle queries. Yet, we can build a restricted adversary from an arbitrary one by guessing the subset L' uniformly at random in advance. Since our guess is correct with probability at least 2^{-L}, we have

$$\Pr[\wedge_{\ell \in [L]} \mathcal{E}_\ell = 1] \leq 2^L \cdot \Pr[\wedge_{\ell \in [L]} \bar{\mathcal{E}}_\ell = 1] \leq 2^L \left(\tfrac{8 \log M \cdot T_A}{L \cdot N}\right)^{L/2} = \left(\tfrac{32 \log M \cdot T_A}{L \cdot N}\right)^{L/2}.$$

Since $L \geq 16 \log T$ and $T_A \leq N/4$, we get $\Pr[\wedge_{\ell \in [L]} \mathcal{E}_\ell = 1] \leq 2^{-L/2} \leq T^{-8}$.

The proposition follows by a union bound over all $(i, j) \in E_s$ (whose size is less than $M^2 \leq T^2$). ■

5.2 Main Theorem

The formal version of Theorem 1 is given below.

Theorem 2. *Assume that $M \geq 64$, $T \geq 20000$, $T \geq M$ and $|E_s| = \tilde{O}(M)$. Let $\hat{T} = \tilde{O}(T)$ be sufficiently large. Then, Protocol 2 is a*

$$\left(M, \alpha = 1 - \tilde{O}(\hat{T}^{-1}), \hat{T}, \beta = 1 - \tilde{O}(\hat{T}^{-1})\right)\text{-}\mathit{DKAP}$$

which is

$$\left(T_A = \tilde{\Theta}(M \cdot \hat{T}^2), \alpha_A = \tilde{O}\left(\tfrac{1}{M \cdot \hat{T}^2}\right)\right)\text{-}\mathit{secure.}$$

Proof. We prove the equivalent statement that Protocol 2 is a

$$\left(M, \alpha = 1 - \tilde{O}(T^{-1}), \tilde{O}(T), \beta = 1 - \tilde{O}(T^{-1})\right)\text{-DKAP}$$

which is

$$\left(T_A = \tilde{\Theta}(M \cdot T^2), \alpha_A = \tilde{O}\left(\tfrac{1}{M \cdot T^2}\right)\right)\text{-secure.}$$

Correctness. The protocol is correct if at least L of the setup protocol executions do not fail and all pairs of players agree on consistent keys. By Proposition 2 and a union bound over the $2L$ executions, a pair of players output inconsistent keys with probability at most $2L \cdot T^{-2}$.

By Proposition 8, each execution of the setup protocol fails (the players output $\perp$) with probability at most $2e \cdot M^{-1} \leq 1/8$ (since $M \geq 64$). In a sequence of L independent executions, all fail with probability at most 2^{-3L}. Hence, there exists such a sequence among the $2L$ executions of the protocol with probability at most $2^{2L} \cdot 2^{-3L} = 2^{-L}$.

Therefore, the protocol is correct, except with probability at most

$$2L \cdot T^{-2} + 2^{-L} \leq T^{-1} = \tilde{O}(T^{-1})$$

(as $T \geq 20000$).

Queries and communication. The setup protocol is executed $2L = 2\lceil 16 \log T \rceil < 34 \log T$ times. By Proposition 3, in each execution each player makes at most T queries with probability 1 and communicates $\tilde{O}(T)$ bits, except with probability $M \cdot 2^{-T} + (36 \log T \cdot T)^{-1} + T^{-2}$. Moreover, each edge in E_s results in at most L additional queries (and L messages) per player in the strong secure link protocol (Algorithm 3).

Thus, each player makes less than

$$34 \log T \cdot T + 17 \log T \cdot |E_s| = \tilde{O}(T)$$

queries, and communicates $\tilde{O}(T)$ bits, except with probability at most

$$2L \cdot (M \cdot 2^{-T} + (36 \log T \cdot T)^{-1} + T^{-2}) \leq T^{-1} = \tilde{O}(T^{-1})$$

(since $T \geq 20000$).

Security. By Proposition 10, any adversary that makes at most

$$T_A = \tfrac{N}{8} \geq \tfrac{M \cdot T^2}{208 \ln M} = \tilde{\Theta}(M \cdot T^2)$$

queries to the random oracle outputs $((i,j), k_{i,j})$ for $(i,j) \in E_s$ with probability at most

$$T^{-6} \cdot 1 + (1 - T^{-6}) \cdot 1/N \leq \tfrac{2}{N} \leq \tfrac{52 \ln M}{M \cdot T^2} = \tilde{O}\left(\tfrac{1}{M \cdot T^2}\right).$$

■

6 Optimality of the Distributed Key Agreement Protocol

We prove the optimality of our key agreement protocol (up to logarithmic factors) with respect to the ratio of the number of queries made by each honest player and the adversary. We use the following result.

Theorem 3 ([3], Theorem 3.1, adapted). *Let Π be a 2-player key agreement protocol between p_1 and p_2 using a random oracle H in which:*

- *p_1 makes at most T_1 queries to H and outputs $k_{1,2}$.*

- *p_2 makes at most T_2 queries to H and outputs $k_{2,1}$.*
- $\Pr[k_{1,2} = k_{2,1}] \geq \alpha$

Then, for every $0 < \delta < \alpha$, there is an adversary with access to the transcript of the protocol that makes at most $400 \cdot T_1 \cdot T_2/\delta^2$ queries to H and outputs $k_{2,1}$ with probability at least $\alpha - \delta$.

Proposition 11. *Let Π be a protocol between M players using a random oracle H in which:*

- *Every player makes at most T queries to H.*
- *p_1 outputs $j \in [M]\backslash\{1\}$ and $k_{1,j}$, and p_j outputs $k_{j,1}$.*
- $\Pr[k_{1,j} = k_{j,1}] \geq \alpha$.

Then, for every $0 < \delta < \alpha$, there is an adversary with access to the transcript of the protocol that makes at most $400 \cdot M \cdot T^2/\delta^2$ queries to H and outputs $k_{j,1}$ with probability at least $\alpha - \delta$.

Proof. Given an M-player protocol Π as above with players $p_1, \ldots, p_M$, we devise a 2-player protocol Π' with players p'_1 and p'_2 as follows: player p'_1 simulates p_1 by sending all messages intended for $p_2, \ldots, p_M$ to p'_2. Player p'_2 simulates $p_2, \ldots, p_M$ by sending all messages intended for p_1 to p'_1 (messages sent among $p_2, \ldots, p_M$ do not require communication). Finally, if p_1 outputs j and $k_{1,j}$, then p'_1 outputs $k'_{1,2} = k_{1,j}$ and sends j to p'_2 that outputs $k'_{2,1} = k_{j,1}$.

We have $\Pr[k_{1,j} = k_{j,1}] \geq \alpha$, and hence $\Pr[k'_{1,2} = k'_{2,1}] \geq \alpha$. Moreover, p'_1 makes at most T queries to H, while p'_2 makes at most $(M-1) \cdot T < M \cdot T$ queries to H. Therefore, by Theorem 3, there exists an adversary $\mathcal{A}'$ with access to the transcript of Π' that makes at most $400 \cdot M \cdot T^2/\delta^2$ queries to H and outputs $k'_{2,1}$ with probability at least $\alpha - \delta$.

We devise an adversary $\mathcal{A}$ for Π using $\mathcal{A}'$: $\mathcal{A}$ gives to $\mathcal{A}'$ only the messages sent and received by p_1 (so that the transcript is identical to the corresponding execution of Π') and outputs the same value. Thus, $\mathcal{A}$ makes at most $400 \cdot M \cdot T^2/\delta^2$ queries to H and outputs $k_{j,1} = k'_{2,1}$ with probability at least $\alpha - \delta$. ■

Theorem 4. *Any (M, α, T, β)-DKAP that is (T_A, α_A)-secure for non-empty $E_s \subseteq [M] \times [M]$ such that $\alpha \geq 3/4$ and $T_A \geq 6400M \cdot T^2$, satisfies $\alpha_A \geq 1/2$.*

Proof. Apply Proposition 11 for an edge $(j, 1) \in E_s$ (by renaming the players) and $\delta = 1/4$. Since $\Pr[k_{1,j} = k_{j,1}] \geq 3/4$, and $T_A \geq 6400M \cdot T^2 = 400M \cdot T^2/\delta^2$, there exists an adversary that makes at most T_A queries to H and outputs $k_{j,1}$ with probability at least $3/4 - 1/4 = 1/2$. ■

7 Extensions

We briefly discuss two extensions of the protocol.

7.1 The Semi-Honest Model

We consider security in a model where adversarial players execute the protocol as designed, but try to learn the secret keys of the honest players.

With small overhead, the protocol can be extended to provide resistance against an adversary that controls a fraction of $O(1/\log M)$ of the players in the semi-honest model, which are chosen in advance (i.e., static corruptions). In particular, such an extension allows any two honest players to communicate securely, except with negligible probability (unless the adversary makes $\tilde{\Omega}(M \cdot T^2)$ queries to the random oracle).

The extension is simple. Fix some edge $(i, j) \in E_s$ between two honest players. Note that the only advantage of the corrupted players (over an eavesdropping adversary) is in the strong secure link protocol. Specifically, in this protocol p_i chooses $k_{i,j}$ and sends each of its shares on a path to p_j, encrypted using secure links created by a setup protocol execution. In order to maintain security, we must ensure that with high probability, there is at least one path in which all players are honest.

Recall that each path chosen by p_i to encrypt $k_{i,j}$ is of length at most $D = 4 \log M$. Therefore, each path does not include any corrupted player with constant probability. Repeating the setup protocol independently $\Omega(\log T)$ times (while choosing among shortest paths independently via randomization), ensures that $\Omega(\log T)$ paths do not include a corrupted player (except with small probability) and the analysis of the original protocol applies to these paths with small modifications. Thus, the only change required is to repeat the setup protocol according to the fraction of adversarial players we wish to tolerate. On the other hand, we conjecture that it is not possible to tolerate a constant fraction of adversarial players with a small overhead of $\tilde{O}(1)$ in query complexity.

7.2 Communication-Security Tradeoff

For a parameter $B \geq 1$ such that $T = \tilde{\Omega}(M \cdot B)$, it is possible to extend the protocol such that each player makes T queries and communicates $\tilde{O}(T/B)$ bits, while any adversary has to make $\tilde{\Omega}\left(\frac{M \cdot T^2}{B}\right)$ queries to recover any key with high probability. As for standard Merkle's puzzles, this can done by defining a new random oracle H' based on H by partitioning its domain into groups of size B. The output of a query to H' is computed by summing (modulo N') the outputs of the corresponding group (consisting of B queries to H).

References

1. Barak, B.: The Complexity of Public-Key Cryptography. In: Lindell, Y. (ed.) Tutorials on the Foundations of Cryptography, pp. 45–77. Springer International Publishing (2017)

2. Barak, B., Mahmoody-Ghidary, M.: Merkle Puzzles Are Optimal - An $O(n^2)$-Query Attack on Any Key Exchange from a Random Oracle. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5677, pp. 374–390. Springer (2009)
3. Barak, B., Mahmoody-Ghidary, M.: Merkle's Key Agreement Protocol is Optimal: An $\mathrm{O}(\mathrm{n}^2)$ Attack on Any Key Agreement from Random Oracles. J. Cryptol. 30(3), 699–734 (2017)
4. Biham, E., Goren, Y.J., Ishai, Y.: Basing Weak Public-Key Cryptography on Strong One-Way Functions. In: Canetti, R. (ed.) Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008. Lecture Notes in Computer Science, vol. 4948, pp. 55–72. Springer (2008)
5. Chan, H., Perrig, A., Song, D.X.: Random Key Predistribution Schemes for Sensor Networks. In: 2003 IEEE Symposium on Security and Privacy (S&P 2003), 11-14 May 2003, Berkeley, CA, USA. p. 197. IEEE Computer Society (2003)
6. Chung, F., Lu, L.: The Diameter of Sparse Random Graphs. Adv. Appl. Math. 26(4), 257–279 (2001)
7. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory 22(6), 644–654 (1976)
8. Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly Secure Message Transmission. J. ACM 40(1), 17–47 (1993)
9. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: Atluri, V. (ed.) Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002. pp. 41–47. ACM (2002)
10. Fischer, M.J., Wright, R.N.: Multiparty Secret Key Exchange Using a Random Deal of Cards. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings. Lecture Notes in Computer Science, vol. 576, pp. 141–155. Springer (1991)
11. Goldreich, O., Nisan, N., Wigderson, A.: On Yao's XOR-Lemma. In: Goldreich, O. (ed.) Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman, Lecture Notes in Computer Science, vol. 6650, pp. 273–301. Springer (2011)
12. Haitner, I., Mazor, N., Oshman, R., Reingold, O., Yehudayoff, A.: On the Communication Complexity of Key-Agreement Protocols. In: Blum, A. (ed.) 10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA. LIPIcs, vol. 124, pp. 40:1–40:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
13. Impagliazzo, R., Luby, M.: One-way Functions are Essential for Complexity Based Cryptography (Extended Abstract). In: 30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989. pp. 230–235. IEEE Computer Society (1989)
14. Ingemarsson, I., Tang, D.T., Wong, C.K.: A conference key distribution system. IEEE Trans. Inf. Theory 28(5), 714–719 (1982)
15. Krivelevich, M.: Expanders - how to find them, and what to find in them (2019)

16. Leighton, F.T., Micali, S.: Secret-Key Agreement without Public-Key Cryptography. In: Stinson, D.R. (ed.) Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings. Lecture Notes in Computer Science, vol. 773, pp. 456–479. Springer (1993)
17. Merkle, R.C.: Secure Communications Over Insecure Channels. Commun. ACM 21(4), 294–299 (1978)
18. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press (2005)