Key-Agreement with Perfect Completeness from Random Oracles

Noam Mazor*

August 21, 2023

Abstract

In the Random Oracle Model (ROM) all parties have oracle access to a common random function, and the parties are limited in the number of queries they can make to the oracle. The Merkle's Puzzles protocol, introduced by Merkle [CACM '78], is a key-agreement protocol in the ROM with a quadratic gap between the query complexity of the honest parties and the eavesdropper. This quadratic gap is known to be optimal, by the works of Impagliazzo and Rudich [STOC '89] and Barak and Mahmoody [Crypto '09].

When the oracle function is injective or a permutation, Merkle's Puzzles has perfect completeness. That is, it is certain that the protocol results in agreement between the parties. However, without such an assumption on the random function, there is a small error probability, and the parties may end up holding different keys. This fact raises the question: Is there a key-agreement protocol with *perfect* completeness and super-linear security in the ROM?

In this paper we give a positive answer to the above question, showing that changes to the query distribution of the parties in Merkle's Puzzles, yield a protocol with perfect completeness and roughly the same security.

Keywords: Key-Agreement, Random Oracle, Merkle's Puzzles, Perfect Completeness

1 Introduction

Key-Agreement ([DH76]) is a fundamental primitive in modern cryptography. Key-Agreement protocols allow two parties, Alice and Bob, to agree on a secret key that is hidden from any eavesdropper, just by *publicly* communicating with each other. While the above is one of the most important tasks in cryptography, still there are a respectively small number of assumptions from which we know how to construct such protocols ([RSA78; Rab79; DH76; ElG84; McE78; AD97; Reg09; BCNHR22; Ale03; ABW10; Sho99]). Moreover, these assumptions are more structured compared to the assumptions which are used to base private-key cryptography, making them more vulnerable to attacks, and arguably less reliable.

The Random Oracle Model. An important alternative to such assumptions is the use of key-agreement protocols with proven security in the *Random Oracle Model* (ROM). In this model we assume that the parties (and the eavesdropper) have oracle access to an idealized hash function - a perfectly random function $F\colon [N] \to [M]$. Then, the efficiency of the eavesdropper is measured with respect to the number of oracle calls it needs to make in order to break the security.

*Cornell Tech. E-mail: noammaz@gmail.com. Supported in part by NSF CNS-2149305 and NSF CNS-2128519.

Unfortunately, while we usually want the gap between the running time of the honest parties in the protocol and the running time of the adversary to be exponential (or at least super-polynomial), this cannot be achieved in the random oracle model. Indeed, as shown by Impagliazzo and Rudich [IR89] and Barak and Mahmoody [BM09], any protocol in which the honest parties make ℓ queries can be broken by an adversary that makes $O(\ell^2)$ queries. Yet, while this quadratic gap is not ideal, in the lack of other reliable protocols it may be sufficient (see for example [HMORY19]).

Merkle's Puzzles. The above quadratic gap is achievable, using a very elegant and simple protocol, called Merkle's Puzzles [Mer78]. In this protocol, Alice and Bob each choose ℓ random queries, $(x_1, \ldots, x_\ell)$ and $(y_1, \ldots, y_\ell)$ respectively, from a domain of size $O(\ell^2)$. Then, Alice and Bob use the images of the queries to find an intersecting query (which exists with high probability due to the birthday paradox). Specifically, Alice sends $F(x_1), \ldots, F(x_\ell)$; Bob then search for indexes j and k such that $F(x_j) = F(y_k)$, and sends j to Alice; Lastly, Alice outputs x_j and Bob outputs y_k as as their keys.

For security, it is not hard to see that an eavesdropper that wants to learn Alice's output needs to invert the function F on a random image $F(x_j)$, and thus essentially needs to query all the $O(\ell^2)$ elements in the domain of the function.

When the function F is a random function with a large enough range, the probability that the protocol succeeds, that is, results in an agreement between Alice and Bob, is approaching to 1. Indeed, an error can occur only when two distinct queries have the same image. Moreover, when using a more structured function, such as a permutation or any injective function, it follows that Merkle's Puzzles has perfect agreement probability. Yet, it is an interesting question whether there exists such a key-agreement protocol without assuming any structure on the oracle function. That is:

> *Is there a key-agreement protocol with perfect agreement and super-linear security in the ROM?*

1.1 Our Results

In this paper, we give a positive answer to the above question. More specifically, by changing the distribution of queries in the Merkle's Puzzles protocol, we get a protocol with perfect agreement and roughly the same security.

Theorem 1.1 (Main theorem, informal)**.** *There exists a ℓ-query key-agreement protocol with perfect agreement in the random oracle model, secure against all $o(\ell^2)$-query adversaries.*

See Theorem 3.1 for the formal statement. As mentioned above, we get this result by changing the query distribution of Merkle's Puzzles. The main observation is that in our distribution, the parties always have a single query in common. Then, by verifying that the sets of images of the parties only intersect in (exactly) one point, the parties can be sure that the outputs are equal.[1]

[1] Choosing a distribution such that Alice and Bob always have an intersection is not hard. Indeed, for $F\colon [\ell]\times[\ell] \to [M]$, consider the distribution in which Alice samples a random index $i \leftarrow [\ell]$, and query $F(i,1), \ldots, F(i,\ell)$, while Bob queries $F(1,j), \ldots, F(\ell,j)$ for a random $j \leftarrow [\ell]$. It is not hard to see that $F(i,j)$ is always in the intersection. However, if the parties continue as in Merkle's Puzzles (namely, Alice sends $F(i,1), \ldots, F(i,\ell)$, and Bob finds the intersection), an eavesdropper can learn *all* the queries made by Alice, just by simulating Bob. Finding such a distribution for which the protocol remains secure is the main challenge.

The protocol. We next describe the query distribution we consider. For simplicity, we think of F as a function over the domain $[\ell] \times [\ell]$. To choose her queries, Alice samples ℓ uniformly random points $x_1, \ldots, x_\ell \in [\ell]$. Then, Alice queries $(1, x_1), \ldots, (\ell, x_\ell)$. On the other side, Bob chooses a random index $j \in [\ell]$, and queries $(j, 1), \ldots, (j, \ell)$. This distribution guarantees that (j, x_j) is a (unique) intersection between Alice's and Bob's queries.

The protocol proceeds similarly to Merkle's Puzzles: Alice sends $F(1, x_1), \ldots, F(\ell, x_\ell)$ to Bob, but in a random order. Bob then searches for an intersection between $F(1, x_1), \ldots, F(\ell, x_\ell)$ and his own images, $F(j, 1), \ldots, F(j, \ell)$. Crucially, if Bob finds more than one intersection, he notifies Alice and they both output 0. Otherwise, he sends the intersection and both parties output its preimage, (j, x_j).

Security. While the agreement of this protocol follows immediately by the fact that there is always an intersection, the security is a bit less obvious. Indeed, an adversary that tries to invert $z = F(j, x_j)$ can learn some information on j by observing the other images sent by Alice. Specifically, if the adversary finds a query (a, b) such that $F(a, b) \neq z$ appears in the communication from Alice to Bob, she learns that $j \neq a$. This can potentially rule out ℓ possible preimages. Nevertheless, we are able to show by a simple reduction that this cannot help too much.

To prove the above we consider a *balls and boxes* game, defined as follows. There are ℓ^2 boxes ordered in ℓ rows and ℓ columns. In each row, we chose a box uniformly at random and place one ball in it (all other boxes are left empty). All the balls are blue, except exactly one random ball which is red. The goal of the player is to find the red ball, by opening as few boxes as possible.

The security of Merkle's Puzzles corresponds to a similar game, in which there is only one red ball within a single random row, and there are no other (blue) balls. In this case, it is not hard to see (by the union bound) that any player that opens q boxes, succeeds in probability at most q/ℓ^2 in finding the red ball. To prove that our protocol is secure we first show that the security of the protocol is equivalent to the hardness of the above box and balls game. Then, we show that the presence of the blue balls cannot help the player too much. This is done by observing a simple reduction between the games with and without the blue balls. Specifically, we show that any player that opens q boxes, cannot win with probability larger than $2q/\ell^2$.

Comparison with Merkle's Puzzles. Similarly to Merkle's Puzzles, the above protocol is a two-message protocol, and has non-adaptive queries. Both protocols also have near-optimal communication complexity for protocols with these properties [HMORY19] (and can be generalized to get other possible tradeoffs between communication and security).

To compare the security of Merkle's Puzzles and the protocol given in this work, recall that to have an intersection with $1 - o(1)$ probability, the parties in Merkle's Puzzles should query ℓ queries each from a domain $\mathcal{D}$ of size at most $o(\ell^2)$. Thus, an q-query adversary can learn the key with probability $q/|\mathcal{D}| = \omega(q/\ell^2)$. In this case, we are able to get a slightly better security due to the fact that, in the protocol presented here, there is an intersection with probability 1, when the domain of the function is of size (exactly) ℓ^2.

Perspective and Additional Related Work

Perfect primitives in the ROM. Being an idealized hash function, the random oracle model is used to prove black-box separation between cryptographic primitives. Notably, Kahn, Saks, and

Smyth [KSS00] prove a black-box separation between one-way permutations and one-way functions, showing that it is impossible to construct a one-way permutation from random oracles alone.

As mentioned above, Impagliazzo and Rudich [IR89] prove that there is no key-agreement with super-polynomial security in the ROM, and Barak and Mahmoody [BM09] prove a tight quadratic bound on the security of key-agreement protocols in the ROM. The latter result was generalized by Haitner, Omri, and Zarosim [HOZ16] to hold for any semi-honest, no-input, two-party task. Prior to the work of [BM09], Brakerski, Katz, Segev, and Yerukhimovich [BKSY11] proved a similar quadratic bound on the security of any key-agreement protocol with *perfect* completeness in the ROM. However, as mentioned by Barak and Mahmoody [BM09], before the work presented here, it was conceivable that every key-agreement protocol with perfect completeness in the ROM can be broken with a linear number of queries to the oracle.

Public-key encryption with perfect completeness. A long line of work is dedicated to the task of eliminating errors in public-key encryption schemes (or, equivalently, two-message key-agreement) and in other cryptographic primitives [GGH97; DNR04; HR05; LT13; BV17]. As mentioned in [DNR04; BV17], even a small decryption error in a public-key encryption scheme can be problematic for some uses. Moreover, as demonstrated by Proos [Pro03] and Boneh, DeMillo, and Lipton [BDL01], such decryption error can be used by an adversary to attack the cryptographic system.

2 Preliminaries

2.1 Notations

We use calligraphic letters to denote sets and distributions, uppercase for random variables, and lowercase for values and functions. When unambiguous, we will naturally view a random variable as its marginal distribution. For a set $\mathcal{S}$, let $x \leftarrow \mathcal{S}$ denote that x is drawn uniformly from $\mathcal{S}$. For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. For two sets $\mathcal{A}, \mathcal{B}$, let $\mathcal{A} \times \mathcal{B} = \{(a, b) \colon a \in \mathcal{A}, b \in \mathcal{B}\}$.

2.2 Interactive Protocols and Oracle-Aided Protocols

A no-input, two-party protocol $\Pi = (\mathsf{A}, \mathsf{B})$ is simply a pair of probabilistic interactive algorithms. The interaction between the parties A and B is carried out in rounds, where in each round one party is active and the other is idle. In the j-th round of the protocol, the active party sends a message to the other party, according to its partial view (its randomness and the messages sent in the protocol so far). The transcript of a given execution of the protocol is the list of messages exchanged between the parties in the execution of the protocol.

For a protocol Π, we use $(\mathsf{trans}, \mathrm{out}_{\mathsf{A}}, \mathrm{out}_{\mathsf{B}}) \leftarrow \Pi$ to denote that $(\mathsf{trans}, \mathrm{out}_{\mathsf{A}}, \mathrm{out}_{\mathsf{B}})$ is a triplet sampled according to the joint distribution induced by Π, which contains a transcript of the protocol, an output of party A, and an output of party B in a random execution of Π.

An oracle-aided no-input two-party protocol $\Pi = (\mathsf{A}, \mathsf{B})$ is a pair of interactive oracle-aided algorithms. For a function f, we use $\Pi^f = (\mathsf{A}^f, \mathsf{B}^f)$ to denote the protocol Π when using f as the oracle function. Such a protocol $\Pi = (\mathsf{A}, \mathsf{B})$ is a ℓ-query protocol, if for every function f, each party always queries f on at most ℓ points during the interaction between A^f and B^f.

2.3 Key-Agreement Protocols

We now define key-agreement protocols with respect to a function family $\mathcal{F}$. In the random oracle model, we choose $\mathcal{F}$ to be the all-function family, over some fixed domain and range.

Definition 2.1 (Key-agreement protocol)**.** *A no-input oracle-aided two-party protocol* (A, B) *is a* (q, α, γ) key-agreement protocol *with respect to a function family* $\mathcal{F}$ *if the following holds:*

1. Agreement:
$$\Pr_{f \leftarrow \mathcal{F}, (\mathsf{trans}, \mathrm{out}_{\mathsf{A}}, \mathrm{out}_{\mathsf{B}}) \leftarrow \Pi^f}[\mathrm{out}_{\mathsf{A}} = \mathrm{out}_{\mathsf{B}}] \geq 1 - \alpha.$$
2. Security: *For every algorithm* E*, making at most* q *queries, it holds that*
$$\Pr_{f \leftarrow \mathcal{F}, (\mathsf{trans}, \mathrm{out}_{\mathsf{A}}, \mathrm{out}_{\mathsf{B}}) \leftarrow \Pi^f}\left[\mathsf{E}^f(\mathsf{trans}) = \mathrm{out}_{\mathsf{A}}\right] \leq \gamma.$$

Note that the above security definition requires that the adversary will not be able to guess the key, but the adversary may know some partial information on the output of A (for example, the first bit). Given such a security guarantee, the parties can amplify it to get a stronger security notion, in which the key is (somewhat) indistinguishable from random in the eyes of the eavesdropper. This can be done by either extracting randomness from the outputs (for example, using the Goldreich-Levin hardcore predicate [GL89]), or using the random oracle, by querying the common output and using its image as the key.[2]

Let $\mathcal{D}$ and $\mathcal{R}$ be two sets, and let $\mathcal{F} = \{f \mid f\colon \mathcal{D} \to \mathcal{R}\}$ be the all-function family over the domain $\mathcal{D}$ and the range $\mathcal{R}$. When $|\mathcal{D}| = \Omega(\ell^2)$, and $|\mathcal{R}| >> \ell^2$, the Merkle's Puzzles protocol, mentioned in the introduction, is a ℓ-query, $(q, 0.01, O(q/\ell^2))$ key-agreement protocol with respect to $\mathcal{F}$, and for every $q \in \mathbb{N}$.

3 The Key-Agreement Protocol

We now describe our key-agreement protocol. Let $\ell, M \in \mathbb{N}$ be two parameters, and let $\mathcal{F}_{\ell,M} = \left\{f \mid f\colon [\ell^2] \to [M]\right\}$ be the family of all functions from $[\ell^2]$ to $[M]$. For simplicity, we think of the domain of every $f \in \mathcal{F}_{\ell,M}$ as $[\ell] \times [\ell]$. Namely, we think of every $f \in \mathcal{F}_{\ell,M}$ as $f\colon [\ell] \times [\ell] \to [M]$.

Theorem 3.1. *The following holds for every* $\ell, M \in \mathbb{N}$*. There exists an* ℓ*-query two-party protocol, which is a* $(q, 0, \frac{2(q+1)}{\ell^2} + \frac{2\ell^2}{M})$ *key-agreement protocol with respect to to* $\mathcal{F}_{\ell,M}$*, for every* $q \in \mathbb{N}$*.*

The protocol is as follows.

Protocol 3.2.

Parameters: $\ell, M \in \mathbb{N}$*.*

Common function: $f\colon [\ell] \times [\ell] \to [M]$*.*

Operation:

1. *Party* A *samples* $x_1, \ldots, x_\ell \leftarrow [\ell]$ *uniformly and independently at random, and queries* $(1, x_1), \ldots, (\ell, x_\ell)$*. Let* $a_i = f(i, x_i)$ *for every* $i \in [\ell]$*.*

[2] A problem can occur in the last approach if the image of the key is used in the protocol, as happens in Merkle's Puzzles. In this case, the parties can, for example, use only the prefix of the images in the protocol, and use the suffix as the secret key.

2. *Party* B *samples an index* $j \leftarrow [\ell]$ *uniformly at random, and queries* $(j,1),\ldots,(j,\ell)$. *Let* $b_i = f(j,i)$ *for every* $i \in [\ell]$.

3. A *sends* $a_1,\ldots,a_\ell$ *in a random order.*

4. B *checks if the number of different images in* $\{a_1,\ldots,a_\ell,b_1,\ldots,b_\ell\}$ *is exactly* $2\ell-1$ *(that is, if there is exactly one collision). If not,* B *notifies* A*, and both parties output* 0. *Otherwise,* B *finds the unique* $z \in \{a_1,\ldots,a_\ell\} \cap \{b_1,\ldots,b_\ell\}$*, and sends* z *to* A.

5. *Party* A *outputs* (i,x_i) *for the unique* $i \in [\ell]$ *for which* $f(i,x_i) = z$. B *outputs* (j,k) *for the unique* $k \in [\ell]$ *for which* $f(j,k) = z$.

We turn to analyze the agreement and security of Protocol 3.2. Towards this, fix the parameters M and ℓ, and let $\mathcal{F} = \mathcal{F}_{\ell,M}$. Let $F \leftarrow \mathcal{F}$ be a random variable representing a random function from $\mathcal{F}$, and let O_{A} and O_{B} be the outputs of A and B, respectively, in a random execution of Protocol 3.2 using the function F. Finally, Let T be the transcript of the protocol in this execution. Theorem 3.1 is a direct implication of the following two claims. The first states that Protocol 3.2 has perfect agreement.

Claim 3.3 (Agreement)**.**

$$\Pr[O_{\mathsf{A}} = O_{\mathsf{B}}] = 1.$$

The second claim establishes the security of the protocol.

Claim 3.4 (Security)**.** *For every* q*-query algorithm* E*, it holds that*

$$\Pr\left[\mathsf{E}^F(T) = O_{\mathsf{A}}\right] \leq \frac{2(q+1)}{\ell^2} + \frac{2\ell^2}{M}.$$

We prove Claims 3.3 and 3.4 below, but first let us conclude the proof of Theorem 3.1.

Proof of Theorem 3.1. The proof follows by Claims 3.3 and 3.4, together with the observation that the parties in Protocol 3.2 make ℓ queries each. □

3.1 Agreement - Proving Claim 3.3

We start by showing that Protocol 3.2 has perfect agreement.

Proof of Claim 3.3. Claim 3.3 follows immediately by the query distribution of the parties. Specifically, observe that for every sampling of the ℓ points of A, $x_1,\ldots,x_\ell$, and for every sampling of j, A and B have a common query, (j,x_j). Thus, the parties have at least one common image. If further there are no other collisions in the union of the two sets of images, then $O_{\mathsf{A}} = O_{\mathsf{B}} = (j,x_j)$.

On the other hand, when there are other collisions in the union of the sets, $O_A = O_B = 0$. Overall, we get that $O_{\mathsf{A}} = O_{\mathsf{B}}$, with certainty, as wanted. □

3.2 Security - Proving Claim 3.4

In this part we prove that Protocol 3.2 has roughly the same security as Merkle's Puzzles. We do it by defining simplified security games, and showing a reduction between the security of Protocol 3.2 and these games.

Balls and Boxes games. To analyze the security of Protocol 3.2, in the following we define two "balls and boxes" games, to which we refer as the "unique-ball" game, and the "multi-ball" game. Intuitively, breaking the security of Merkle's Puzzles is equivalent to winning the unique-ball game, and, on the other hand, we will show that breaking the security of Protocol 3.2 is equivalent to winning the multi-ball game. In the following we define these two games, and show that every strategy for the multi-ball game is an (almost) as good strategy for the unique-ball game. This will imply that Protocol 3.2 has security similar to the security of Merkle's Puzzles.

We start with defining the unique-ball game.

Definition 3.5 (The unique-ball game)**.** *Let $\ell, q \in \mathbb{N}$ be two parameters. In the (ℓ, q)-*unique-ball game*, there are ℓ^2 boxes ordered in ℓ rows and ℓ columns. All the boxes are empty, except for one box, chosen uniformly at random, that contains a red ball in it.*

The player is allowed to (adaptively) open up to q boxes. The player wins the game if it finds the ball.

It is not hard to see that for every strategy of the player, the probability of finding the ball is at most q/ℓ^2. We now define the multi-ball game.

Definition 3.6 (The multi-ball game)**.** *Let $\ell, q \in \mathbb{N}$ be two parameters. In the (ℓ, q)-*multi-ball game*, there are ℓ^2 boxes ordered in ℓ rows and ℓ columns. In each row, there is a single box chosen uniformly and independently at random, that contains a ball in it. All the other boxes are empty. The ball in one row, chosen uniformly at random, is red, and the other balls are blue.*

The player is allowed to (adaptively) open up to q boxes. The player wins the game if it finds the red ball.

That is, as in the unique-ball game, the goal of the player is to find the box that contains the red ball, out of ℓ^2 possible boxes. However, this time when the player finds a blue ball, it knows that the red ball is not in this row, and thus can stop opening boxes in this row. Intuitively, this corresponds to the images A sends in Protocol 3.2, which can help the adversary to rule out some possible indices. The connection between the security of Protocol 3.2 to the multi-ball game is stated in the following claim.

Claim 3.7. *Assume there exists a q-query algorithm E such that $\Pr\left[\mathsf{E}^F(T) = O_{\mathsf{A}}\right] = p$. Then there exists a player P that wins the $(\ell, q+1)$-multi-ball game with probability at least $p - \frac{2\ell^2}{M}$.*

We prove Claim 3.7 in Section 3.3, but first let us use it to prove Claim 3.4. To do so, we need to prove an upper bound on the winning probability of every strategy for the multi-ball game. This is done by showing that every strategy to the multi-ball game can be used in the unique-ball game with roughly the same success probability. Formally,

Claim 3.8. *Assume there exists a strategy for the (ℓ, q)-multi-ball game that wins with probability p. Then there exists a strategy for the (ℓ, q)-unique ball game with winning probability $p/2$.*

Proof of Claim 3.8. Let P be a player for the (ℓ, q)-multi-ball game, that wins with probability p. We show how to use P to construct P' that wins in the (ℓ, q)-unique-ball game with probability $p/2$.

Given ℓ^2 boxes, P' samples $x_1, \ldots, x_\ell \leftarrow [\ell]$ uniformly and independently at random. Informally, P' will "plant" a blue ball in the box at location (i, x_i) for every $i \in [\ell]$, and then run P. More

formally, P' simulates the game for P. Every time P chooses to open a box (i, j), P' opens the same box. If this box contains the red ball, P' wins. If this box is empty and $j = x_i$, P' reports to P that the box contains a blue ball. Otherwise, P' reports to P that the box is empty.

Observe that P' wins if and only if P finds the red ball in the simulated game.[3] To see that P wins the simulated game with probability at least $p/2$, first observe that in the simulated game, there is one row i^* that contains two balls (one red ball and one blue ball, possibly at the same box), and all the other rows contain exactly one (blue) ball. Notice that if both of the balls in row i^* were red, the winning probability of P in the simulated game was at least p (as we only added a red ball to the grid). That is, P finds (at least) one ball on row i^* with probability at least p. By symmetry, the probability that the first ball in row i^* found by P is indeed the true red ball is $1/2$. Thus, the winning probability of P is at least $p/2$, as stated. □

As a direct corollary from Claim 3.3 and the fact that there is no strategy for the unique-ball game with a winning probability better than q/ℓ^2, we get that every strategy for the (ℓ, q)-multi-ball game wins with probability at most $2q/\ell^2$.

Corollary 3.9. *The winning probability of any strategy for the (ℓ, q)-multi-ball game is at most $2q/\ell^2$.*

We can now conclude Claim 3.4.

Proof of Claim 3.4. Immediately follows by Corollary 3.9 and Claim 3.7. □

3.3 Proving Claim 3.7

Lastly, we show that an adversary E that makes q queries to F and breaks Protocol 3.2 with probability p, can be used to win the $(\ell, q+1)$-multi-ball game with roughly the same probability.

To prove Claim 3.7, consider the following player P that plays the multi-ball game by simulating E on a random function F and transcript T. In the following, assume without loss of generality that E never asks the same query twice. Moreover, at the cost of one additional query, assume that E always queries F on its output.

Algorithm 3.10 (P)**.**

Parameters: $\ell, M \in \mathbb{N}$.

Operation:

1. P *samples* $a_1, \ldots, a_\ell \leftarrow [M]$ *uniformly and independently at random, and* $j \leftarrow [\ell]$.
2. P *simulates* E *on the transcript* $\mathsf{trans} = ((a_1, \ldots, a_\ell), z = a_j)$. *Every time* E *makes a query* (i, j), P *opens the box at location* (i, j). P *answers to* E*'s query as follows:*
   - *If the box contains a red ball,* P *wins and halts.*
   - *If the box is empty,* P *answers with a random element from* $[M]$.
   - *If the box contains a blue ball,* P *answers with a random element from* $\{a_1, \ldots, a_\ell\} \setminus \{a_j\}$ *that was not in use before.*

[3] notice that P may find ℓ blue balls in the simulated game, and this cannot happen in the real multi-ball game. However, this event is not counted as a win

We are now ready to prove Claim 3.7.

Proof of Claim 3.7. Let E be a $(q+1)$-query algorithm that, given the transcript of Protocol 3.2, queries the function F on the output O_{A} of A with probability at least p. Let P be Algorithm 3.10.

We start with considering a modified version of Protocol 3.2 (which cannot be implemented as a protocol), in which B does not notify A in step 4 in case B sees a collision in the function f. Instead, in this case we assume that B sends the correct image $z = f(j, x_j)$, and that A outputs the correct intersection (j, x_j).

As the above modified version and Protocol 3.2 differ only in the case that there is a collision in f, it is not hard to see that in this modified version, E guesses the output of party A with probability at least

$$\begin{aligned}
&p - \Pr\big[F(q) = F(q') \textit{ for two distinct queries } q \neq q' \textit{ of } \mathsf{A} \textit{ and } \mathsf{B}\big] \\
&\geq p - \binom{2\ell}{2} \cdot \frac{1}{M} \\
&\geq p - \frac{2\ell^2}{M},
\end{aligned}$$

where the first inequality holds by the union bound, since every two queries have probability $1/M$ to collide, and there are at most 2ℓ queries made by the parties.

To see that the success probability of Algorithm 3.10 is $p - \frac{2\ell^2}{M}$, note that it readily follows by inspection that the distribution seen by E in a random execution of Algorithm 3.10 on a random multi-ball game, is exactly the same distribution seen by E in a random execution of the above modified protocol. Moreover, Algorithm 3.10 finds the red ball in the multi-ball game if and only if E queries the secret key in the modified protocol. The claim follows. □

Acknowledgments

We thank Tal Yankovitz and Jad Silbak for very useful discussions.

References

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. "Public-key cryptography from different assumptions". In: *Proceedings of the forty-second ACM symposium on Theory of computing*. 2010, pp. 171–180 (cit. on p. 1).

[AD97] Miklós Ajtai and Cynthia Dwork. "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence". In: *stoc29*. See also ECCC TR96-065. 1997, pp. 284–293 (cit. on p. 1).

[Ale03] Michael Alekhnovich. "More on average case vs approximation complexity". In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE. 2003, pp. 298–307 (cit. on p. 1).

[BCNHR22] Andrej Bogdanov, Miguel Cueto Noval, Charlotte Hoffmann, and Alon Rosen. "Public-Key Encryption from Homogeneous CLWE". In: *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part II*. Springer. 2022, pp. 565–592 (cit. on p. 1).

[BDL01] Dan Boneh, Richard A DeMillo, and Richard J Lipton. "On the importance of eliminating errors in cryptographic computations". In: *Journal of cryptology* 14 (2001), pp. 101–119 (cit. on p. 4).

[BKSY11] Zvika Brakerski, Jonathan Katz, Gil Segev, and Arkady Yerukhimovich. "Limits on the power of zero-knowledge proofs in cryptographic constructions". In: *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*. Springer. 2011, pp. 559–578 (cit. on p. 4).

[BM09] Boaz Barak and Mohammad Mahmoody. "Merkle puzzles are optimal—an $O(n^2)$-query attack on any key exchange from a random oracle". In: *Annual International Cryptology Conference*. Springer. 2009, pp. 374–390 (cit. on pp. 2, 4).

[BV17] Nir Bitansky and Vinod Vaikuntanathan. "A note on perfect correctness by derandomization". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2017, pp. 592–606 (cit. on p. 4).

[DH76] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography". In: *IEEE Transactions on Information Theory* (1976), pp. 644–654 (cit. on p. 1).

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. "Immunizing encryption schemes from decryption errors". In: *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer. 2004, pp. 342–360 (cit. on p. 4).

[ElG84] Taher ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". In: *Annual International Cryptology Conference (CRYPTO)*. 1984, pp. 10–18 (cit. on p. 1).

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. "Eliminating decryption errors in the Ajtai-Dwork cryptosystem". In: *Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17*. Springer. 1997, pp. 105–111 (cit. on p. 4).

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for all One-Way Functions". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC)*. 1989, pp. 25–32 (cit. on p. 5).

[HMORY19] Iftach Haitner, Noam Mazor, Rotem Oshman, Omer Reingold, and Amir Yehudayoff. "On the Communication Complexity of Key-Agreement Protocols". In: *10th Innovations in Theoretical Computer Science* (2019) (cit. on pp. 2, 3).

[HOZ16] Iftach Haitner, Eran Omri, and Hila Zarosim. "Limits on the usefulness of random oracles". In: *Journal of Cryptology* 29.2 (2016), pp. 283–335 (cit. on p. 4).

[HR05] Thomas Holenstein and Renato Renner. "One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption". In: *Annual International Cryptology Conference*. Springer. 2005, pp. 478–493 (cit. on p. 4).

[IR89] Russell Impagliazzo and Steven Rudich. "Limits on the provable consequences of one-way permutations". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC)*. ACM Press, 1989, pp. 44–61 (cit. on pp. 2, 4).

[KSS00] Jeff Kahn, Michael Saks, and Cliff Smyth. "A dual version of Reimer's inequality and a proof of Rudich's conjecture". In: *15th Annual IEEE Conference on Computational Complexity*. 2000, pp. 98–103 (cit. on p. 3).

[LT13] Huijia Lin and Stefano Tessaro. "Amplification of chosen-ciphertext security". In: *Advances in Cryptology–EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings 32*. Springer. 2013, pp. 503–519 (cit. on p. 4).

[McE78] Robert J McEliece. "A public-key cryptosystem based on algebraic". In: *Coding Thv* 4244 (1978), pp. 114–116 (cit. on p. 1).

[Mer78] Ralph C Merkle. "Secure communications over insecure channels". In: *Communications of the ACM* 21.4 (1978), pp. 294–299 (cit. on p. 2).

[Pro03] John Proos. "Imperfect decryption and an attack on the NTRU encryption scheme". In: *Cryptology ePrint Archive* (2003) (cit. on p. 4).

[Rab79] Michael O Rabin. *Digitalized signatures and public-key functions as intractable as factorization*. Tech. rep. Massachusetts Inst of Tech Cambridge Lab for Computer Science, 1979 (cit. on p. 1).

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40 (cit. on p. 1).

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126 (cit. on p. 1).

[Sho99] Peter W Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332 (cit. on p. 1).