Limits on the Power of Private Constrained PRFs

Mengda Bi*, Chenxin Dai*, Yaohua Ma*

*Tsinghua University
*{bmd22, daicx21, ma-yh21}@mails.tsinghua.edu.cn

Abstract. Private constrained PRFs are constrained PRFs where the constrained key hides information about the predicate circuit. Although there are many constructions and applications of PCPRF, its relationship to basic cryptographic primitives, such as one-way functions and public-key encryptions, has been unclear. For example, we don't know whether one-way functions imply PCPRFs for general predicates, nor do we know whether 1-key secure PCPRF for all polynomial-sized predicates imply public-key primitives such as public-key encryption and secret-key agreement.

In this work, we prove the black-box separation between a 1-key secure PCPRF for any predicate and a secret-key agreement, which is the first black-box separation result about PCPRF. Specifically, we prove that there exists an oracle relative to which 1-key secure PCPRFs exist while secret-key agreement does not. Our proof is based on the simulation-based technique proposed by Impagliazzo and Rudich (STOC 89). The main technical challenge in generalizing the simulation-based technique to PCPRF is the issue of *unfaithfulness* of Eve's simulation to the real world because our oracle is more complicated than a random oracle. We introduce a new technique which we call the "weighting" technique and show how to leverage it to circumvent the issue of unfaithfulness in the proof framework of Impagliazzo and Rudich.

1 Introduction

1.1 Private Constrained Pseudorandom Functions

Constrained pseudorandom functions (CPRFs) are useful building blocks in cryptography [6, 8, 19]. Suppose $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a pseudorandom function (PRF), F is called a CPRF if it additionally exhibits the following functionality: given a master key $k \in \mathcal{K}$, and a binary predicate $c : \mathcal{X} \to \{0, 1\}$, one can derive a constrained key $k[c]$. $k[c]$ can be used to evaluate $F(k, x)$ if $c(x) = 1$. However, if $c(x) = 0$, $k[c]$ should not reveal non-trivial information of $F(k, x)$.

Since the concept of CPRF was first proposed, there have been constructions of CPRF for various predicate classes under various assumptions. For example, CPRF for the simplest predicate, a point indicator function, also known as puncturable PRF, is shown to be constructable from one-way functions (OWFs) using

the classical GGM tree-based PRF [6, 8, 16, 19]. Currently, the most expressive predicate family that can be constructed from OWF are the CNF predicates [13]. Other constructions require structural assumptions such as LWE [11], DDH [2], pairing [5,6], or even stronger primitives such as indistinguishability obfuscation (iO) [3, 5, 7, 13, 17].

A CPRF can further be *private* [5], if the constrained key k_c reveals no information about the predicate c. A private CPRF is also abbreviated as PCPRF. The notion of privacy varies depending on how many constrained keys the adversary is allowed to obtain. Multi-key secure PCPRF for $\mathsf{P/poly}$ predicates is only known to be constructible from iO [5]. In fact, even 2-key secure PCPRF for a predicate class implies iO for the same circuit class [12]. To the best of our knowledge, without iO, multi-key privacy is only achievable for left-right predicates in the random oracle model from bilinear maps [6]. In contrast, PCPRF with 1-key privacy for all polynomial-sized predicates is known to be constructable from the LWE assumption [10].

Even with just 1-key privacy, PCPRF still has wide applications in cryptography because of its powerful predicate-hiding ability. Known applications include searchable encryption, watermarking, deniable encryption and others in [5], private information retrieval [21, 24], and 1-key secure private-key functional encryption [12]. However, even though PCPRFs seem to be powerful, we still do not know how to construct secret-key agreement—the fundamental primitive in Cryptomania—from 1-key secure PCPRFs. Therefore, it is natural to ask whether 1-key secure PCPRFs fall within Minicrypt or Cryptomania.

1.2 Black-Box separation

In cryptography, it is almost impossible to directly prove separation results in the form of "the existence of a certain primitive P is not sufficient to construct another primitive Q". Instead, great progress has been made in proving the so-called "black-box separations". A black-box separation refers to a cryptographic impossibility result showing that Q cannot be achieved using only black-box access to P. In other words, it demonstrates that there exists no efficient reduction from Q to P when the reduction is restricted to using P as a black-box, without any knowledge of its internal structure. This concept is fundamental in understanding the limitations of cryptographic primitives and provides insights into the relationship between different cryptographic primitives.

One of the most significant black-box separation results is the seminal work of Impagliazzo and Rudich [18]. It shows that OWF cannot be used to construct secret-key agreement (KA) in a black-box way, employing a relativizing technique in which an oracle is constructed such that OWFs exist but KA does not, relative to this oracle. Their result naturally establishes a black-box separation between the following two classes: (1) Minicrypt: comprising primitives that are implied by OWFs, such as pseudorandom generators and pseudorandom functions; (2) Cryptomania: comprising all primitives that imply KA in a black-box way, such as public key encryption and oblivious transfer.

Inspired by [18], researchers have developed black-box separation techniques to obtain various impossibility results, including [14, 15, 22, 25]. In this work, we mainly focus on the impossibility results concerning KA. Significant progress has been made in this field by many researchers: [4] improves the efficiency of the adversary in [18], and [9] simplifies the analysis for perfectly complete KA protocols.

1.3 Motivations and Questions

Although significant work has been devoted to the study of PCPRF, its relationship with OWFs and cryptographic primitives in Cryptomania remains unclear. Firstly, we do not know how to use OWFs to construct PCPRFs for more expressive predicate classes than CNF (even providing only 1-key security). Secondly, we do not know how to use 1-key secure PCPRFs to construct any primitives in Cryptomania.[1]

Naturally, we aim to establish black-box separation results. Specifically, we pose the following questions:

- Open question 1: *Can we establish or refute black-box separation results between OWFs and more expressive PCPRFs?*
- Open question 2: *Can we establish or refute black-box separation results between 1-key secure PCPRFs and primitives in Cryptomania?*

1.4 Our Result

In this work, we prove the black-box separation between 1-key PCPRF for arbitrary predicate classes and KA, which gives a positive result regarding the second open question. Concretely, we prove the following theorem:

Theorem 1 (Informal). *There exists an oracle relative to which 1-key PCPRF exists while any KA protocol can be broken. This implies that it is impossible to construct KA using PCPRF in a black-box manner.*

Here, it is worth mentioning the predicate family of the PCPRF relative to our oracle. Specifically, what we care is that whether the predicates are allowed to access the oracle. We use the setting of oracle-aided circuits proposed in [1] in which the circuits are only allowed to access a random oracle H but not the rest of the oracle. Under this setting, we can even rule out some of the reductions that use primitives in minicrypt in a non-black-box way. As stated in [1], this setting characterizes most of the non-black-boxes constructions such as punctured programming approach of Sahai and Waters [23] and its variants.

[1] In [12] authors construct indistinguishability obfuscators for arbitrary circuits from 2-key secure PCPRF. Since we can construct public key encryption from iO and OWF [23], we already obtain a **non-black-box** construction of primitives in Cryptomania from 2-key secure PCPRF.

Our proof framework follows a similar approach to that of [18]. However, proving our result based on the techniques in [18] is not straightforward. The main challenge is explained in Section 2, where we introduce a new technique to overcome this issue.

2 Technical Overview

In this section, we first review the techniques from [18]. Then, we identify the main reason why the proof in [18] does not directly apply to our problem. Finally, we briefly outline our solution to this obstacle.

2.1 Techniques of [18]

In [18], it is proved that there exists an oracle relative to which a one-way function exists, but KA does not. The oracle consists of two parts: a random function f and a PSPACE-complete oracle. Clearly, f itself is one-way even when the PSPACE-complete oracle is present. Now we review how the attacker Eve in [18] breaks the KA protocol under such an oracle. Suppose Alice and Bob are the two parties of a KA protocol; Eve tries to find all the intersection queries—queries made by both Alice and Bob. It is shown in [18] that finding all the intersection queries suffices to break the KA protocol. During the attack, Eve maintains a list of query-answer pairs, recording the queries and oracle answers she has received so far. Eve's algorithm repeats the following two phases (which is called one "segment") polynomially many times after each message is sent.

- **Simulation Phase**: Eve uniformly samples a simulated view of Alice that is consistent with the transcript and Eve's list (if Bob has just sent a message, she samples Bob's run instead) using the PSPACE-complete oracle. Any simulated query not on Eve's list receives a random answer. In general, such a random answer may differ from the actual one.
- **Update Phase**: For each query in the simulated run, Eve queries the actual oracle f and adds the query-answer pair to her list.

Suppose that Alice has just sent a message. We temporarily fix the view of Bob, the transcript and Eve's list, while only Alice's view is considered unknown. Let BPQ denote the set of "Bob's private queries", which are the queries made by Bob but not Eve. Clearly, the size of BPQ is bounded by the total number of queries made by Bob. We classify each simulated view of Alice into the following two types:

1. **Hit**: this simulated view contains at least 1 query in BPQ.
2. **Miss**: this simulated view does not contain any queries in BPQ.

By the above definition, each hit contains at least one query in BPQ. Then in the following update phase, these queries will be made by Eve. This means the size of BPQ decreases by at least 1 in this segment. Since the size of BPQ is

bounded, the number of hits is also bounded. Thus, as long as Eve repeats the simulation sufficiently many times, most of the simulations will be misses.

The key to the proof in [18] is that, conditioned on a miss, the probability distribution of Alice's view in the simulation matches the distribution of her actual view. In other words, Eve's simulation of Alice's view must be faithful to the real one, given a "miss". The high-level idea behind this lemma is that if Alice's simulated run results in a miss, then the oracle queries made by Alice and Bob can only intersect at positions already known to Eve. Consequently, the random oracle f does not provide any common information that is known to both Alice and Bob but hidden from Eve.

Since the transcript and Bob's view is fixed, the query that will be made by Bob in the next round is also fixed. If Bob, in the next round, makes an intersection query with high probability, a significant portion of Alice's simulated view would already include this query. As a result, Eve has a high probability of identifying the intersection query. Therefore, even before a query becomes an intersection query, it is highly likely that Eve predicts it and adds it to her list.

2.2 Limitation of [18]

Now we consider how to apply the technique from [18] to our problem of separating 1-key secure PCPRF and KA. For now, we only consider predicates that are not allowed to access the oracle. In other words, they are just binary circuits in the common sense. As it turns out, this is already non-trivial.

The first step is to construct an oracle relative to which PCPRF exists. A natural approach is to define the oracle in the shape of a PCPRF. Let K, K', C be the spaces of master keys, constrained keys and predicates, respectively, and let X, Y be the spaces of inputs and outputs of the PRFs, we then define the oracle to consist of the following components:

- $\mathsf{F} : K \times X \to Y$ serves as the PRFs for master keys;
- $\mathsf{f} : K' \times X \to Y$ serves as the PRFs for constrained keys;
- $\mathsf{Keygen} : K \times C \to K'$ serves as the constrained key generation algorithm (henceforth we denote $k[c] = \mathsf{Keygen}(k, c)$).
- PSPACE-complete oracle, as usual.

The above oracle construction actually implements a multiple-key secure PCPRF. Since 2-key secure PCPRF and OWF implies KA, we cannot hope to prove that KA does not exist under such an oracle. In order to capture one-key security, a natural idea is to add another recovery oracle R with the following functionality: given two different constrained keys $k[c_1]$ and $k[c_2]$ of the same master key k, $\mathsf{R}(k[c_1], k[c_2])$ will return k, c_1, c_2. In any other cases, R returns $\perp$. For the R oracle to be well-defined, every element $k' \in K'$ must correspond to **at most one** pair $(k, c) \in K \times C$ satisfying the relation $k' = k[c]$. In other words, the oracle Keygen should be injective.

The correctness of PCPRF states that $\mathsf{F}(k, x) = \mathsf{f}(k[c], x)$ for $c(x) = 1$. This correctness property naturally divides all possible F, f queries into many

equivalent classes, and the answer for each equivalent class is sampled uniformly and independently from each other.

Now, we can see the first limitation of [18]. The major idea behind [18] is that: the only way that two parties achieve key agreement in the presence of Eve is essentially by making the same queries to the oracle. However, when we are considering a more sophisticated oracles like Γ instead of a simple random oracle, Alice and Bob can reach an agreement even if they do not make the same queries at all. Consider the following protocol:

- Alice samples $k \in K, c \in C, x \in X$ such that $c(x) = 1$. Then Alice queries $k' = k[c] = \mathsf{Keygen}(k, c)$ and sends k', x to Bob. The output of Alice is $\mathsf{F}(k, x)$.
- Bob receives k', x from Alice and outputs $\mathsf{f}(k', x)$.

The correctness of this key agreement protocol is guaranteed by the correctness of PCPRF: $\mathsf{F}(k, x) = \mathsf{f}(k[c], x)$. However, they make completely different queries.

To address this issue, we consider "intersection elements" instead of intersection queries. We say an element in $K \cup K' \cup C \cup X$ appears in a party's view if this element is a part of the input or output of a query in this party's view. Here, we do not consider elements in Y because they cannot appear as the input of a query. Analogous to intersection queries, intersection elements are elements that appears in both Alice and Bob's views. Our attacker Eve's goal is to find all the intersection elements. In the above example, Alice's view contains k, c, x, k'; Bob's view contains k', x. Thus, k', x are both intersection elements. If Eve finds k', x, she can make the query $\mathsf{f}(k, x)$ and the answer would be the agreed key.

However, the above problem is not the major challenge we faced trying to apply the technique of [18] to our case. The main reason is that the distribution of Alice's view will depend on Bob's view, even conditioned on miss. Since Bob's view is invisible to Eve, such dependency cannot be captured by Eve, which causes Eve's simulation to be unfaithful to the real Alice's view. In the following, we use a few examples to demonstrate why the distribution of Alice's simulated view can be unfaithful to Alice's actual view even conditioned on "miss".

- **Example 1**: Consider the following 2-pass protocol. In the first pass, Bob randomly samples $k_B \in K$ and queries $\mathsf{F}(k_B, i) = y_B^i$ for $i = 1, 2, \ldots, n$ where n is a sufficiently large polynomial (such that $|K'|^{-1} > |Y|^{-n}$). No communication between Alice and Bob is made in this round. In the next pass, Alice randomly samples $k'_A \in K'$ and queries $\mathsf{f}(k'_A, i) = y_A^i$ for $i = 1, 2, \ldots, n$. Now we fix Bob's view, i.e. k_B and $y_B^i, i \in [n]$. Let E be the event that $y_A^i = y_B^i$ for all $i \in [n]$. In the real execution, as long as $k' = k[c]$ where $c(x) = 1$ for $x \in [n]$, we would have $y_A^i = y_B^i$ for all $i \in [n]$. The probability of Alice picking $k' = k[c]$ for such c is at least $|K'|^{-1}$. Thus, the probability of E happening in the real execution is at least $|K'|^{-1}$. However, since Eve does not have Bob's view, the probability that $y_A^i = y_B^i$ for all $i \in [n]$ in Eve's simulation is $|Y|^{-n}$. Thus, as long as n is sufficiently large, we would notice that the event E is more likely to happen in the real execution than in

Eve's simulation, which means that Eve's simulation is unfaithful to Alice's actual view.

– **Example 2**: Consider the following 2-pass protocol. In the first pass, Bob randomly samples $k_B \in K, c_B \in C$ and queries $\mathsf{Keygen}(k_B, c_B) = k'_B$. In the second pass, Alice randomly samples $k_A \in K$ and $c_A \in C$ and queries $\mathsf{Keygen}(k_A, c_A) = k'_A$. Fix the view of Bob. Let E be the event that $k_A \neq k_B$ and $k'_A = k'_B$. Because Keygen is sampled from injective functions, the probability of E is 0 in the real execution. However, in the distribution of Eve's simulated view of Alice, event E has a non-zero probability. This means that Eve's simulation is unfaithful to Alice's actual view.

Examples 1 and 2 show that the distribution of Alice's actual view and the distribution of Eve's simulated view of Alice might have very different probabilities on some specific "miss" views. In other words, the distribution of Eve's simulation is unfaithful to Alice's actual view conditioned on "miss". Note that even with the recovery oracle R, these two examples can still happen.

To overcome this issue of unfaithfulness, we will modify the proof of [18]. In the proof of [18], the most important step is to prove that Eve's simulation is faithful to Alice's actual view conditioned on "miss", for any **fixed** tuple of Bob's view, Eve's view and the transcript so far. We denote such tuple as G. In our proof, we will analyze the statistical distance between Eve's simulation and Alice's actual view, conditioned on "miss". We show that the expectation of this statistical distance is negligible over a specific distribution of G, which will be enough under our proof framework. To do so, we introduce a new technique which we call the **weighting** technique. This is further discussed in the next section.

2.3 Our Approach

In this section we provide a high level overview of our approach. We first give a simplified version of our oracle.

Oracle description. Let K, K', C, X, Y be the space of master keys, constrained keys, predicates, the input and the output of the PCPRF, respectively. The size of these spaces are set as follows: $|K| = |C| = |X| = 2^\kappa, |K'| = 2^{\kappa^2}, |Y| = 2^{\kappa^3}$. Here, the elements in C are viewed as P.P.T.Ms that have access to H and the PSPACE-complete oracle. Then, our oracle contains two layers. The first layer is a random oracle $\mathsf{H} : Z = \{0,1\}^* \to \{0,1\}$. The second layer consists of the following components:

– $\mathsf{F} : K \times X \to Y$ is uniformly sampled from all functions.
– $\mathsf{Keygen} : K \times C \to K'$ is sampled from all injective functions. Henceforth we denote $k[c] = \mathsf{Keygen}(k, c)$
– $\mathsf{f} : K' \times X \to Y$ is defined as follows: if $k' = k[c]$ and $c(x) = 1$, then $\mathsf{f}(k', x) = \mathsf{F}(k, x)$. Otherwise it is randomly sampled from Y.

– $\mathsf{R} : (K \times K') \times K' \to C \cup (K \times C \times C) \cup \{\perp\}$ is defined as follows: if the input is $(k, k[c])$ for some $k \in K, c \in C$, then R returns c; if the input is $(k[c_1], k[c_2])$ for some $k \in K, c_1, c_2 \in C$ and $c_1 \neq c_2$, R returns (k, c_1, c_2); in any other cases, R returns $\perp$ as a sign of rejection.

We denote the whole oracle as $\Gamma = (\mathsf{H}, \mathsf{F}, \mathsf{Keygen}, \mathsf{f}, \mathsf{R})$. Here, we remark that, in our formal definition, the oracle is defined in a more complicated way: the second layer of Γ is sampled for each input length κ. For simplicity, we only consider a fixed input length in this overview.

The Weighting Technique Before proceeding to our attacking algorithm, we first introduce the weighting technique that is used to address the unfaithfulness issue raised in the previous section. We start by redefining the concept of views by introducing artificial views that might contain collisions. Generally speaking, a view is what can be seen by a party (or parties). It includes the randomness of the party, the transcript and the query answer pairs made by the party. Recall that the Keygen oracle is sampled from all injective functions, collisions will never happen in a view. However, we allow a view to contain collisions such as $k_1[c_1] = k_2[c_2] = k'$ where $k_1 \neq k_2$. (Here, by collisions, we do not specifically mean the collision of Keygen queries. R queries can also be a part of collisions. For example, the two query-answer pairs such as $\mathsf{Keygen}(k_1, c_1) = k', \mathsf{R}(k_2, k') = c_2$ also constitute a collision.) That being said, we do not wish a view to contain arbitrary collisions. We say a view V is "valid" if it satisfies the following 2 properties:

1. If two master keys k_1 and k_2 collide at k', that is $k_1[c_1] = k_2[c_2] = k'$ for some c_1, c_2, then there will be no other collisions related to k_1 or k_2 in V.
2. There are no self-collisions, i.e. $k[c_1] = k[c_2]$ for $c_1 \neq c_2$.

To better understand the first property, we here give two counterexamples which violate it: (1), one master key simultaneously collides with two other master keys: $k_1[c_1] = k_2[c_2], k_1[c_1'] = k_3[c_3]$; (2), two master keys collide at two different constrained keys: $k_1[c_1] = k_2[c_2], k_1[c_1'] = k_2[c_2']$.

We now define the weight of a valid view. Given a valid view V and a specific instantiation H of the oracle H, define $w_H(V)$ as follows:

– If the H queries made in V are inconsistent with H, $w_H(V) = 0$.
– If the correctness of PCPRF (with H instantiated as H) is violated in V, $w_H(V) = 0$.
– Otherwise, define $w_H(V)$ as follows: each different constrained relation $k[c] = k'$ in V contributes a factor of $|K'|^{-1} = 2^{-\kappa^2}$; each different F or f queries contribute a factor of $|Y|^{-1} = 2^{-\kappa^3}$ However, those F or f queries that can be deduced to be the same from the correctness of PCPRF only count as one contribution. $w_H(V)$ is defined to be the product of all contributions.

Then, the weight of a view V, denoted as $w(V)$, is defined as follows

$$w(V) = \frac{1}{2^M} \sum_H w_H(V)$$

Here, the summation is taken over all possible instantiations H. M is the size of the possible input space of H. Indeed, the input of H can be arbitrarily long. However, since all parties are P.P.T.Ms, they cannot make H queries with arbitrarily long inputs. Therefore we can view the input space of H as finite. Then 2^M is the number of possible instantiations of H, which means that $w(V)$ can be viewed as the average value of $w_H(V)$.

We now define two distributions $\mathcal{D}$ and $\mathcal{D}'$ over all valid views. Define $\mathcal{D}$ as the distribution of views generated by first randomly sampling the randomness of all three parties (Alice, Bob and Eve) and oracle Γ, then executing the protocol. $\mathcal{D}'$ is a distribution over all valid views. The probability of a valid view V in $\mathcal{D}'$ is proportional to V's weight $w(V)$. Then, we will prove that the statistical distance between $\mathcal{D}$ and $\mathcal{D}'$ is negligible, which is a key lemma in our analysis. We refer the readers to Appendix B for more information.

Our Attacking Algorithm. Now we consider how to break KA protocols relative to this oracle. Our attacker Eve is similar to [18]. She locally maintains a list of elements in $K \cup K' \cup C \cup X \cup Z$ and repeats the two phases (i.e. a segment) mentioned in Section 2.1 after each message is sent. Suppose Alice speaks in round $r-1$ and Bob speaks in round r. Then Eve's algorithm in round r will repeat the following two phases sufficiently many times:

- **Simulation Phase**: Eve samples a view of Alice up until the end of round $r-1$, denoted as Sim_A. Sim_A should be consistent with Eve's view and the transcript so far. Furthermore, Sim_A combined with Eve's view V_E (denoted as $\mathsf{Sim}_A \cup V_E$) should form a collision-free valid view. The probability that Sim_A is sampled is proportional to $w(\mathsf{Sim}_A \cup V_E)$.
- **Update Phase:** Eve will add all the elements in the sampled view Sim_A to her list. Then, Eve will try to find out all the information related to the elements in her list. For every pair of $k \in K, c \in C$ in her list, Eve will query $\mathsf{Keygen}(k, c)$ to find out $k[c]$ which is automatically added to Eve's list. For every pair of $k \in K$ (resp. $k' \in K'$) and $x \in X$, Eve will query $\mathsf{F}(k, x)$ (resp. $\mathsf{f}(k', x)$). For all master keys and constrained keys in Eve's list, Eve will use the recovery oracle R to verify whether there are some hidden relations among them.

We now show that our attacking algorithm address the unfaithfulness issue raised in Section 2.2. We will show that, for any fixed Bob and Eve's view V_B, V_E, if Sim_A is a "miss" simulation, that is, the intersection elements between Sim_A and V_B all appear in V_E, then the combined view $\mathsf{Sim}_A \cup V_E \cup V_B$ is a valid view which might contain collisions as discussed before. This is exactly why we allow a valid view to contain collisions. Furthermore, we will show that $w(\mathsf{Sim}_A \cup V_E)$ is proportional to $w(\mathsf{Sim}_A \cup V_E \cup V_B)$. $w(\mathsf{Sim}_A \cup V_E)$ is proportional to the probability that Sim_A is sampled. By the definition of $\mathcal{D}'$, $w(\mathsf{Sim}_A \cup V_E \cup V_B)$ is proportional to the probability of Alice's view being Sim_A in $\mathcal{D}'$ conditioned on V_B , V_E and the transcript so far. This means, Eve's simulation is actually faithful to Alice's view in $\mathcal{D}'$, conditioned on "miss". After that, we use the result

$\mathcal{D} \approx \mathcal{D}'$ to show that Eve's simulation is also faithful to Alice's view in $\mathcal{D}$ with high probability. Then, we can apply the technique of [18] to our problem.

2.4 Paper Organization

The rest of our paper is organized as follows. In Section 3, we give a formal definition of PCPRF. In Section 4, we give a formal definition of our oracle Γ and prove that 1-key secure PCPRF exists under such an oracle Γ. In Section 5, we provide formal definitions of secret-key agreement protocols and views, then define the distribution $\mathcal{D}$. Section 6 is the foundation of our technique. In this section, we give the formal definition of valid views, the weights of valid views, and the distribution $\mathcal{D}'$. Then, we prove our key lemma $\mathcal{D} \approx \mathcal{D}'$. In Section 7, we present the algorithm of Eve that breaks any KA protocol, and our main black-box separation result.

3 Preliminaries

A non-negative function $\mu : \mathbb{N}^* \to \mathbb{R}^+ \cup \{0\}$ is called **negligible** if $\mu(n) = O\left(\frac{1}{n^c}\right)$ for any constant $c \in \mathbb{N}^*$, denoted by $\mu = \mathsf{negl}(n)$.

In this work, we use a notion stronger than negligible: a non-negative function $\mu : \mathbb{N}^* \to \mathbb{R}^+ \cup \{0\}$ is **super-negligible** if $\mu(n) = O\left(\exp\left(-n^{1+\epsilon}\right)\right)$ for some constant $\epsilon > 0$, denoted by $\mu = \mathsf{snegl}(n)$. In the proof, we will use the following fact: $\mathsf{snegl}(\log n) = \mathsf{negl}(n)$.

We define two notations for simplifying our proof: $a(n) \approx b(n)$ means that $\left|\frac{a(n)}{b(n)} - 1\right| = \mathsf{negl}(n)$, and notation $a(n) \approx_s b(n)$ means that $\left|\frac{a(n)}{b(n)} - 1\right| = \mathsf{snegl}(n)$.

Given a probability distribution $\mathcal{S}$ over the set S, denote $\sup(\mathcal{S})$ as the support set of $\mathcal{S}$. That is, $\sup(\mathcal{S})$ is the set of elements in S that have non-zero probability in $\mathcal{S}$.

3.1 Statistical Distance

Definition 1. *Given two (discrete) probability distributions $\mathcal{S}_1$ and $\mathcal{S}_2$ over the set S, the statistical distance between them is defined as*

$$\Delta(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{2} \sum_{s \in S} \left| \Pr_{s' \leftarrow \mathcal{S}_1}[s' = s] - \Pr_{s' \leftarrow \mathcal{S}_2}[s' = s] \right|.$$

The statistical distance satisfies triangular inequality, i.e., for any distributions $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ over S, we have $\Delta(\mathcal{S}_1, \mathcal{S}_3) \leq \Delta(\mathcal{S}_1, \mathcal{S}_2) + \Delta(\mathcal{S}_2, \mathcal{S}_3)$. Another important fact is that statistical distance does not increase when applying a (possibly randomized) function f, i.e. $\Delta(\mathcal{S}_1, \mathcal{S}_2) \geq \Delta(f(\mathcal{S}_1), f(\mathcal{S}_2))$.

We now prove two lemmas about statistical distances that will be useful in our proof:

Lemma 1. *Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two distributions over S where $\Delta(\mathcal{S}_1, \mathcal{S}_2) = d$, $f : S \to T$ be an arbitrary function. Then we have*

$$\mathbb{E}_{s \leftarrow \mathcal{S}_1}[\Delta[(\mathcal{S}_1|f(s)), (\mathcal{S}_2|f(s))]] \leq 2d.$$

Here, $(\mathcal{S}_1|t)$ denotes the conditional distribution $\{s|s \leftarrow \mathcal{S}_1, f(s) = t\}$, the same holds for S_2.

Proof. We define two distributions over T:

$$\mathcal{T}_1 = \{f(s)|s \leftarrow \mathcal{S}_1\}, \mathcal{T}_2 = \{f(s)|s \leftarrow \mathcal{S}_2\}.$$

Then, because applying a function does not increase the statistical distance, $\Delta(\mathcal{T}_1, \mathcal{T}_2) \leq \Delta(\mathcal{S}_1, \mathcal{S}_2) = d$.

Define distribution $\mathcal{S}'$ over S as follows:

- First sample $t \leftarrow \mathcal{T}_1$.
- Then sample $s \leftarrow \mathcal{S}_2$ conditioned on $f(s) = t$, and output s.

Note that $\mathcal{S}_2$ can be sampled in the following way:

- First sample $t \leftarrow \mathcal{T}_2$.
- Then sample $s \leftarrow \mathcal{S}_2$ conditioned on $f(s) = t$, and output s.

We observe that the only difference between these two sampling processes is the first step, and the statistical distance of the first step is bounded by d. So we have $\Delta(\mathcal{S}_2, \mathcal{S}') \leq d$.

Then, we have $\Delta(\mathcal{S}_1, \mathcal{S}') \leq \Delta(\mathcal{S}_1, \mathcal{S}_2) + \Delta(\mathcal{S}_2, \mathcal{S}') \leq d + d = 2d$. This directly translates to

$$\mathbb{E}_{s \leftarrow \mathcal{S}_1}[\Delta[(\mathcal{S}_1|f(s)), (\mathcal{S}_2|f(s))]] \leq 2d.$$

□

Lemma 2. *Let $\mathcal{S}_1, \mathcal{S}_2$ be two distributions over S where $\Delta(\mathcal{S}_1, \mathcal{S}_2) = d$. $L \subseteq S$, $\Pr_{s \leftarrow \mathcal{S}_1}[s \in L] = p$. Then we have*

$$\Delta[(\mathcal{S}_1|L), (\mathcal{S}_2|L)] \leq \frac{2d}{p}.$$

Here, $\mathcal{S}_1|L$ denotes the conditional distribution $\{s|s \leftarrow \mathcal{S}_1, s \in L\}$, the same holds for $\mathcal{S}_2$.

Proof. Define $f : S \to \{0, 1\}$ as follows: $f(s) = 1$ if and only if $s \in L$. Then, by Lemma 1, we have

$$\mathbb{E}_{s \leftarrow \mathcal{S}_1}[\Delta[(\mathcal{S}_1|f(s)), (\mathcal{S}_2|f(s))]] \leq 2d.$$

Equivalently, we have

$$p\Delta[(\mathcal{S}_1|L), (\mathcal{S}_2|L)] + (1 - p)\Delta[(\mathcal{S}_1|S - L), (\mathcal{S}_2|S - L)] \leq 2d,$$

which implies $\Delta[(\mathcal{S}_1|L), (\mathcal{S}_2|L)] \leq \frac{2d}{p}$. □

3.2 Private Constrained PRF

In this work, we use a simulation-based definition of 1-key oracle-aided private constrained PRF (PCPRF), our notion is similar to [12] except that both adversary and simulator have access to an oracle $\mathcal{O}$.

Definition 2. *Consider a family of functions $\mathcal{F} = \{\mathcal{F}_\kappa\}_{\kappa\in\mathbb{N}^*}$, where $\mathcal{F}_\kappa = \{F_k : D_\kappa \to R_\kappa\}$ for $\kappa \in \mathbb{N}^*$, along with a triple of efficient functions $(\mathbf{Gen}, \mathbf{Constrain}, \mathbf{Eval})$ that take an oracle $\mathcal{O}$ as input. For a predicate class $C = \{C_\kappa : D_\kappa \to \{0,1\}\}_{\kappa\in\mathbb{N}^*}$: the key generation algorithm $\mathbf{Gen}(1^\kappa)$ generates the master secret key MSK; the constraining algorithm $\mathbf{Constrain}(\mathsf{MSK}, c)$ takes the master secret key MSK, a predicate c, outputs the constrained key CK; the evaluation algorithm $\mathbf{Eval}(k, x)$ takes a key k, an input x, outputs $F_k(x)$.*

We say that $\mathcal{F}$ is simulation-based 1-key secure PCPRF relative to oracle $\mathcal{O}$ for predicate class C if:

- ***Perfect Correctness:*** *For all $x \in D_\kappa, c \in C$ s.t. $c(x) = 1$, we have $\Pr[\mathbf{Eval}(\mathsf{MSK}, x) = \mathbf{Eval}(\mathsf{CK}, x)] = 1$ for $\mathsf{CK} = \mathbf{Constrain}(\mathsf{MSK}, c)$.*
- ***Simulation Indistinguishability:*** *For any oracle-aided P.P.T.M. adversary $\mathbf{Adv}^{\mathcal{O}}$, there exists an oracle-aided P.P.T.M. simulator $\mathbf{Sim}^{\mathcal{O}}$ such that:*

$$\left| \Pr_{\mathsf{MSK}, r_{\mathbf{Adv}}}[\mathbf{REAL}(1^\kappa, \mathbf{Adv}^{\mathcal{O}}, \mathsf{MSK}) = 1] - \Pr_{r_{\mathbf{Adv}}, r_{\mathbf{Sim}}}[\mathbf{IDEAL}(1^\kappa, \mathbf{Adv}^{\mathcal{O}}, \mathbf{Sim}^{\mathcal{O}}) = 1] \right| < \mathsf{negl}(\kappa), \tag{1}$$

*where experiments **REAL** and **IDEAL** are defined in Algorithm 1 and Algorithm 2.*

4 PCPRF Relative to Oracle Γ

4.1 Oracle Design

Our oracle consists of the following parts:

- A PSPACE-complete oracle.
- $\mathsf{H} : Z = \{0,1\}^* \to \{0,1\}$ is a random oracle, i.e., for arbitrary binary string z, $\mathsf{H}(z)$ is sampled from uniform distribution over $\{0,1\}$.
- For every $\kappa \in \mathbb{N}^*$, we denote PCPRF oracle of length κ as Γ_κ defined below:
  - $K_\kappa = X_\kappa = C_\kappa = \{0,1\}^\kappa, K'_\kappa = \{0,1\}^{\kappa^2}, Y_\kappa = \{0,1\}^{\kappa^3}$. Here elements of C_κ are viewed as arbitrary $\mathsf{P/poly}$ predicates with input length κ. They are allowed to query H and the PSPACE-complete oracle. However, they are only allowed to query H for at most κ times.
  - $\Gamma_\kappa = (\mathsf{F}_\kappa, \mathsf{f}_\kappa, \mathsf{Keygen}_\kappa, \mathsf{R}_\kappa)$ are four oracles sampled by the following way:
    1. $\mathsf{F}_\kappa : K_\kappa \times X_\kappa \to Y_\kappa$ is uniformly sampled from all functions.

Algorithm 1 REAL$(\mathbf{Adv}^{\mathcal{O}}, \mathsf{MSK})$

loop
  $\mathbf{Adv} \rightarrow (x, d_x)$
  $y \leftarrow \mathbf{Eval}(\mathsf{MSK}, x)$
  $\mathbf{Adv} \leftarrow y$
$\mathbf{Adv} \rightarrow c$
if $d_x \neq c(x)$ for some x **then**
  output $\perp$
else
  $\mathbf{Adv} \leftarrow \mathbf{Constrain}(\mathsf{MSK}, c)$
loop
  $\mathbf{Adv} \rightarrow x$
  $y \leftarrow \mathbf{Eval}(\mathsf{MSK}, x)$
  $\mathbf{Adv} \leftarrow y$
output $b \leftarrow \mathbf{Adv}$

Fig. 1: The Experiment **REAL**.

Algorithm 2 IDEAL$(\mathbf{Adv}^{\mathcal{O}}, \mathbf{Sim}^{\mathcal{O}})$

loop
  $\mathbf{Adv} \rightarrow (x, d_x)$
  $y \leftarrow \mathbf{Sim}(x, d_x)$
  if $d_x = 0$ **then**
    $y \xleftarrow{R} U$
  $\mathbf{Adv} \leftarrow y$
$\mathbf{Adv} \rightarrow c$
if $d_x \neq c(x)$ for some x **then**
  output $\perp$
else
  $\mathbf{Adv} \leftarrow \mathbf{Sim}(1^{|c|})$
loop
  $\mathbf{Adv} \rightarrow x$
  $y \leftarrow \mathbf{Sim}(x, c(x))$
  if $c(x) = 0$ **then**
    $y \xleftarrow{R} U$
  $\mathbf{Adv} \leftarrow y$
output $b \leftarrow \mathbf{Adv}$

Fig. 2: The Experiment **IDEAL**.

2. $\mathsf{Keygen}_\kappa : K_\kappa \times C_\kappa \to K'_\kappa$ is sampled from all **injective** functions. Denote $k[c] = \mathsf{Keygen}(k, c)$.
3. $\mathsf{f}_\kappa : K'_\kappa \times X_\kappa \to Y_\kappa$ is defined as follows: given $k' \in K'_\kappa, x \in X_\kappa$, if there exists $k \in K_\kappa, c \in C_\kappa$ s.t. $k' = k[c]$ and $c(x) = 1$, then $\mathsf{f}_\kappa(k', x) = \mathsf{F}_\kappa(k, x)$. The other locations of f_κ are sampled uniformly from Y_κ.
4. $\mathsf{R}_\kappa : (K_\kappa \cup K'_\kappa) \times K'_\kappa \to C_\kappa \cup (K_\kappa \times C_\kappa \times C_\kappa) \cup \{\perp\}$ is defined as follows: if the input is (k, k') where $k \in K'_\kappa$ and $k' = k[c]$, then $\mathsf{R}_\kappa(k, k') = c$. If the input is (k'_1, k'_2) where $k'_1, k'_2 \in K'_\kappa$ s.t. there exists $k \in K_\kappa$, $c_1, c_2 \in C_\kappa, c_1 \neq c_2$, $k'_1 = k[c_1], k'_2 = k[c_2]$, then $\mathsf{R}_\kappa(k'_1, k'_2) = (k, c_1, c_2)$. In any other cases, R_κ will return $\perp$ as a sign of rejection.

– We define oracle Γ to be the concatenation of Γ_κ for every length, H and the PSPACE-complete oracle, i.e., $\Gamma = (\text{PSPACE-complete oracle}, \mathsf{H}, \{\Gamma_\kappa\}_{\kappa \in \mathbb{N}^*})$, also define K, X, C, K', Y to be the concatenation of all corresponding fields.

Here we remark that the elements of the sets we defined before are different elements, even though they might have the same binary representation. For example, suppose $z \in Z$ is of length κ, then z can only be used as input of H queries. We cannot use z as the input of a Γ_κ query. Similarly, an element $k \in K_\kappa$ can only be used as the first input of F_κ or R_κ queries; an element $x \in X_\kappa$ can only be used as the second input of F_κ or f_κ queries, etc.

In the following, we will often omit the index κ of the oracles in Γ_κ. For example, we might simply write $\mathsf{F}(k, x)$ instead of $\mathsf{F}_\kappa(k, x)$. This does not introduce ambiguity because κ is fully determined by the length of the input.

4.2 Construction of PCPRF

Now we propose our 1-key secure PCPRF construction:

– **Gen**(1^κ): randomly sample a master key $\mathsf{MSK} \xleftarrow{R} K_\kappa$.
– **Constrain**(MSK, c): answer $\mathsf{Keygen}(\mathsf{MSK}, c)$.
– **Eval**(k, x): if k is a master key, i.e., $k \in K_\kappa$, return $\mathsf{F}(k, x)$; otherwise k is a constrain key, i.e., $k \in K'_\kappa$, return $\mathsf{f}(k, x)$.

4.3 Correctness

The perfect correctness of our scheme follows directly the functionality of oracle Γ: whenever $c(x) = 1$, we have

$$\mathbf{Eval}(\mathsf{MSK}, x) = \mathsf{F}(\mathsf{MSK}, x) = \mathsf{f}(\mathsf{Keygen}(\mathsf{MSK}, c), x) = \mathbf{Eval}(\mathsf{CK}, x).$$

4.4 Security

We prove that our PCPRF construction is 1-key simulation-secure relative to our oracle Γ. Formally, we will prove the following theorem:

Theorem 2. *For almost all oracles Γ, simulation indistinguishability inequality Equation (1) holds for any oracle-aided P.P.T.M. adversary $\mathbf{Adv}^{\Gamma}$, therefore the construction in Section 4.2 is a 1-key secure PCPRF for arbitrary predicate class relative to almost all oracles Γ.*

We defer the proof of Theorem 2 to Appendix A.

5 Additional Preliminaries

5.1 Secret-Key Agreement Protocol

Definition A secret-key agreement protocol consists of two (oracle-aided) P.P.T.M. called Alice and Bob. In addition, they have a common communication tape that both parties can read and write. A run of the protocol is as follows: Alice and Bob both start with input 1^{λ}, where λ is the security parameter; Alice and Bob run, communicating via the common tape; at last, Alice and Bob both output a string of length l, where $l = l(\lambda)$ is a polynomial over λ. If these two strings are the same, Alice and Bob are said to agree. The entire history of the writes to the communication tape is called the transcript. $\alpha(\lambda)$ will denote the probability that Alice and Bob agree on the same secret with security parameter λ.

We say (oracle-aided) P.P.T.M. Eve breaks a secret-key agreement protocol, if Eve, given only the transcript, can guess the secret with probability $1/\mathsf{poly}(\lambda)$ conditioned on Alice and Bob agree on the same secret. A protocol is secure if no P.P.T.M Eve can break it.

Normal Form In this section, we will convert arbitrary KA protocol into a protocol satisfying certain conditions, which we call the "normal form". There are many steps in this conversion. In each step, the protocol gains a certain restriction that will facilitate our analysis. The conversion does not compromise the security of the KA protocol and only causes polynomial blow-up in running time.

Restriction 1: Avoid querying Γ_κ with small κ Let λ be the security parameter of the key agreement protocol. The first step in our conversion is to eliminate any Γ_κ queries with $\kappa \leq \log \lambda$.

We first assume that all three parties have a preparation phase at the beginning of the protocol. In this phase, they make sufficient queries to H so that for every pair of (c, x) where $c \in C_\kappa, x \in X_\kappa, \kappa \leq \log \lambda$, the queries made in the preparation phase is enough to evaluate $c(x)$. For each $\kappa \leq \log \lambda$, $|C_\kappa| = |X_\kappa| = 2^\kappa \leq \lambda$. This means that there are only polynomially many such pairs. For each pair (c, x), to evaluate $c(x)$, at most $\log \lambda$ queries to H need to be

made. Thus each party only need to make polynomially many queries to H in the preparation phase. Clearly, adding the preparation phase will not compromise the security of the KA protocol.

Then, we assume that each party also recovers the whole Keygen_κ oracle in the preparation phase. Since the input of Keygen_κ is in the set $|K_\kappa| \times |C_\kappa|$ which has size $2^{2\kappa} \leq \lambda^2$, this step only takes polynomial time. Once Keygen_κ is known to every party, there is no need to query R_κ in Γ_κ, and all F_κ and f_κ queries are naturally divided into many equivalent classes by the correctness of PCPRF. The queries in each equivalent class have the same answer, and the answer for each equivalent class is individually and independently sampled from Y_κ. We can simulate the F_κ and f_κ oracle by a part of H that is never used in the original protocol. Thus, we may assume that F_κ and f_κ are never queried. As a result, only Keygen_κ is queried in Γ_κ.

Now, instead of letting each party recover Keygen_κ, we let Alice generate a "fake" Keygen_κ oracle and send it to Bob. Whenever a party needs to query Keygen_κ, they will instead use the fake Keygen_κ. As a result, each party will never access Γ_κ.

Restriction 2: Maintain the "completeness" with $\mathsf{F}, \mathsf{f}, \mathsf{Keygen}$ Queries We first formally define what is the meaning of the "completeness" of a view:

Definition 3. *We say a view is **complete** if for every κ and every pair of $c \in C_\kappa, x \in X_\kappa$ that appears in this view, the H queries made in this view are enough to evaluate $c(x)$.*

We are now ready to describe restriction 2 of the normal form. This restriction ensures that, if a party's view is complete before a $\mathsf{F}, \mathsf{f}, \mathsf{Keygen}$ query, then the view is also complete after this query. Specifically, suppose the next query of a party is $\mathsf{F}(k_0, x_0)$, $\mathsf{f}(k_0', x_0)$ or $\mathsf{Keygen}(k_0, c_0)$. Before making this query, this party will make sure that

- If the next query is $\mathsf{F}_\kappa(k_0, x_0)$ or $\mathsf{f}_\kappa(k_0', x_0)$, for every $c \in C_\kappa$ that has appeared in this party's view, enough H queries must be made in this view to evaluate $c(x_0)$.
- If the next query is $\mathsf{Keygen}_\kappa(k_0, c_0)$, for every $x \in X_\kappa$ that has appeared in this party's view, enough H queries must be made in this view to evaluate $c_0(x)$.

Restriction 3: Repair the completeness after R queries Suppose a party just made an R_κ query and the answer is not $\perp$. Then, the answer must contain 1 or 2 predicates. Denote them as c_1 (and c_2). This party will first make sure that, for every $x \in X_\kappa$ that has appeared in this view, enough H queries are made to evaluate $c_1(x)$ (and $c_2(x)$).

Note that restriction 2 of the normal form ensures that $\mathsf{F}, \mathsf{f}, \mathsf{Keygen}$ queries cannot break the completeness of a party's view. Clearly, H queries cannot break the completeness of a party's view. Thus, only R queries can break the completeness of a party's view. Indeed, if an R query yields an answer other than $\perp$, then

the answer might include 1 or 2 predicates. Denote them as c_1 (and c_2). If c_1 (and c_2) did not appear in this party's view, the completeness of a party's view might be temporarily broken. Restriction 3 ensures that the completeness of this party's view is fixed (after a number of H queries.)

Restriction 4: Find hidden constrained relations among the keys Suppose the next query of a party is $\mathsf{F}(k_0, x_0)$, $\mathsf{f}(k'_0, x_0)$ or $\mathsf{Keygen}(k_0, c_0)$, this party will first make sure that there are no hidden constrained relations among the master keys and constrained keys that have appeared in this view (along with k_0 or k'_0). Specifically, for each $k \in K_\kappa$ and $k' \in K'_\kappa$ that have appeared in this view (along with k_0 or k'_0), it must be clear in this view whether k' is a constrained key generated from k; for each $k'_1, k'_2 \in K'_\kappa$ appeared in this view (along with k'_0), it must be clear in this view whether they are the constrained keys generated from the same master key k. All of this can be achieved by the recovery oracle R.

The Normal Form Let $\kappa_{\max}(\lambda)$ be the maximum κ such that Alice or Bob might query Γ_κ. Clearly, $\kappa_{\max}$ is also a polynomial over λ. We assume that the protocol has $n = n(\lambda)$ rounds where $n(\lambda)$ is a polynomial over λ. Alice speaks in odd rounds and Bob speaks in even rounds. Suppose r is even, then in round r:

- Bob first does his own computation. Then he makes 1 arbitrary query. This query is later referred as the first query of round r. This query can be an H query or Γ_κ query for some $\kappa \geq \log \lambda$. If this query is not an H query, then the input contains two elements. We make the restriction that at least one of the two input elements has already appeared in Bob's view. In other words, only one can be new between the two input elements. Denote the new input element as $q_{r,1}$. (If neither input elements are new or the first query is an H query, let $q_{r,1}$ be the first input element.)
- After the first query, Bob will make $2n\kappa_{\max}$ H queries. Let $q_{r,j}, (2 \leq j \leq 2n\kappa_{\max} + 1)$ be the input of the jth query. Especially, if the first query of round r is an R query and the answer is not $\perp$, then the following $2n\kappa_{\max}$ H queries must be first used to make sure that the completeness of the view still holds as requested by restriction 3. ($2n\kappa_{\max}$ H queries are enough to do this: R query introduces at most 2 new predicates; there are at most n elements in X in Bob's view because at most 1 new elements in X appears in Bob's view in each round; to evaluate $c(x)$ where both c, x appeared in this view, it takes at most $\kappa_{\max}$ H queries. Thus $2n\kappa_{\max}$ H queries are enough.)
- Bob then sends one message Conv^r to Alice. Denote $M^r = (\mathsf{Conv}^1, \mathsf{Conv}^2, \ldots, \mathsf{Conv}^r)$ be the messages sent so far.

For odd r, we only need to change the role of Alice and Bob.

Moreover, before Alice and Bob output their secrets s_A, s_B, we add 2 rounds at the last of the protocol. In these two rounds, the first query of Alice and Bob is $\mathsf{H}(s_A)$ and $\mathsf{H}(s_B)$. These two rounds ensure that, if Alice and Bob agree on the same secret s, both of them would have made the query $\mathsf{H}(s)$.

As we have discussed before, $\mathsf{H}, \mathsf{Keygen}, \mathsf{F}, \mathsf{f}$ queries will not break the completeness of a party's view. Only R queries might temporarily break the completeness. If the completeness of a party's view is broken by an R query, the following $2n\kappa_{\max}$ H queries make sure that the completeness is fixed. Therefore, each party's view is complete at the end of each round.

5.2 Views

Denote V to be the *view* of all three parties at the end of the secret-key agreement protocol. It consists of all three parties' randomness r_A, r_B, r_E, all messages M^n, as well as all the queries and corresponding answers made so far by all three parties to Γ. For now, the oracle behavior in a view can be arbitrary, as long as the same query yields the same answer. We will define what is a valid view in the next section.

Given a view V, if we only consider a part of it, we get a partial view. Let r be a round number, denote $V_A^{r,j}$ as the partial view of Alice in V up to and include the jth query of round r. This does not include the message sent by Alice in round r. Denote V_A^r as the partial view of Alice in V up to the end of round r which includes the message sent by Alice at round r. Same for $V_B^{r,j}$ and V_B^r. The notation is slightly different for Eve, $V_E^{r,i}$ denotes the partial view of Eve in V up to and include segment i of round r. Here, we only need to know that Eve's algorithm in round r includes many segments. In each segment, Eve might make multiple queries. Especially, $V_E^{r,0}$ denotes the view of Eve in V right before Eve's algorithm starts in round r.

Given a (possibly partial) view V, let $P(V)$ be the set of oracle query-answer pairs that appear in V. We also denote $Q(V)$ as the elements of $Z \cup X \cup C \cup K \cup K'$ that appears in this view. Note that we have excluded the elements in Y, because they can only appear as an output of a query.

Conversely, given several partial views, if their common parts are the same, we can combine them together to form another (possibly partial) view. We slightly abuse the notion $\cup$ to denote the combination of partial views. For example, let U, V are two views and M^r are the same in these two views, then we denote $U_A^r \cup V_B^r \cup V_E^{r,i}$ to be the view obtained by replacing the partial view of Alice in V^r to U_A^r.

In this work, we focus on a probability distribution over views:

Definition 4. *Define $\mathcal{D}$ as the distribution of views generated by first randomly sampling all parties' randomness and oracle Γ defined in Section 4, then execute the protocol.*

6 Valid View and Indistinguishability Lemma $\mathcal{D} \approx \mathcal{D}'$

Recall that a "view" contains the randomness, messages, and query-answer pairs to Γ made by all parties. A view may include multiple master keys $k \in K$ and constrained keys $k' \in K'$ and many constrained relations $k[c] = k'$. A

Keygen query $\mathsf{Keygen}(k, c) = k'$ is equivalent to 1 such relation: $k[c] = k'$. An R query $\mathsf{R}(k, k') = c$ is equivalent to 1 such relation $k[c] = k'$. An R query $\mathsf{R}(k'_1, k'_2) = (k, c_1, c_2)$ is equivalent to 2 such relations: $k[c_1] = k'_1, k[c_2] = k'_2$. Since the behavior of the oracles in a view can be arbitrary, $k_1[c_1]$ and $k_2[c_2]$ might be the same for $(k_1, c_1) \neq (k_2, c_2)$. We refer to the scenario where $k_1[c_1]$ and $k_2[c_2]$ are identical despite (k_1, c_1) being different from (k_2, c_2) as a **collision**.

In this section, we will study a special type of view called *valid view* that only allows "mild" collisions. Then we introduce another distribution $\mathcal{D}'$ defined over valid views and show that $\mathcal{D}$ and $\mathcal{D}'$ are statistically close which is the key lemma in proving our main theorem.

6.1 Valid View

In this section, we will define a special type of view called "valid view". Formally, we make the following definition:

Definition 5. *A view V is a valid view if it satisfies the following conditions:*

1. *For two distinct master keys $k_1, k_2 \in K_\kappa$, if they collide at k' in V, i.e. $k_1[c_1] = k_2[c_2] = k'$, then there will be no collisions related to k_1, k_2.*
2. *There are no self-collisions. Here, a self-collision means that one master key generates the same constrained key for two different predicates, i.e. $k[c_1] = k[c_2]$ for $c_1 \neq c_2$.*

In a valid view, master keys and constrained keys can be divided into the following 3 types of groups:

1. **Isolated constrained key:** This group only contains 1 constrained key k' which does not appear to be any master key's constrained key in the view. Formally speaking, there exists no k, c such that $k' = k[c]$ in this view.
2. **Normal group:** this group contains one master key k and the constrained keys generated from k and k does not collide with any other master keys.
3. **Hybrid group:** this group can be viewed as two normal groups colliding at a single constrained key. Suppose k_0, k_1 are two different master keys, $\{c_0, c_1, \ldots, c_s\}$ and $\{c'_0, c'_1, \ldots, c'_t\}$ are two sets of predicates (the predicates in each set are different from each other). $k_0[c_0] = k_1[c'_0] = k'$ represents the colliding constrained key. The rest constrained keys $k_0[c_i], k_1[c'_j], i, j \geq 1$ are different from each other, and all different from k'.

To better understand the concept of valid view, we can think of the master keys and constrained keys in a view as nodes. A master key is represented by a filled node and a constrained key is represented by an unfilled node. A constrained relation $k[c] = k'$ is represented by an edge c connecting k and k'. The following example is a valid view that contains a collision.

Example 1. Suppose a view V contains the following query-answer pairs: $\mathsf{R}(k'_1, k'_3) = \perp$, $\mathsf{f}(k'_2, x) = y_1, \mathsf{R}(k'_4, k'_5) = (k_1, c_2, c_3), \mathsf{Keygen}(k_1, c_1) = k'_3, \mathsf{f}(k'_6, x) = y_2, \mathsf{R}(k'_6, k'_8) =$

$(k_2, c_4, c_6), \mathsf{R}(k_2, k'_7) = c_5, \mathsf{R}(k'_8, k'_9) = (k_3, c_7, c_8), \mathsf{f}(k'_9, x) = y_3$. Then, figure 3 is the graph illustration of the relations between the keys in V. Note that, V contains a collision: $k_2[c_6] = k_3[c_7] = k'_8$, this means that $k_2, k_3, k'_6, k'_7, k'_8, k'_9$ form a hybrid group.

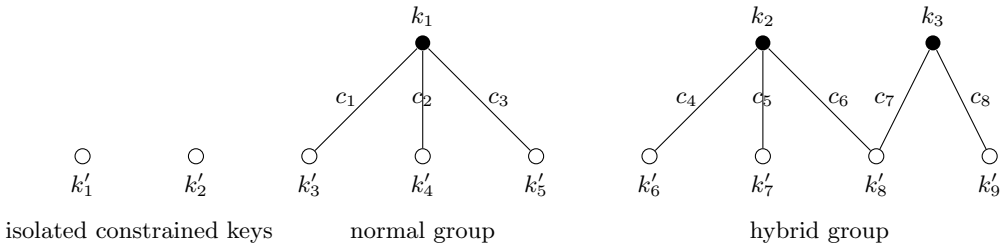


Fig. 3: Example of a valid view V which contains 2 isolated constrained keys, 1 normal group and 1 hybrid group.

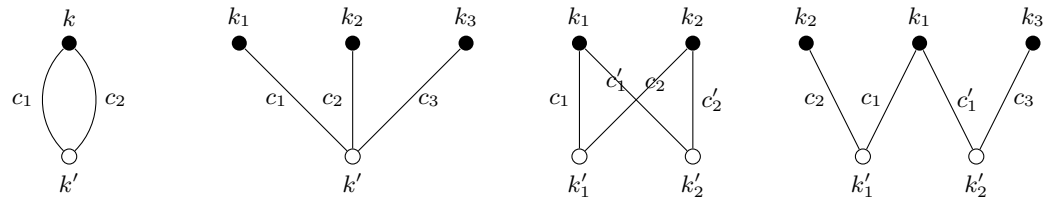
If a view is not valid, we can easily deduce that at least one of the following 4 complicated collisions must occur:

- Self-collision:
$$k[c_1] = k[c_2] = k'$$
- 3-to-1 collision: three master keys collide at one constrained key:
$$k_1[c_1] = k_2[c_2] = k_3[c_3] = k'$$
- 2-to-2 collision: two master keys collide at two different constrained keys:
$$k_1[c_1] = k_2[c_2] = k'_1, k_1[c'_1] = k_2[c'_2] = k'_2$$
- "W" collision: one master key simultaneously collides with two other master keys at two different constrained keys:
$$k_1[c_1] = k_2[c_2] = k'_1, k_1[c'_1] = k_3[c_3] = k'_2$$

Figure 4 is the graph illustration of these 4 types of complicated collisions.

We remark that valid views may contain collisions in the form of hybrid groups, this makes them different from the views in $\mathsf{sup}(\mathcal{D})$ which cannot contain any collisions because Keygen is sampled from injective functions. Thus, valid views can be seen as an extension over $\sup(\mathcal{D})$.

The readers may wonder why we allow a valid view to contain collisions. After all, collisions will never happen in a real execution because Keygen is sampled from all injective functions. The reason is that Eve's simulated view of Alice, combined with the view of Eve and Bob, might contain a collision. Looking ahead, we can see that the simulated view of Alice, combined with the view of Bob, forms a valid view if the simulation is a "miss".



self-collision 3-to-1 collision 2-to-2 collision "W" collision

Fig. 4: Complicated collisions that are not allowed in a valid view

6.2 Equivalent F or f Queries

Notice that in the definition of valid views, the correctness of PCPRF is not considered. Recall the correctness of PCPRF states that $\mathsf{F}(k,x) = \mathsf{f}(k[c],x)$ if $c(x) = 1$. The reason that we have not yet considered the correctness of PCPRF is that, in order to evaluate $c(x)$, we might need to make extra H queries that are not in V. Thus, the correctness of PCPRF must be considered under a fixed instantiation H of the oracle H.

Suppose V is a valid view. H is an instantiation of the oracle H that is consistent with the H queries made in V. Suppose $k[c] = k'$ in V. The correctness of PCPRF under V, H states that $\mathsf{F}(k,x) = \mathsf{f}(k',x)$ if $c^H(x) = 1$. We say that two F or f queries are equivalent under V and H if they can be deduced to have the same answer from the correctness of PCPRF under V, H.

Still take Example 1 as an illustration. Suppose H is an instantiation of the oracle H. Since no H queries are made in V, H is consistent with V. If $c_4^H(x) = c_6^H(x) = c_8^H(x) = c_9^H(x) = 1$, we can say that $\mathsf{f}(k'_6,x)$ and $\mathsf{f}(k'_9,x)$ are equivalent, because the correctness of PCPRF states that $\mathsf{f}(k'_6,x) = \mathsf{F}(k_2,x) = \mathsf{f}(k'_8,x) = \mathsf{F}(k_3,x) = \mathsf{f}(k'_9,x)$. However, if at least one of $c_4^H(x), c_6^H(x), c_8^H(x), c_9^H(x)$ is 0, $\mathsf{f}(k'_6,x)$ and $\mathsf{f}(k'_9,x)$ are not equivalent under V, H. This is clearer in the graph illustration Figure 3. Note that the edges on the shortest path from k'_6 to k'_9 are exactly c_4, c_6, c_8, c_9. Thus, $\mathsf{f}(k'_6,x)$ and $\mathsf{f}(k'_9,x)$ are equivalent under V and H if and only if $c(x) = 1$ for every c on this shortest path.

We remark that the answer of $\mathsf{f}(k'_6,x)$ and $\mathsf{f}(k'_9,x)$ in V, i.e. y_2, y_3 are irrelevant to whether $\mathsf{f}(k'_6,x)$ and $\mathsf{f}(k'_9,x)$ are equivalent under V, H.

6.3 Weights of Valid Views and the Indistinguishability Lemma

In this section, we define the notion of *weight* of valid views. With this notion, we may define a new distribution of views called $\mathcal{D}'$ whose support set contains all valid views, thus contains views with hybrid groups. Then, we will prove that $\Delta(\mathcal{D}, \mathcal{D}') \leq \mathsf{negl}(\lambda)$ (Lemma 3), which is essential to our black-box separation result.

First, even though the input of H can be any string, each party cannot make H queries with arbitrarily long input. Thus, we think of H as finite and denote

the size of its domain as M. For a fixed instantiation H of the oracle H and a valid view V, we define $w_H(V)$ as follows:

1. If the H queries made in V are not consistent with H, then $w_H(V) = 0$.
2. If two F or f queries made in V are equivalent under V, H, but their answers are different in V, then $w_H(V) = 0$.
3. Otherwise, define $w_H(V)$ as follows: each constrained relation $k[c] = k'$ in V contributes a factor of $|K'_\kappa|^{-1}$. Here $k \in K_\kappa, c \in C_\kappa, k' \in K'_\kappa$. Each different F_κ or f_κ query in V contributes a factor of $|Y_\kappa|^{-1}$. However, equivalent F or f queries under V and H are counted only once. $w_H(V)$ is the product of all contributions.

Still take Example 1 as an illustration. Suppose the queries made in V are all in Γ_κ. Suppose H is an instantiation of the oracle H such that $c_4^H(x) = c_6^H(x) = c_7^H(x) = c_8^H(x) = 1$. Now we want to compute $w_H(V)$:

1. Since no H queries are made in V, H is naturally consistent with V.
2. There are 3 F or f queries in V: $\mathsf{f}(k'_2, x), \mathsf{f}(k'_6, x), \mathsf{f}(k'_9, x)$. Since k'_2 is an isolated constrained key in V, $\mathsf{f}(k'_2, x)$ is nonequivalent to the other two queries. As we have shown in Section 6.2, $\mathsf{f}(k'_6, x), \mathsf{f}(k'_9, x)$ are equivalent under V, H. Thus, if $y_2 \neq y_3$, $w_H(V) = 0$. In the following, we assume $y_2 = y_3$.
3. There are 8 constrained relations in V, each of which contributes a factor of $|K'_\kappa|^{-1}$. Next, there are 3 F or f queries in V, and two of them are equivalent. Thus, these 3 queries contribute a factor of $|Y_\kappa|^{-2}$ in total. Therefore, $w_H(V) = |K'_\kappa|^{-8}|Y_\kappa|^{-2} = 2^{-2\kappa^3-8\kappa^2}$.

Finally, the weight of a valid view V is defined as

$$w(V) = \frac{1}{2^M} \sum_H w_H(V)$$

Here, the summation is taken over all instantiations of H.

Then, the distribution $\mathcal{D}'$ is defined as follows:

Definition 6. *$\mathcal{D}'$ is distributed over all valid views, the probability of a valid view V in $\mathcal{D}'$ is proportional to $w(V)$.*

We can, on some level, think of $w_H(V)$ as an approximated probability of the view V conditioned on H instantiated as H: for each pair of $k \in K_\kappa, c \in C_\kappa$, the constrained key $k[c]$ is sampled from K'_κ. Thus, for a specific k', the probability that $k[c] = k'$ is $|K'_\kappa|^{-1}$. Also, for each pair of $k \in K_\kappa, x \in X_\kappa$, the value of $\mathsf{F}(k, x)$ can be seen as sampled from Y_κ. Thus, for a specific y, the probability that $\mathsf{F}(k, x) = y$ is $|Y_\kappa|^{-1}$. Following this line of thought, we can think of $w(V)$ as an estimated probability of the view V, since $w(V)$ is the average value of $w_H(V)$ over all possible H. Therefore, it is plausible that $\mathcal{D}'$ and $\mathcal{D}$ should be close. Indeed, we can prove the following indistinguishability lemma:

Lemma 3. *For any secret-key agreement protocol that each party never queries Γ_κ for $\kappa < \log_2 \lambda$ as we required in Section 5.1, it holds that $\Delta(\mathcal{D}, \mathcal{D}') \leq \mathsf{negl}(\lambda)$.*

We defer the proof of Lemma 3 to Appendix B.

6.4 "Good" Views

In the previous section, we defined the weight of a view $w(V)$. However, the definition is complicated and not favorable for our purpose. Fortunately, we will mainly deal with a special type of view whose weight is much easier to compute and comprehend. We call this type of views as "**good**" views.

Recall the process of computing $w_H(V)$: if H is not consistent with the H queries in V, then $w_H(V) = 0$; if two equivalent queries do not have the same answer in V, then $w_H(V) = 0$; otherwise, each constrained relation $k[c] = k'$ contributes a factor of $|K'_\kappa|^{-1}$, each nonequivalent F_κ or f_κ query contributes a factor of $|Y_\kappa|^{-1}$. Then, we can see that $w_H(V)$ depend on H in an indirect way: $w_H(V)$ only depend on the equivalency relations of the F and f queries made in V, and these relations depend on H. Therefore, we make the following definition:

Definition 7. *We say a valid view V is "**good**" if the H queries made in V are enough to determine the equivalency relations of the F and f queries made in V.*

Still take Example 1 as an illustration. In the view V, only 3 F or f queries are made: $\mathsf{f}(k'_2, x), \mathsf{f}(k'_6, x)$ and $\mathsf{f}(k'_9, x)$. k'_2 is not in the same group as k'_6, k'_9, so $\mathsf{f}(k'_2, x)$ is not equivalent to the other two. Thus, only the equivalency relation between $\mathsf{f}(k'_6, x)$ and $\mathsf{f}(k'_9, x)$ is unclear. As we have analyzed in Section 6.2, $\mathsf{f}(k'_6, x)$ and $\mathsf{f}(k'_9, x)$ are equivalent if and only if $c_4(x) = c_6(x) = c_7(x) = c_8(x) = 1$. Therefore, in order to make V a good view, V must additionally contain enough H queries to ensure that $c_4(x) = c_6(x) = c_7(x) = c_8(x) = 1$, or at least one of $c_4(x) = 0, c_6(x) = 0, c_7(x) = 0, c_8(x) = 0$.

For a good view V and any instantiation H of the oracle H that is consistent with V, the value of $w_H(V)$ does not depend on those parts of H that are not queried in V. This means that the value of $w_H(V)$ is the same for any H consistent with V. Therefore, we use the notation $w_*(V)$ to indicate that the specific instantiation H in $w_H(V)$ can be arbitrary, as long as it is consistent with V.

For a good view V, we will show that the weight $w(V)$ can be easily computed. Suppose there are t different H queries in V. The number of instantiations H that is consistent with V is 2^{M-t}. (Recall that M is H's domain size.) Then, we have

$$w(V) = \frac{1}{2^M} \sum_H w_H(V) = \frac{1}{2^M} \cdot 2^{M-t} w_*(V) = 2^{-t} w_*(V)$$

This means we can use the algorithm for computing $w_*(V)$ to compute $w(V)$, with the difference that each different H query in V also contributes a factor of $1/2$.

From Definition 3, for a complete view V and any $c \in C_\kappa$ and $x \in X_\kappa$ in V, the H queries made in V are enough to evaluate $c(x)$. This means the equivalency relations of the F and f queries made in V can be determined solely from the H queries made in V. Namely, V is a good view.

Suppose V is a collision free view generated by one party under restriction 2, 4 of the normal form. Then V might not be complete, because R query might

temporarily break the completeness. However, we can prove that V is good. This is straight forward: if V is not good, then there are two F or f queries in V whose equivalency relations cannot be determined by V. Denote them as Query_0 and Query_1. We assume Query_1 is made after Query_0. Then, right before making Query_1, the equivalency relation between Query_0 and Query_1 should be clear because of restriction 2,4 of the normal form. This contradicts to the fact that V is not good.

Suppose V_1 and V_2 are two complete views generated from two parties with these restriction 2,4 of the normal form. We can prove that the combined view $V_1 \cup V_2$ is also a good view if $V_1 \cup V_2$ is collision-free.

Lemma 4. *Suppose V_1 and V_2 are complete views of two parties whose algorithms are both under restriction 2, 4 of the normal form. If $V_1 \cup V_2$ is collision free, then $V_1 \cup V_2$ is good.*

Proof. We use proof by contradiction. Denote $V = V_1 \cup V_2$. If V is not a good view, then there are only two possibilities:

1. $k \in K, k' \in K'$ both appeared in V and it can be observed in V that $k' = k[c]$ for some $c \in C$. Both $\mathsf{F}(k, x)$ and $\mathsf{f}(k', x)$ are queried in V. But the H queries made in V are not enough to evaluate $c(x)$.
2. $k \in K, k'_1, k'_2 \in K'$ both appeared in V and it can be observed in V that $k'_1 = k[c_1], k'_2 = k[c_2]$ for some c_1, c_2. Both $\mathsf{f}(k'_1, x)$ and $\mathsf{f}(k'_2, x)$ are queried in V. But the H queries made in V are not enough to evaluate $c_1(x)$.

Here, we only present the contradiction in the first case. The second case follows the same logic. Since it can be observed that $k' = k[c]$, then c must appear in V_1 or V_2. Without loss of generality, suppose c appears in V_1. If x also appears in V_1, enough H queries must be made in V_1 to evaluate $c(x)$ because V_1 is complete, a contradiction. Thus, both $\mathsf{F}(k, x)$ and $\mathsf{f}(k', x)$ are queried in V_2. One of these two queries must be queried after the other. Because of restriction 4, before querying the later, Bob must first ensure that there are no hidden constrained relations between k and k', which means that Bob will find out the relation $k' = k[c]$. Because V_2 is also complete, V_2 contains enough H queries to evaluate $c(x)$, a contradiction. □

We remark that the condition "$V_1 \cup V_2$ is collision free" is necessary. Without this condition, lemma 4 is wrong. Consider the following case. V_1 consists of the following 2 queries: $\mathsf{Keygen}(k_1, c_1) = k', \mathsf{Keygen}(k_2, c_2) = k'$. V_2 consists of the following 2 queries: $\mathsf{F}(k_1, x) = y_1, \mathsf{F}(k_2, x) = y_2$. We can see that restriction 2 and 4 are strictly followed in V_1 and V_2. Also, V_1 and V_2 are complete views. However, $V_1 \cup V_2$ is not a good view because the equivalency relation between $\mathsf{F}(k_1, x)$ and $\mathsf{F}(k_2, x)$ is unclear.

7 Breaking Secret-Key Agreement Protocol Relative to Oracle Γ

In Theorem 2, we have shown that relative to almost all oracles Γ (together with a PSPACE-complete oracle), 1-key PCPRF scheme exists. Our next goal is to

construct an adversary Eve who can break any secret-key agreement protocol in the normal form

7.1 Eve's Goal

In the normal form of KA protocol (see Section 5.1) Alice and Bob queries $\mathsf{H}(s_A)$ and $\mathsf{H}(s_B)$ respectively in the last round, where s_A and s_B are Alice and Bob's output. With probability $\alpha(\lambda)$, $s_A = s_B = s$, then both Alice and Bob would have made the same query $\mathsf{H}(s)$ at the end of the protocol. This means, $s \in Q(V_A) \cap Q(V_B)$, here V_A and V_B denote the view of Alice and Bob at the end of the protocol. Thus, Eve's mission can be reduced to find all the intersection elements $Q(V_A) \cap Q(V_B)$. If Eve accomplishes to include all the intersection elements with probability $1-\alpha/2$, then she finds all the intersection elements with probability at least $1/2$ conditioned on $s_A = s_B$. Thus, because Eve's view only contains polynomially many elements, Eve can simply choose one element randomly from her view as her guess. The guess would be s with at least $1/\mathsf{poly}(\lambda)$ probability. This means the protocol is broken.

7.2 Eve's Algorithm

Suppose r is even, then Bob speaks in round r and Alice speaks in round $r-1$. Eve's algorithm in round r can be seen as running simultaneously with Bob's algorithm in round r. Eve's algorithm in round r consists of many repeated segments. Before segment i, Eve's partial view is $V_E^{r,i-1}$ which is complete. Then, segment i is divided into the following two phases:

- **Simulation Phase**: sample a view of Alice up to the end of round $r-1$: $\mathsf{Sim}_A \leftarrow \mathsf{Sim}(M^r, V_E^{r,i-1})$. Here, the distribution $\mathsf{Sim}(M^r, V_E^{r,i-1})$ is defined as follows: $\mathsf{SIM}(M^{r-1}, V_E^{r,i-1})$ is distributed over all possible partial views of Alice up to the end of round $r-1$ (denoted as Sim_A) that are consistent with M^{r-1}. Besides, $\mathsf{Sim}_A \cup V_E^{r,i-1}$ must form a collision-free valid view. By lemma 4, $\mathsf{Sim}_A \cup V_E^{r,i-1}$ is a good view. The probability of Sim_A in $\mathsf{SIM}(M^r, V_E^{r,i-1})$ is proportional to $w(\mathsf{Sim}_A \cup V_E^{r,i-1})$.
- **Update Phase**: Eve includes all the elements in $Q(\mathsf{Sim}_A)$ in her view. Here, Sim_A is the sampled view of Alice in the last simulation phase. Then, Eve will make her view full, meaning that she will try to acquire informations related to $Q(V_E)$ as much as possible. Specifically, for each pair of $k \in K, c \in C$ in her view, she will ask $\mathsf{Keygen}(k, c)$, the answer is automatically included in her view; for each pair of $x \in X, k \in K$ (resp. $k' \in K'$), she will ask $\mathsf{F}(k, x)$ (resp. $\mathsf{f}(k', x)$); Eve always make best efforts to recover the relationships of the master-keys and constrained keys in her view, this can be done by the recovery oracle R. Additionally, Eve will always maintain the completeness of her view. After the update, Eve's view is $V_E^{r,i}$. Naturally, $V_E^{r,i}$ is also complete.

Then, round r proceeds as follows (including Bob and Eve's algorithm):

1. Eve runs m segments described above. (Here $m = m(\lambda)$ is polynomial over λ. Its value is determined later.)
2. Bob makes the first query in round r.
3. We assume that there is a fourth party Charlie. She has the view of all three parties. Charlie is designated for one purpose: if Bob's first query in the last step is an R query and the answer is not $\perp$, then the answer must contain 1 or 2 predicates, say c_1 (and c_2). If c_1 has never appeared in Bob and Eve's view before, Charlie will send c_1 to Eve. Same for c_2. Otherwise Charlie simply sends $\perp$ to Eve.
4. If Charlie's message is not $\perp$ in the last step, Eve will include the received predicates in her view and make her view full, same as the update phase of a segment.
5. Eve runs m segments.
6. Bob makes the rest H queries in round r.
7. Bob sends the message M^r to Alice.

Here, Bob's queries are divided into 2 parts: the first query and the rest $2n\kappa_{\max}$ H queries. Before each part, Eve runs m segments. Between these two parts, we introduces a fourth party Charlie that might send one or two predicates to Eve. If Eve does not receive $\perp$, she will run another update phase to include the received predicates into her view. Indeed, Eve might receive private informations from Charlie, which renders Eve's attack invalid. Therefore, we make the following rule: as long as Charlie sends a message other than $\perp$, Eve is considered failed. In other words, we say Eve fails if (1), Eve does not include all the intersection elements in her view; (2), Charlie sends a message other than $\perp$. We will show that the probability that Eve fails is small. This means that the probability that Eve fails is also small even without Charlie.

7.3 Distribution $\mathcal{D}^{r,i,j}$ and $\mathcal{D}'^{r,i,j}$

The definition of $\mathcal{D}^{r,i,j}, \mathcal{D}'^{r,i,j}$ is the same as $\mathcal{D}, \mathcal{D}'$, except that we make the algorithms of Alice and Bob stop after the j th query of round r and Eve's algorithm stops after segment i of round r. (Especially, if $i = m$, the view sampled from $\mathcal{D}^{r,m,j}$ or $\mathcal{D}'^{r,m,j}$ should also contain the part that Charlie sends messages to Eve and Eve makes updates.) Proof of Lemma 3 also shows $\Delta(\mathcal{D}^{r,i,j}, \mathcal{D}'^{r,i,j}) \leq \mathsf{negl}(\lambda)$.

Recall that the sampling process of $\mathcal{D}$ is as follows: first sample the randomness of three parties r_A, r_B, r_E (Charlie is deterministic) and the whole oracle Γ, then run the protocol. The only difference between $\mathcal{D}$ and $\mathcal{D}^{r,i,j}$ is that the protocol of $\mathcal{D}^{r,i,j}$ stops earlier than $\mathcal{D}$. Thus, we can sample $V^{r,i,j} \leftarrow \mathcal{D}^{r,i,j}$ in the following way: sample $V \leftarrow \mathcal{D}$ first, then output the partial view $V^{r,i,j}$. [2]

[2] However, the same does not necessarily hold for $\mathcal{D}'$ and $\mathcal{D}'^{r,i,j}$ because $\mathcal{D}'$ and $\mathcal{D}'^{r,i,j}$ are defined in a completely different way than $\mathcal{D}$ and $\mathcal{D}'^{r,i,j}$.

7.4 Main Result

Under certain parameter regime, we prove that Eve can break the secret-key agreement protocol with high probability. Let $S(\lambda) = 1/\mathsf{poly}(\lambda)$, we have

Theorem 3. *At the end of the protocol, Eve finds all the intersection elements of Alice and Bob's view while Charlie always sends $\perp$ with high probability. Formally, for some sufficiently large polynomial m depending on $S(\lambda)$ (determined later),*

$$\Pr_{V \leftarrow \mathcal{D}}[Q(V_A^n) \cap Q(V_B^n) \subseteq Q(V_E^{n,2m}) \wedge \textit{Charlie always sends } \perp] > 1 - S(\lambda)$$

for sufficiently large λ.

Thus, as long as $S(\lambda) < \alpha(\lambda)/2$, Eve can break the KA protocol as discussed in Section 7.1.

7.5 Proof of Theorem 3

In the rest of this section, we will give a proof sketch for our main result Theorem 3.

Recall that Eve's goal is to include all the intersection elements in her view, while Charlie always sends $\perp$. We instead prove a stronger result: before an element q becomes an intersection element, Eve will anticipate q with high probability. We say Eve fails if

1. An element q becomes an intersection element as an input to a query, but Eve's view does not include it.
2. An element q becomes an intersection element as an output of a query, but Eve's view does not include it.
3. Charlie sends a message other than $\perp$.

If Eve never fails through the whole execution of the protocol, all intersection elements must be included in Eve's view and Charlie always sends $\perp$. This means that Eve accomplishes her goal. If Eve fails, there must be a first failure. We first show the probability that Eve's first failure is type 2 or 3 is negligible.

Lemma 5. *Suppose V is sampled from $\mathcal{D}$. Then the probability that Eve fails in V and the first failure is type 2 or 3 is negligible.*

Proof. For a view V sampled form $\mathcal{D}$, define the event spoof as the following: there exists a constrained key k' appears in view V. The first time k' appears is **not** as the output of a Keygen query. However, k' is later proved to be a constrained key $k[c]$ for some k, c in the view V. Clearly, spoof happens with negligible probability for $V \leftarrow \mathcal{D}$. This is because the proportion of constrained keys in K'_κ is $\frac{|K_\kappa| \cdot |C_\kappa|}{|K'_\kappa|} = 2^{2\kappa - \kappa^2}$. For $\kappa \geq \log \lambda$, this proportion is negligible.

With this observation, this lemma can be easily proved. We show that, if Eve fails and the first failure is type 2 or 3, then spoof occurs. This implies that the probability of Eve failing and the first failure is type 2 or 3 is negligible.

First consider the case that the first failure is type 2. Then there are 3 different scenarios: (1), the query is $\mathsf{Keygen}(k, c) = k'$, then $q = k'$. (2), the query is $\mathsf{R}(k, k') = c$, then $q = c$. (3), the query is $\mathsf{R}(k'_1, k'_2) = (k, c_1, c_2)$, then q is k, c_1 or c_2. Here we only analyze the first scenario, the other two follows the same logic. Without loss of generality, suppose the query is made by Bob. Since $q = k'$ becomes an intersection element at this point, k' never appeared in Bob's view before, and k' appeared in Alice's view. This means, k' first appeared in Alice's view. If spoof does not occur, Alice must have generated k' with Keygen queries. This means, k, c has appeared in Alice's view. Therefore, k, c are both intersection elements. Since k' is Eve's first failure, k, c must have already appeared in Eve's view. This means Eve has made the query $\mathsf{Keygen}(k, c)$, and the answer would be k', a contradiction. Thus, spoof must occur.

Type 3 failure follows the same logic as well. If Charlie sends a message other than $\perp$, Bob's first query in this round would be: (1), $\mathsf{R}(k, k') = c$ and Charlie sends c to Eve; (2), $\mathsf{R}(k'_1, k'_2) = (k, c_1, c_2)$ and Charlie sends at least one of c_1, c_2 to Eve. Here, we only analyze the first case, the second case follows the same logic. Since Charlie sends c to Eve, c must have not appeared in Bob or Eve's view before this query. This means, k' is not generated by Eve or Bob with Keygen queries. Thus, if spoof does not occur, k' must be generated by Alice with a Keygen query. This means k, c, k' all appeared in Alice's view. Since Charlie sending c is Eve's first failure, k, k' must have appeared in Eve's view. This means Eve must have queried $\mathsf{R}(k, k')$ and the answer would be c, a contradiction. Thus, spoof must occur. □

With this lemma, we only need to concern type 1 failures. Recall that Alice and Bob can make at most $1 + 2n\kappa_{\max}$ queries in each round, and there are n rounds in total, Alice and Bob can make at most $n(1+2n\kappa_{\max})$ queries in total. Denote $p(\lambda) = S(\lambda)/n(\lambda)(1 + 2n(\lambda)\kappa_{\max}(\lambda))$. If we can bound the probability that the input of the jth query of round r is Eve's first failure by p, we can bound the probability that Eve fails. Here, we consider the first query and the rest H queries separately. Formally, we prove the following 2 lemmas:

Lemma 6. *For some sufficiently large polynomial $m = m(\lambda)$ depending on $p(\lambda)$, for each round number $r \in [1, n]$ (without loss of generality assume Bob speaks in round r), the probability that either*

1. *$Q(V_A^{r-1}) \cap Q(V_B^{r-1}) \not\subseteq Q(V_E^{r,0})$. (At the end of round $r-1$, Eve failed to include all intersection elements. This is stronger than Eve failing before round r).*
2. *$q_{r,1} \notin Q(V_A^{r-1})$. (For $q_{r,1}$ to be an intersection element, it must have appeared in Alice's view before round r.)*
3. *$q_{r,1} \in Q_E^{r,m}$. (The definition of Eve finding $q_{r,1}$ at the end of the first m segments.)*

is greater than $1 - p(\lambda)/2$ for sufficiently large λ.

Lemma 7. *For some sufficiently large polynomial $m = m(\lambda)$ depending on $p(\lambda)$, for each round number $r \in [1, n]$ (without loss of generality assume Bob speaks in round r) and query number $j \in [2, 1 + 2n\kappa_{\max}]$, the probability that either*

1. $Q(V_A^{r-1}) \cap Q(V_B^{r,j-1}) \not\subseteq Q(V_E^{r,m})$. *(Right before the jth query of round r, Eve failed to include all intersection elements with the first m segments. This is stronger than Eve failing before the jth query of round r).*
2. $q_{r,j} \notin Q(V_A^{r-1})$. *(For $q_{r,j}$ to be an intersection element, it must have appeared in Alice's view before round r.)*
3. $q_{r,j} \in Q(V_E^{r,2m})$. *(The definition of Eve finding $q_{r,j}$ at the end of round r.)*

is greater than $1 - p(\lambda)/2$ for sufficiently large λ.

Note that, if $\mathsf{Condition1} \vee \mathsf{Condition2} \vee \mathsf{Condition3}$ has probability greater than $1 - p/2$, then $\neg\mathsf{Condition1} \wedge \neg\mathsf{Condition2} \wedge \neg\mathsf{Condition3}$ has probability smaller than $p/2$, and this is weaker than Eve first failing at this query.

We defer the detailed proofs of Lemma 6 and Lemma 7 to Appendix C.

Now Theorem 3 can be proved by Lemma 6 and Lemma 7:

Proof. Combining Lemma 6 and Lemma 7, for each round number r and query number j, the probability that Eve's first failure is the input of the jth query of round r and the failure is type 1 is bounded by $p/2$. Apply a union bound, the probability that Eve fails and the first failure is type 1 is bounded by $(p/2) \cdot n(1 + 2n\kappa_{\max}) = S/2$. Combined with Lemma 5, the probability that Eve fails is bounded by $S/2 + \mathsf{negl}(\lambda) < S$ for sufficiently large λ. This is stronger than Theorem 3. □

References

1. Asharov, G., Segev, G.: Limits on the power of indistinguishability obfuscation and functional encryption. SIAM Journal on Computing **45**(6), 2117–2176 (2016). https://doi.org/10.1137/15M1034064, https://doi.org/10.1137/15M1034064
2. Attrapadung, N., Matsuda, T., Nishimaki, R., Yamada, S., Yamakawa, T.: Constrained prfs for NC^1 in traditional groups. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018. pp. 543–574. Springer International Publishing, Cham (2018)
3. Attrapadung, N., Matsuda, T., Nishimaki, R., Yamada, S., Yamakawa, T.: Adaptively single-key secure constrained prfs for NC^1. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 11443, pp. 223–253. Springer (2019)
4. Barak, B., Mahmoody-Ghidary, M.: Merkle puzzles are optimal — an $\mathrm{O}(n^2)$-query attack on any key exchange from a random oracle. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009. pp. 374–390. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
5. Boneh, D., Lewi, K., Wu, D.J.: Constraining pseudorandom functions privately. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 10175, pp. 494–524. Springer (2017)
6. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 8270, pp. 280–300. Springer (2013)
7. Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology – CRYPTO 2014. pp. 480–499. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)

8. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: Krawczyk, H. (ed.) Public-Key Cryptography – PKC 2014. pp. 501–519. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
9. Brakerski, Z., Katz, J., Segev, G., Yerukhimovich, A.: Limits on the power of zero-knowledge proofs in cryptographic constructions. In: Ishai, Y. (ed.) Theory of Cryptography. pp. 559–578. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
10. Brakerski, Z., Tsabary, R., Vaikuntanathan, V., Wee, H.: Private constrained prfs (and more) from lwe. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography. pp. 264–302. Springer International Publishing, Cham (2017)
11. Brakerski, Z., Vaikuntanathan, V.: Constrained key-homomorphic prfs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In: TCC (2). Lecture Notes in Computer Science, vol. 9015, pp. 1–30. Springer (2015)
12. Canetti, R., Chen, Y.: Constraint-Hiding Constrained PRFs for NC^1 from LWE. In: Coron, J.S., Nielsen, J.B. (eds.) Advances in Cryptology – EUROCRYPT 2017, vol. 10210, pp. 446–476. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_16, http://link.springer.com/10.1007/978-3-319-56620-7_16, series Title: Lecture Notes in Computer Science
13. Davidson, A., Katsumata, S., Nishimaki, R., Yamada, S., Yamakawa, T.: Adaptively secure constrained pseudorandom functions in the standard model. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 559–589. Springer International Publishing, Cham (2020)
14. Gennaro, R., Trevisan, L.: Lower bounds on the efficiency of generic cryptographic constructions. In: Proceedings 41st Annual Symposium on Foundations of Computer Science. pp. 305–313 (2000). https://doi.org/10.1109/SFCS.2000.892119
15. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: Proceedings 41st Annual Symposium on Foundations of Computer Science. pp. 325–335 (2000). https://doi.org/10.1109/SFCS.2000.892121
16. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
17. Hofheinz, D., Kamath, A., Koppula, V., Waters, B.: Adaptively secure constrained pseudorandom functions. In: Goldberg, I., Moore, T. (eds.) Financial Cryptography and Data Security. pp. 357–376. Springer International Publishing, Cham (2019)
18. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing. p. 44–61. STOC '89, Association for Computing Machinery, New York, NY, USA (1989). https://doi.org/10.1145/73007.73012, https://doi.org/10.1145/73007.73012
19. Kiayias, A., Papadopoulos, S., Triandopoulos, N., Zacharias, T.: Delegatable pseudorandom functions and applications. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. p. 669–684. CCS '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2508859.2516668, https://doi.org/10.1145/2508859.2516668
20. Kretschmer, W., Qian, L., Sinha, M., Tal, A.: Quantum Cryptography in Algorithmica. In: Proceedings of the 55th Annual ACM Symposium on Theory of Computing. pp. 1589–1602. ACM, Orlando FL USA (Jun 2023). https://doi.org/10.1145/3564246.3585225, https://dl.acm.org/doi/10.1145/3564246.3585225
21. Lazzaretti, A., Papamanthou, C.: Near-optimal private information retrieval with preprocessing. In: Rothblum, G., Wee, H. (eds.) Theory of Cryptography. pp. 406–435. Springer Nature Switzerland, Cham (2023)

22. Rudich, S.: The use of interaction in public cryptosystems (extended abstract). In: 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. 242–251. Springer (1991). https://doi.org/10.1007/3-540-46766-1_19
23. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing. p. 475–484. STOC '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2591796.2591825, https://doi.org/10.1145/2591796.2591825
24. Shi, E., Aqeel, W., Chandrasekaran, B., Maggs, B.: Puncturable pseudorandom sets and private information retrieval with near-optimal online bandwidth and time. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021. pp. 641–669. Springer International Publishing, Cham (2021)
25. Simon, D.R.: Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In: Nyberg, K. (ed.) Advances in Cryptology — EUROCRYPT'98. pp. 334–345. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)

Supplementary Materials

A Deferred Proof of Theorem 2

To prove security, we construct a universal simulator S^{Γ}: S^{Γ} first randomly samples a $\mathsf{MSK} \xleftarrow{R} K_\kappa$ and $c_0 \xleftarrow{R} C_\kappa$. Let $\mathsf{CK} = \mathsf{Keygen}(\mathsf{MSK}, c_0)$. For an evaluation query x, S^{Γ} returns $\mathsf{f}(\mathsf{CK}, x)$; for a constrain query c, S^{Γ} returns CK.

First, we prove that for any oracle-aided P.P.T.M. adversary $\mathbf{Adv}^{\Gamma}$, a weaker variant of simulation indistinguishability holds.

Lemma 8. *Under Section 4.2, for any oracle-aided P.P.T.M. adversary* $\mathbf{Adv}^{\Gamma}$ *and* $\kappa \in \mathbb{N}^*$*, the following inequality always holds:*

$$\left| \Pr_{\Gamma, \mathsf{MSK}, r_{\mathbf{Adv}}}[\mathbf{REAL}(1^\kappa, \mathbf{Adv}^{\Gamma}, \mathsf{MSK}) = 1] - \Pr_{\Gamma, r_{\mathbf{Adv}}, r_S}[\mathbf{IDEAL}(1^\kappa, \mathbf{Adv}^{\Gamma}, S^{\Gamma}) = 1] \right| < \mathsf{negl}(\kappa).$$

Proof. Note that MSK is uniformly sampled from K_κ in both the **REAL** and **IDEAL** experiments. Denote $G(\mathsf{MSK}) = \{\mathsf{MK}, \mathsf{MK}[c] | c \in C_\kappa\}$. Our first observation is that other than CK which is the response to the constrain query, the elements in $G(\mathsf{MSK})$ will not appear in the view of $\mathbf{Adv}^{\Gamma}$ with probability $1 - \mathsf{negl}(\kappa)$ in both experiments. This is because $G(\mathsf{MSK})$ takes a negligible portion in both K_κ and K'_κ, and the oracle queries, including PCPRF oracle queries, constrain oracle queries and evaluation oracle queries, do not reveal any information about the elements in $G(\mathsf{MSK})$ except CK. Therefore, it is almost impossible to find an element in $G(\mathsf{MSK})$ other than CK.

Then, as long as the elements in $G(\mathsf{MSK}) - \{\mathsf{CK}\}$ do not appear in the view of $\mathbf{Adv}^{\Gamma}$, we argue that **REAL** and **IDEAL** are essentially the same experiment

from the perspective of $\mathbf{Adv}^{\Gamma}$. Consider each $\mathbf{Adv}^{\Gamma}$'s interaction with the oracle, one of the following cases occur:

1. $\mathbf{Adv}^{\Gamma}$ makes an PCPRF oracle query such that CK is not one of the input: in this case, the answer is the same in **REAL** and **IDEAL**.
2. $\mathbf{Adv}^{\Gamma}$ makes a constrain oracle query c: In **REAL**, the answer is $\mathsf{MSK}[c]$; in **IDEAL**, the answer is $\mathsf{MSK}[c_0]$ where c_0 is sampled by the simulator in advance. However, from the perspective of $\mathbf{Adv}^{\Gamma}$, since the elements in $G(\mathsf{MSK})$ do not appear in the view of $\mathbf{Adv}^{\Gamma}$, the response of this constrain oracle query is uniformly distributed over all the elements in K'_{κ} that has not appeared in the view of $\mathbf{Adv}^{\Gamma}$ in both **REAL** and **IDEAL**.
3. $\mathbf{Adv}^{\Gamma}$ makes an R query with CK being one of the input: since CK is the only element in $G(\mathsf{MSK})$ that appears in the view of $\mathbf{Adv}^{\Gamma}$, the answer is $\perp$ in both experiments.
4. $\mathbf{Adv}^{\Gamma}$ makes a query $\mathsf{f}(\mathsf{CK}, x)$ for some x: if $\mathbf{Adv}^{\Gamma}$ has made an evaluation oracle query x and $c(x) = 1$, then the answer of $\mathsf{f}(\mathsf{CK}, x)$ is the same as the response of the evaluation oracle query in both experiments; otherwise, $\mathsf{f}(\mathsf{CK}, x)$ is uniformly sampled from Y_{κ} in both experiments.
5. $\mathbf{Adv}^{\Gamma}$ makes a evaluation oracle query x: if $\mathbf{Adv}^{\Gamma}$ has made the query $\mathsf{f}(\mathsf{CK}, x)$ and $c(x) = 1$, then the response of this evaluation query is the same as the answer of $\mathsf{f}(\mathsf{CK}, x)$ in both experiments; otherwise, the response of this evaluation oracle query is uniformly sampled from Y_{κ} in both experiments.

Therefore, as long as the elements in $G(\mathsf{MSK}) - \{\mathsf{CK}\}$ do not appear in the view of $\mathbf{Adv}^{\Gamma}$, **REAL** and **IDEAL** are essentially the same experiments from the perspective of $\mathbf{Adv}^{\Gamma}$. Therefore, these two experiments are indistinguishable to $\mathbf{Adv}^{\Gamma}$. □

Note that in the previous inequality, we take expectation over Γ inside the absolute value, but we aim to prove the existence of PCPRF under a **certain oracle** Γ. Fortunately, inspired by [20, Corollary 32], we can upgrade the above weaker variant of simulation indistinguishability to a standard version using a generic method:

Lemma 9. *Under Section 4.2, for any oracle-aided P.P.T.M. adversary* $\mathbf{Adv}^{\Gamma}$ *and* $\kappa \in \mathbb{N}^*$*, the following inequality always holds:*

$$\mathbb{E}_{\Gamma}\left[\left|\Pr_{\mathsf{MSK},r_{\mathbf{Adv}}}[\mathbf{REAL}(1^{\kappa},\mathbf{Adv}^{\Gamma},\mathsf{MSK})=1]-\Pr_{r_{\mathbf{Adv}},r_S}[\mathbf{IDEAL}(1^{\kappa},\mathbf{Adv}^{\Gamma},S^{\Gamma})=1]\right|\right]$$
$$<\mathsf{negl}(\kappa).$$

Proof. Fix an adversary $\mathbf{Adv}^{\Gamma}$ and $\kappa \in \mathbb{N}^*$, Without loss of generality, we assume the output of $\mathbf{Adv}$ is always $\{\pm 1\}$. For simplicity, we define

$$a_{\mathbf{Adv}}(\Gamma) = \Pr_{\mathsf{MSK},r_{\mathbf{Adv}}}[\mathbf{REAL}(1^{\kappa},\mathbf{Adv}^{\Gamma},\mathsf{MSK})=1]$$

and

$$b_{\mathbf{Adv}}(\Gamma) = \Pr_{r_{\mathbf{Adv}}, r_S}[\mathbf{IDEAL}(1^\kappa, \mathbf{Adv}^\Gamma, S^\Gamma) = 1]$$

to be two random variables, then we can rewrite Lemma 8 as

$$|\mathbb{E}_\Gamma[a_{\mathbf{Adv}}(\Gamma) - b_{\mathbf{Adv}}(\Gamma)]| < \mathsf{negl}(\kappa) \tag{2}$$

for any $\mathbf{Adv}$.

Now we construct another oracle-aided P.P.T.M. adversary $\mathcal{A}^\Gamma(1^\kappa)$ which can "predict" the output of $\mathbf{Adv}$ in the following way:

1. Randomly chooses $c \in \{\pm 1\}$.
2. If $c = 1$, picks a random $\mathsf{MSK} \xleftarrow{R} K_\kappa$ and sets $d \leftarrow \mathbf{REAL}(1^\kappa, \mathbf{Adv}^\Gamma, \mathsf{MSK})$; if $c = -1$, sets $d \leftarrow \mathbf{IDEAL}(\mathbf{Adv}^\Gamma, S^\Gamma)$.
3. Simulates $\mathbf{Adv}$ but changes the output to be $c \cdot d \cdot b$ where b is the output of $\mathbf{Adv}$.

It is not hard to verify that the value of $c \cdot d$ is independent of which experiment $\mathcal{A}$ is actually in, and we have $\Pr[c \cdot d = 1] = \frac{1 + a_{\mathbf{Adv}}(\Gamma) - b_{\mathbf{Adv}}(\Gamma)}{2}$, $\Pr[c \cdot d = -1] = \frac{1 + b_{\mathbf{Adv}}(\Gamma) - a_{\mathbf{Adv}}(\Gamma)}{2}$.

Thus,

$$\begin{aligned}
\mathbb{E}_\Gamma[a_{\mathcal{A}}(\Gamma) - b_{\mathcal{A}}(\Gamma)] =& \Pr[c \cdot d = 1]\mathbb{E}_\Gamma[a_{\mathbf{Adv}}(\Gamma) - b_{\mathbf{Adv}}(\Gamma)] \\
&+ \Pr[c \cdot d = -1]\mathbb{E}_\Gamma[b_{\mathbf{Adv}}(\Gamma) - a_{\mathbf{Adv}}(\Gamma)] \\
=& \mathbb{E}_\Gamma\left[\frac{1 + a_{\mathbf{Adv}}(\Gamma) - b_{\mathbf{Adv}}(\Gamma)}{2} \cdot (a_{\mathbf{Adv}}(\Gamma) - b_{\mathbf{Adv}}(\Gamma))\right. \\
&\left. + \frac{1 + b_{\mathbf{Adv}}(\Gamma) - a_{\mathbf{Adv}}(\Gamma)}{2} \cdot (b_{\mathbf{Adv}}(\Gamma) - a_{\mathbf{Adv}}(\Gamma))\right] \\
=& \mathbb{E}_\Gamma[(a_{\mathbf{Adv}}(\Gamma) - b_{\mathbf{Adv}}(\Gamma))^2].
\end{aligned}$$

Observe that $\mathcal{A}$ is also an oracle-aided P.P.T.M. adversary, thus should also satisfyEquation (2). Therefore

$$\begin{aligned}
\mathbb{E}_\Gamma[|a_{\mathbf{Adv}}(\Gamma) - b_{\mathbf{Adv}}(\Gamma)|] &\leq \sqrt{\mathbb{E}_\Gamma[(a_{\mathbf{Adv}}(\Gamma) - b_{\mathbf{Adv}}(\Gamma))^2]} \\
&= \sqrt{\mathbb{E}_\Gamma[a_{\mathcal{A}}(\Gamma) - b_{\mathcal{A}}(\Gamma)]} \\
&< \sqrt{\mathsf{negl}(\kappa)} \\
&= \mathsf{negl}(\kappa).
\end{aligned}$$

This completes the proof.

Applying Markov's inequality, we immediately obtain:

Corollary 1. *Under Section 4.2, for any oracle-aided P.P.T.M. adversary* $\mathbf{Adv}^{\Gamma}$ *and* $\kappa \in \mathbb{N}^*$*, with probability at least* $1 - \frac{1}{\kappa^2}$ *over randomness of* Γ*, the following inequality holds:*

$$\left| \Pr_{\mathsf{MSK}, r_{\mathbf{Adv}}}[\mathbf{REAL}(1^\kappa, \mathbf{Adv}^{\Gamma}, \mathsf{MSK}) = 1] - \Pr_{r_{\mathbf{Adv}}, r_S}[\mathbf{IDEAL}(1^\kappa, \mathbf{Adv}^{\Gamma}, S^{\Gamma}) = 1] \right|$$
$$< \kappa^2 \mathsf{negl}(\kappa) = \mathsf{negl}(\kappa). \tag{3}$$

Finally, since $\sum_{\kappa=1}^{\infty} \frac{1}{\kappa^2}$ converges, for any oracle-aided P.P.T.M. adversary $\mathbf{Adv}^{\Gamma}$, by the Borel-Cantelli Lemma, there is only a zero measure set of oracles $\Gamma_{\mathbf{Adv}}$ that violates Equation (3) for infinitely many κ.

Observe that there are only countably many oracle-aided P.P.T.M., so if we exclude $\Gamma_{\mathbf{Adv}}$ for each $\mathbf{Adv}^{\Gamma}$, there still remains measure one of the oracles. For these measure one of oracles, simulation indistinguishability Equation (1) holds for any $\mathbf{Adv}^{\Gamma}$.

B Deferred Proof of Lemma 3

B.1 Proof Overview

Suppose H is a fixed instantiation of the oracle H, r is a fixed randomness of all parties. Define distribution $\mathcal{D}_{H,r}$ and $\mathcal{D}'_{H,r}$ as follows:

Definition 8. *Define* $\mathcal{D}_{H,r}$ *to be the distribution over views generated by the following process*

1. *Fix the randomness of each party to be* r*.*
2. *Fix the* H *oracle in* Γ *to be* H*.*
3. *Sample the rest of* Γ*.*
4. *Run the protocol and output the view* V*.*

In other words, $\mathcal{D}_{H,r}$ is simply the distribution $\mathcal{D}$ conditioned on the H oracle instantiated with H and the randomness of all parties fixed to be r.

Definition 9. $\mathcal{D}'_{H,r}$ *is the distribution over all valid views in which the randomness of all parties is* r*. The probability of a specific valid view* V *with randomness* r *in* $\mathcal{D}'_{H,r}$ *is proportional to* $w_H(V)$*.*

We first prove that $\mathcal{D}_{H,r} \approx \mathcal{D}'_{H,r}$ for all (H, r) pair. The distribution $\mathcal{D}$ can be seen as a uniform mixing of $\mathcal{D}_{H,r}$ for all (H, r). $\mathcal{D}'$, on the other hand, can also be seen as a mixing of $\mathcal{D}'_{H,r}$ for all (H, r) but the mixing is not uniform. Instead, we can easily check that the weight of $\mathcal{D}'_{H,r}$ in this mixing is $\sum_{V \in \mathrm{sup}(\mathcal{D}'_{H,r})} w_H(V)$. Denote this weight as $w_{H,r}$. We will prove that $w_{H,r} \approx 1$ for every (H, r) pair. This means, $\mathcal{D}'$ can be seen as a mixing of $\mathcal{D}'_{H,r}$ in an **almost** uniform way. This is sufficient to prove that $\mathcal{D} \approx \mathcal{D}'$.

B.2 Valid Oracles

For the rest of this section, the H oracle is fixed to be H and the randomness of all parties is fixed to be r. In order to prove that $\mathcal{D}_{H,r} \approx \mathcal{D}'_{H,r}$ and $w_{H,r} \approx 1$, we first establish some definitions and lemmas.

Our first step is to define valid oracles analogues to valid views. For any $\kappa \in [\log \lambda, \kappa_{\max}]$, we say an oracle Γ_κ is a **κ-valid oracle** if it satisfies the following properties:

- The Keygen_κ oracle in Γ_κ only contains collisions in the form of hybrid groups.
- The correctness of PCPRF holds under H. Specifically, if $\mathsf{Keygen}_\kappa(k, c) = k'$ and $c^H(x) = 1$, then $\mathsf{F}(k, x) = \mathsf{f}(k', x)$.

Note that, if a valid oracle Γ_κ contains a collision of Keygen_κ, it can never be sampled in the real protocol.

Analogues to a valid view, we can also define $w_H(\Gamma_\kappa)$ for a κ-valid oracle Γ_κ: each constrained relation $k[c] = k'$ contributes a factor of $|K'_\kappa|^{-1}$; the F and f queries in the same equivalency class contribute a factor of $|Y_\kappa|^{-1}$ in total. However, in this case, the number of constrained relations is fixed to be $|K_\kappa| \cdot |C_\kappa|$ since each pair of $(k, c) \in K_\kappa \times C_\kappa$ corresponds to a unique constrained relation. Actually, the number of equivalency classes is also fixed. Consider a specific $x \in X_\kappa$. Consider the number of equivalency classes among all $\mathsf{F}(\cdot, x)$ and $\mathsf{f}(\cdot, x)$ queries. First, if Keygen_κ is injective, then we can see that the number of equivalency classes does not depend on Keygen_κ. Suppose k_1 and k_2 are two master keys of normal groups and $\mathsf{Keygen}(k_1, c_1) = k'_1, \mathsf{Keygen}(k_2, c_2) = k'_2$. If we change $\mathsf{Keygen}(k_2, c_2)$ to k'_1, the normal groups of k_1 and k_2 are jointed to be a hybrid group. The equivalency class of $\mathsf{f}(k_1[c_1], x)$ and the equivalency class of $\mathsf{f}(k_2[c_2], x)$ are merged into 1 equivalency class. In the meanwhile, since the value of $\mathsf{Keygen}(k_1, c_1)$ is changed, k_1 becomes an isolated constrained key in Γ_κ. Thus, $\mathsf{f}(k_1, x)$ becomes an equivalency class itself. Therefore, the number of equivalency classes does not change. Using this technique, we can obtain any possible Keygen_κ from an an injective one, while not changing the number of equivalency classes. Thus, the number of equivalency classes is the same across all possible κ-valid oracles.

Lemma 10. *For any $\kappa \in \mathbb{N}^*$, the sum of $w_H(\Gamma_\kappa)$ of all κ-valid oracles Γ_κ is no more than 1.*

Proof. Note that, the weight of Γ_κ can be expressed in the form of $|K'_\kappa|^{-a}|Y_\kappa|^{-b}$, where $a = |K_\kappa| \cdot |C_\kappa|$ is the number of constrained relations; b is the number of F and f equivalency classes which is also fixed. Thus, we only need to show that the number of possible κ-valid oracle Γ_κ is at most $|K'_\kappa|^a|Y_\kappa|^b$. This is obvious: the number of possible Keygen_κ is bounded by $|K'_\kappa|^{|K_\kappa| \cdot |C_\kappa|}$ because Keygen_κ is a function of the form $K_\kappa \times C_\kappa \to K'_\kappa$. Once Keygen_κ is determined, there are exactly $|Y_\kappa|^b$ ways to determine the rest of F_κ and $\mathsf{f}(\kappa)$. Thus, the number of possible Γ_κ is bounded by $|K'_\kappa|^a|Y_\kappa|^b$. □

Define $w_H^\kappa(V)$ as the product of contributions made by Γ_κ queries in V. Then, $w_H(V) = \prod_{\kappa=\log\lambda}^{\kappa_{\max}} w_H^\kappa(V)$.

Definition 10. *We say a κ valid oracle Γ_κ is an κ-expansion of a valid view V if:*

- *Γ_κ does not contradict the queries made in V.*
- *For a key that appears in V, the type of group to which this key belongs is the same in V and Γ_κ. Specifically, if $k' \in K'_\kappa$ is an isolated constrained key in V, then it is also an isolated constrained key in Γ_κ; if $k \in K_\kappa$ or $k' \in K'_\kappa$ belongs to a normal group in V, it also belongs to a normal group in Γ_κ; if $k \in K_\kappa$ or $k' \in K'_\kappa$ belongs to a hybrid group in V, it also belongs to a hybrid group in Γ_κ.*
- *Except the collisions in V, there should be no more collisions of Keygen_κ in Γ_κ.*

Denote the set of all κ-expansions of V as $C_\kappa(V)$. The next lemma shows that $w_H^\kappa(V)$ is estimated by the sum of weights of all its κ-expansions.

Lemma 11. *For any $\kappa \in [\log\lambda, \kappa_{\max}]$ and valid view V where the number of Γ_κ queries in V is at most $\mathsf{poly}(2^\kappa)$, we have*

$$w_H^\kappa(V) \approx_s \sum_{\Gamma_\kappa \in C_\kappa(V)} w_H(\Gamma_\kappa).$$

Here, $a \approx_s b$ means $|\frac{a}{b} - 1| = \mathsf{snegl}(\kappa)$.

Proof. Suppose $w_H^\kappa(V) = |K'_\kappa|^{-a^*}|Y_\kappa|^{-b^*}$ and $w(\Gamma_\kappa) = |K'_\kappa|^{-a}|Y_\kappa|^{-b}$ for all $\Gamma_\kappa \in C_\kappa(V)$. Here a^* is the number of constrained relations $k[c] = k'$ in V, where $k \in K_\kappa, c \in C_\kappa, k' \in K'_\kappa$. b^* is the number of non-equivalent F_κ and f_κ queries in V. a and b are the same as in the proof of Lemma 11.

We can generate all κ-expansions in $C_\kappa(V)$ in following way:

1. Determine Keygen_κ first. We first count the number of possible Keygen_κ in this step:
   - We determine the value of $\mathsf{Keygen}_\kappa(k, c)$ for all $(k, c) \in K_\kappa \times C_\kappa$ in the lexicographical order. Except for a^* of them which has already been determined by P_κ, we still need to determine the value of $\mathsf{Keygen}_\kappa(k, c)$ for $|K_\kappa||C_\kappa| - a^*$ pairs of (k, c).
   - While we are determining the value of $\mathsf{Keygen}_\kappa(k, c)$, at most $|K_\kappa| \times |C_\kappa| + p$ values in K'_κ is already taken (here p is the number of isolated constrained keys in P_κ, which is at most $\mathsf{poly}(2^\kappa)$). The reason that we cannot choose an isolated constrained key to be $\mathsf{Keygen}(k, c)$ is that an isolated constrained key in V should still be an isolated constrained key in an κ-expansion of V. Since the number of Γ_κ queries made in V is bounded by $\mathsf{poly}(2^\kappa)$, we assume that $p < 2^{c\kappa}$, where c is a constant.

Thus, there are still at least $|K'_\kappa| - |K_\kappa||C_\kappa| - p$ left for $\mathsf{Keygen}(k, c)$. So the number of possible Keygen_κ is **at least**

$$\begin{aligned}
&(|K'_\kappa| - |K_\kappa||C_\kappa| - p)^{|K_\kappa||C_\kappa|-a^*} \\
\geq &\left(2^{\kappa^2} - 2^{2\kappa} - 2^{c\kappa}\right)^{2^{2\kappa}-a^*} \\
\approx_s &\left(2^{\kappa^2}\right)^{2^{2\kappa}-a^*} \\
= &|K'_\kappa|^{|K_\kappa||C_\kappa|-a^*}.
\end{aligned}$$

Clearly, the number of possible Keygen_κ is no more than $|K'_\kappa|^{|K_\kappa||C_\kappa|-a^*}$. Thus, the number of possible Keygen_κ can be $(1\pm\mathsf{snegl}(\kappa))$-approximated by $|K'_\kappa|^{|K_\kappa||C_\kappa|-a^*}$.

2. For any fixed Keygen_κ, we want to determine $\mathsf{F}_\kappa(k, x)$ and $\mathsf{f}_\kappa(k', x)$ for any k, k', x. There are b values that need to be determined and b^* of them are already determined by the queries in V. (Because κ-expansion keeps the type of groups to which the keys in V belong, non-equivalent F_κ or f_κ queries in V are still non-equivalent in a κ-expansion.) Thus, for each Keygen_κ, there are $|Y_\kappa|^{b-b^*}$ ways to determine F_κ and f_κ.
3. To sum up, $|C_\kappa(V)| \approx_s |K'_\kappa|^{|K_\kappa||C_\kappa|-a^*}|Y_\kappa|^{b-b^*}$. Then,

$$\begin{aligned}
&\sum_{\Gamma_\kappa \in C_\kappa(V)} w(\Gamma_\kappa) \\
= &|C_\kappa(V)||K'_\kappa|^{-|K_\kappa||C_\kappa|}|Y_\kappa|^{-b} \\
\approx_s &|K'_\kappa|^{-a^*}|Y_\kappa|^{-b^*} \\
= &w_H^\kappa(V).
\end{aligned}$$

□

B.3 $\mathcal{D}_{H,r} \approx \mathcal{D}'_{H,r}$ and $w_{H,r} \approx 1$

After the preparation, we now start to prove that $\mathcal{D}_{H,r} \approx \mathcal{D}'_{H,r}$ and $w_{H,r} = \sum_{V:r(V)=r} w_H(V) \approx 1$.

Given two different valid views in $\sup(\mathcal{D}'_{H,r})$, say V and V'. Since V and V' are different, but the algorithm of each party can be seen as determinant because the randomness in these two views are both r, there must be a first Γ_κ query that has a different answer in V and V' for some $\kappa \in [\log_2 \lambda, \kappa_{\max}]$. Then oracles belonging to $C_\kappa(V)$ and $C_\kappa(V')$ must contradict with each other. That is,

$$\{(\Gamma_\kappa)_{\kappa\in[\log\lambda,\kappa_{\max}]} : \Gamma_\kappa \in C_\kappa(V)\} \cap \{(\Gamma'_\kappa)_{\kappa\in[\log\lambda,\kappa_{\max}]} : \Gamma'_\kappa \in C_\kappa(V')\} = \emptyset.$$

Then we have

$$\begin{aligned}\sum_{V\in\mathsf{Sup}(\mathcal{D}'_{H,r})} w_H(V) &= \sum_{V\in\mathsf{Sup}(\mathcal{D}'_{H,r})} \prod_{\kappa>\log_2\lambda}^{\kappa_{\max}} w_H^{\kappa}(V)\\ &\approx \sum_{V\in\mathsf{Sup}(\mathcal{D}'_{H,r})} \prod_{\kappa>\log_2\lambda}^{\kappa_{\max}} \left(\sum_{\Gamma_\kappa\in C_\kappa(V)} w_H(\Gamma_\kappa)\right)\\ &\leq \prod_{\kappa>\log_2\lambda}^{\kappa_{\max}} \left(\sum_{\Gamma_\kappa} w(\Gamma_\kappa)\right)\\ &\leq 1.\end{aligned}$$

Here, the approximation in the second line is due to Lemma 11. Concretely, since the number of queries in V is bounded by $\mathsf{poly}(\lambda) = \mathsf{poly}(2^\kappa)$, weight of each P_κ can be $(1 \pm \mathsf{sngl}(\kappa)) = (1 \pm \mathsf{negl}(\lambda))$-approximated for any $\kappa > \log_2\lambda$ by the sum of weights of all its κ-expansions. Since there are only polynomial many possible κ, so the product is also $(1 \pm \mathsf{negl}(\lambda))$-approximated. The last summation is over all possible valid oracles Γ_κ of length κ. The last inequality follows Lemma 10.

In conclusion, we have

$$w_{H,r} = \sum_{V\in\mathsf{Sup}(\mathcal{D}'_{H,r})} w_H(V) \leq 1 + \mathsf{negl}(\lambda).$$

We now consider all valid views $V \in \sup(\mathcal{D}_{H,r})$ satisfying the following property:

- For any $\kappa \in [\log_2\lambda, \kappa_{\max}]$, for any two F_κ or f_κ queries in V, if they are non-equivalent under V, H, their answers are different.

The views with this property are called "**nice**". We now argue that nice views happen with almost 1 probability in $\mathcal{D}_{H,r}$.

We now fix a $\kappa > \log_2\lambda$. For randomly sampled Γ (without collision of Keygen) with H oracle instantiated as H, we argue that the following two events happen with $1 - \mathsf{negl}(\lambda)$ probability.

- **Event 1:** In Γ_κ, any two non-equivalent F, f queries have different answers.
- **Event 2:** Suppose view V is generated by running the algorithm with Γ. Then, any isolated constrained key $k' \in K'_\kappa$ in V is also an isolated constrained key in Γ_κ.

We first prove that event 1 happens with $1 - \mathsf{negl}(\lambda)$ probability. Note that $|Y_\kappa| = 2^{\kappa^3}$, but there are only $(|K_\kappa| + |K'_\kappa|) \times |X_\kappa| \leq 2^{2\kappa^2}$ possible inputs for F_κ and f_κ. With probability $1 - \mathsf{sngl}(\kappa) = 1 - \mathsf{negl}(\lambda)$, any two non-equivalent queries will have different answers, since they are sampled independently and uniformly from Y_κ.

Then, we prove that event 2 happens with $1 - \mathsf{negl}(\lambda)$ probability. Consider the first time a constrained -key $k' \in K'_\kappa$ appears **as an input of an oracle query** in view V. If k' is not generated by the Keygen oracle, conditioned on everything so far, the probability that k' is a constrained key in a normal group of Γ_κ is $\mathsf{snegl}(\kappa) = \mathsf{negl}(\lambda)$ because the constrained keys only take a super-negligible proportion in K'_κ. Thus, event 2 happens with probability $1 - \mathsf{negl}(\lambda)$ by a union bound.

Thus, the probability that these events both happen is also $1 - \mathsf{negl}(\lambda)$. Since $\kappa \in [\log \lambda, \kappa_{\max}]$, the probability of these two events both happen for every κ is also $1 - \mathsf{negl}(\lambda)$. If event 1 and 2 both happen, we argue that this view is nice. If not, there exists 2 F_κ or f_κ queries which are non-equivalent under V, H but have the same answer. Since event 1 happens, these two queries must be equivalent under Γ_κ. Since Keygen_κ does not contain collisions, the first inputs of these two queries must be in the same normal group of Γ_κ. Thus, at least one of these two keys must be an isolated constrained key in V, this contradicts to the fact that event 2 happens.

Now we argue that for any nice view $V \in \sup(\mathcal{D}_{H,r})$, the probability of V in $\mathcal{D}_{H,r}$ is approximately the same as $w_H(V)$. Since the randomness is fixed, we only need to compute the probability of a randomly sampled Γ_κ (without collision of Keygen) consistent with V for all $\kappa \in [\log \lambda, \kappa_{\max}]$.

Consider a fixed κ. Let $S \subseteq K'_\kappa$ be all isolated constrained keys in V, then we claim that, among all possible Γ_κ satisfying V, only a negligible portion of them violates the property that any $k' \in S$ is also an isolated constrained key in Γ_κ. We say the rest of Γ_κ are "isolated", meaning that any isolated constrained key in V is also isolated constrained key in Γ_κ, and it's not hard to see that isolated Γ_κ exactly form the set of κ-expansions $C_\kappa(V)$ since V contains no collisions of Keygen_κ.

We can enumerate all Γ_κ satisfying V in the following steps:

1. Sample Keygen_κ;
2. Sample $\mathsf{F}_\kappa, \mathsf{f}_\kappa$ conditioned on Keygen_κ.

We have argue that, for any possible Keygen_κ sampled in step 1, the number of possible Γ_κ is the same in step 2, because the number of equivalency classes of F_κ and f_κ is fixed. Now we turn back to step 1. We only need to compute the proportion of Keygen_κ which makes some isolated constrained key k' in V belonging to a normal group. This only takes up a small portion since it can be bounded by

$$
\begin{aligned}
&1 - \prod_{i=1}^{|K_\kappa||C_\kappa|} \frac{|K'_\kappa| - |S| - i + 1}{|K'_\kappa| - i + 1} = 1 - \prod_{i=1}^{2^{2\kappa}} \frac{2^{\kappa^2} - |S| - i + 1}{2^{\kappa^2} - i + 1} \\
\leq &1 - \left(\frac{2^{\kappa^2} - 2^{2\kappa} + 1 - |S|}{2^{\kappa^2} - 2^{2\kappa} + 1} \right)^{2^{2\kappa}} \leq 2^{2\kappa} \cdot \left(1 - \frac{2^{\kappa^2} - 2^{2\kappa} + 1 - |S|}{2^{\kappa^2} - 2^{2\kappa} + 1} \right) \\
= &\frac{2^{2\kappa}|S|}{2^{\kappa^2} - 2^{2\kappa} + 1} = \mathsf{snegl}(\kappa) = \mathsf{negl}(\lambda).
\end{aligned}
$$

On the other hand, for each isolated Γ_κ, we claim its probability is close to its weight: we have shown its weight is just $|K'_\kappa|^{-a}|Y_\kappa|^{-b}$ where $a = |K_\kappa||C_\kappa|$ and b is the number of equivalent classes, and we can obtain its probability is $\left(\prod_{i=1}^{a} \frac{1}{|K'_\kappa|-i+1}\right)|Y_\kappa|^{-b}$. Their ratio is

$$\prod_{i=1}^{a} \frac{|K'_\kappa|-i+1}{|K'_\kappa|} = \prod_{i=1}^{2^{2\kappa}} \frac{2^{\kappa^2}-i+1}{2^{\kappa^2}} \geq \left(\frac{2^{\kappa^2}-2^{2\kappa}}{2^{\kappa^2}}\right)^{2^{2\kappa}} = 1-\mathsf{snegl}(\kappa) = 1-\mathsf{negl}(\lambda)$$

This means that they are $(1 \pm \mathsf{negl}(\lambda))$-close to each other.

We have already proved that $\Pr_{\Gamma_\kappa}[\Gamma_\kappa \models V] \approx w_H^\kappa(V)$. Again since $\kappa \in [\log\lambda, \kappa_{\max}]$ where $\kappa_{\max}$ is polynomial over λ, the probability of V in $\mathcal{D}_{H,r}$ is approximately the same as $w_H(V)$ for any nice view V:

$$\begin{aligned}
\Pr_{V' \leftarrow \mathcal{D}_{H,r}}[V' = V] &= \Pr_{\Gamma}[\Gamma \models V] \\
&= \prod_{\kappa > \log_2 \lambda}^{\kappa_{\max}} \Pr_{\Gamma_\kappa}[\Gamma_\kappa \models V] \\
&\approx \prod_{\kappa > \log_2 \lambda}^{\kappa_{\max}} w_H^\kappa(V) \\
&= w_H(V).
\end{aligned}$$

Thus,

$$\begin{aligned}
w_{H,r} &= \sum_{V \in \mathsf{Sup}(\mathcal{D}'_{H,r})} w_H(V) \\
&\geq \sum_{\text{nice view } V} w_H(V) \\
&\approx \sum_{\text{nice view } V} \Pr_{V' \leftarrow \mathcal{D}_{H,r}}[V' = V] \\
&\approx \sum_{V \in \mathsf{Sup}(\mathcal{D}_{H,r})} \Pr_{V' \leftarrow \mathcal{D}_{H,r}}[V' = V] \\
&= 1
\end{aligned}$$

Since we have proved that $w_{H,r} \leq 1 + \mathsf{negl}(\lambda)$, we can conclude that $w_{H,r} \approx 1$. We can also conclude that $\sum_{\text{nice view } V} w_H(V) \approx \sum_{V \in \mathsf{Sup}(\mathcal{D}'_{H,r})} w_H(V) \approx 1$, i.e., nice views occupy a $1 - \mathsf{negl}(\lambda)$ fraction of both $\mathcal{D}_{H,r}$ and $\mathcal{D}'_{H,r}$, so we conclude $\Delta(\mathcal{D}_{H,r}, \mathcal{D}'_{H,r}) \leq \Delta(\mathcal{D}_{H,r}, \mathcal{D}_{H,r}\ |_{\text{nice}}) + \Delta(\mathcal{D}_{H,r}\ |_{\text{nice}}, \mathcal{D}'_{H,r}\ |_{\text{nice}}) + \Delta(\mathcal{D}'_{H,r}\ |_{\text{nice}}, \mathcal{D}'_{H,r}) = \mathsf{negl}(\lambda)$.

C Deferred Proofs of Section 7

The proof of Lemma 6 and Lemma 7 are basically the same, differing only in a few places. The proof of Lemma 7 is relatively more complicated. Thus, we

only present the proof of Lemma 7 and point out what is changed in the proof of Lemma 6.

C.1 Proof of Lemma 7

We show that the probability that one of the following occurs is greater than $1 - p(\lambda)/2$ for a view V sampled from $\mathcal{D}$.

1. $Q(V_A^{r-1}) \cap Q(V_B^{r,j-1}) \not\subseteq Q(V_E^{r,m})$
2. $q_{r,j} \notin Q(V_A^{r-1})$
3. $q_{r,j} \in Q(V_E^{r,2m})$

We now fix the round number r and the query number $j \geq 2$. We write $(M^{r-1}, V_E^{r,m}, V_B^{r,j-1}) = G$ as a whole variable. Sometimes we also denote it as $G(V)$ for clarity. Define $\mathcal{G}$ to be the distribution $\{G(V)|V \leftarrow \mathcal{D}\}$.

For every possible G, write

$$\mathcal{D}_G = \{V \leftarrow \mathcal{D}|G(V) = G\}$$

In other words, $\mathcal{D}_G$ is distribution $V \leftarrow \mathcal{D}$ conditioned on $G(V) = G$.

For each G, partition $\mathcal{D}_G$ into three restricted distributions:

- $P_1(G) = \{V \leftarrow \mathcal{D}_G|V \not\models \mathsf{Condition1}, V \not\models \mathsf{Condition2}, (\exists i \geq 0, \Pr_{U \leftarrow \mathcal{D}_G}[U \models \mathsf{Condition1} \text{ or } 2|U_E^{r,m+i} = V_E^{r,m+i}]) > 1 - p/6\}$
- $P_2(G) = \{V \leftarrow \mathcal{D}_G|V \not\models \mathsf{Condition1}, V \not\models \mathsf{Condition2}, (\forall i \geq 0, \Pr_{U \leftarrow \mathcal{D}_G}[U \models \mathsf{Condition1} \text{ or } 2|U_E^{r,m+i} = V_E^{r,m+i}]) \leq 1 - p/6\}$
- $P_3(G) = \{V \leftarrow \mathcal{D}_G|V \models \mathsf{Condition1} \text{ or } 2\}$

Clearly, the support set of P_1, P_2, P_3 are disjoint, and they together form the support set of $\mathcal{D}_G$.

Define $s_i(G) = \Pr_{V \leftarrow \mathcal{D}_G}[V \in \sup(P_i(G))]$, $i = 1, 2, 3$. Then we have $s_1(G) + s_2(G) + s_3(G) = 1$.

Claim 1 *For all possible G, $s_1(G) < p/6$.*[3]

Proof. For each possible $V \leftarrow \mathcal{D}_G$, assign a number First_V: the first i such that

$$\Pr_{U \leftarrow \mathcal{D}_G}[U \models \mathsf{Condition1} \text{ or } 2|U_E^{r,m+i} = V_E^{r,m+i}] > 1 - p/6$$

If there is no such i, then $\mathsf{First}_V = \infty$.

Define an equivalent relation $\sim$ for $\mathsf{sup}(\mathcal{D}_G)$ as follows: all V s.t. $\mathsf{First}_V = \infty$ are equivalent to each other. For V s.t. $\mathsf{First}_V \neq \infty$ and another U, we say $U \sim V$ if $U_E^{r,m+\mathsf{First}_V} = V_E^{r,m+\mathsf{First}_V}$. This means, Eve's view until segment $m + \mathsf{First}_V$ is entirely the same in U and V. Thus, $\mathsf{First}_U = \mathsf{First}_V$. Therefore, we have

$$U \sim V \Leftrightarrow (\mathsf{First}_U, U_E^{r,m+\mathsf{First}_U}) = (\mathsf{First}_V, V_E^{r,m+\mathsf{First}_V})$$

[3] The proof of this claim is similar to [18, Claim 1].

This confirms that $\sim$ is indeed an equivalent relation. Naturally, $\sim$ divides the support set of $\mathcal{D}_G$ into many blocks. Denote Block_V to be all the views U s.t. $U \sim V$. If $\mathsf{First}_V = \infty$, we simply write Block_V as Block_∞. For a Block_V, define $\mathcal{D}_{G,\mathsf{Block}_V} = \{U \leftarrow \mathcal{D}_G | U \in \mathsf{Block}_V\}$. That is, $\mathcal{D}_{G,\mathsf{Block}_V}$ is the conditional distribution of $\mathcal{D}_G$ over Block_V.

We will show that each $\mathcal{D}_{G,\mathsf{Block}_V}$ has a small probability mass distributed on the support set of $P_1(G)$: the blocks Block_∞ has no part in P_1. Consider one of the remaining block Block_V. The proportion of $\mathcal{D}_{G,\mathsf{Block}_V}$ **not** in $P_1(G)$ is given by:

$$\Pr_{U \leftarrow \mathcal{D}_G}[U \models \mathsf{Condition1} \text{ or } \mathsf{Condition2} | U_E^{r,m+\mathsf{First}_V} = V_E^{r,m+\mathsf{First}_V}].$$

which by the definition of First_V is greater than $1 - p/6$.

Thus, we have proved the desired claim. □

We have proved that $s_1(G)$ is small for all possible G. By definition, $P_3(G)$ satisfies condition 1 or 2, thus can only help us. It remains to bound the probability of the views in $P_2(G)$ that does not satisfy condition 3.

Claim 2 $\Pr_{V \leftarrow \mathcal{D}}[V \in P_2(G(V)), V \not\models \mathsf{Condition3}] < p/3$.

Proof. Observe that, if for most of G, $P_2(G)$ only takes up a small portion of $\mathcal{D}_G$, then Claim 2 trivially holds. Thus, we make the following definition

Definition 11. *We say G is "**normal**" if $s_2(G) > p/6$.*

Denote $\mathcal{G}_{\text{normal}} = \{G \leftarrow \mathcal{G} | G \text{ is normal}\}$,$\mathcal{G}_{\neg\text{normal}} = \{G \leftarrow \mathcal{G} | G \text{ is not normal}\}$.
If $\Pr_{G \leftarrow \mathcal{G}}[G \text{ is normal}] < p/6$, then we have

$$\begin{aligned}
&\Pr_{V \leftarrow \mathcal{D}}[V \in P_2(G(V))] \\
=&\mathbb{E}_{G \leftarrow \mathcal{G}}[s_2(G)] \\
=&\Pr_{G \leftarrow \mathcal{G}}[G \text{ is normal}]\mathbb{E}_{G \leftarrow \mathcal{G}_{\text{normal}}}[s_2(G)] + \Pr_{G \leftarrow \mathcal{G}}[G \text{ is not normal}]\mathbb{E}_{G \leftarrow \mathcal{G}_{\neg\text{normal}}}[s_2(G)] \\
\leq&\Pr_{G \leftarrow \mathcal{G}}[G \text{ is normal}] + \Pr_{G \leftarrow \mathcal{G}}[G \text{ is not normal}] \cdot \frac{p}{6} \\
=&\Pr_{G \leftarrow \mathcal{G}}[G \text{ is normal}] + \left(1 - \Pr_{G \leftarrow \mathcal{G}}[G \text{ is normal}]\right) \cdot \frac{p}{6} \\
=&(1 - \frac{p}{6}) \Pr_{G \leftarrow \mathcal{G}}[G \text{ is normal}] + \frac{p}{6} \\
\leq&(1 - \frac{p}{6})\frac{p}{6} + \frac{p}{6} \\
<&\frac{p}{3}
\end{aligned}$$

which implies Claim 2 is proved. In the rest of the proof we assume that $\Pr_{G \leftarrow \mathcal{G}}[G \text{ is normal}] \geq p/6$.

Our next goal is to prove Claim 2 conditioned on G being normal. Formally,

Claim 3 $\mathbb{E}_{G \leftarrow \mathcal{G}_{normal}}[\Pr_{V \leftarrow \mathcal{D}_G}[V \in P_2(G), V \not\models \mathsf{Condition3}]] < p/6$

The proof of Claim 3 is shown in the next section.

Claim 3 is sufficient to show Claim 2:

$$
\begin{aligned}
&\Pr_{V\leftarrow\mathcal{D}}[V\in P_2(G(V)), V\not\models \mathsf{Condition3}]\\
=&\mathbb{E}_{G\leftarrow\mathcal{G}}[\Pr_{V\leftarrow\mathcal{D}_G}[V\in P_2(G), V\not\models \mathsf{Condition3}]]\\
=&\Pr_{G\leftarrow\mathcal{G}}[G \text{ is normal}]\mathbb{E}_{G\leftarrow\mathcal{G}_{\text{normal}}}[\Pr_{V\leftarrow\mathcal{D}_G}[V\in P_2(G), V\not\models \mathsf{Condition3}]]\\
&+\Pr_{G\leftarrow\mathcal{G}}[G \text{ is not normal}]\mathbb{E}_{G\leftarrow\mathcal{G}_{\neg\text{normal}}}[\Pr_{V\leftarrow\mathcal{D}_G}[V\in P_2(G), V\not\models \mathsf{Condition3}]]\\
\le&\Pr_{G\leftarrow\mathcal{G}}[G \text{ is normal}]\cdot\frac{p}{6}\\
&+\Pr_{G\leftarrow\mathcal{G}}[G \text{ is not normal}]\mathbb{E}_{G\leftarrow\mathcal{G}_{\neg\text{normal}}}[s_2(G)]\\
\le&\Pr_{G\leftarrow\mathcal{G}}[G \text{ is normal}]\cdot\frac{p}{6}+\Pr_{G\leftarrow\mathcal{G}}[G \text{ is not normal}]\cdot\frac{p}{6}\\
=&\frac{p}{6}\\
<&\frac{p}{3}
\end{aligned}
$$

□

With Claim 1 and Claim 2, we can prove Lemma 7.

Proof (Proof of Lemma 7). By Claim 1 and Claim 2,

$$
\begin{aligned}
&\Pr_{V\leftarrow\mathcal{D}}[V\not\models \mathsf{Condition1,2,3}]\\
\le&\Pr_{V\leftarrow\mathcal{D}}[V\in P_1(G(V))]+\Pr_{V\leftarrow\mathcal{D}}[V\in P_2(G(V)), V\not\models \mathsf{Condition3}]\\
=&\mathbb{E}_{G\leftarrow\mathcal{G}}[s_1(G)]+\Pr_{V\leftarrow\mathcal{D}}[V\in P_2(G(V)), V\not\models \mathsf{Condition3}]\\
<&\frac{p}{6}+\frac{p}{3}\\
=&\frac{p}{2}.
\end{aligned}
$$

□

The Efficacy of Eve's Simulation In order to prove Claim 3, we prove an intermediate result which is Claim 4. It shows that Eve's simulation and Alice's view, conditioned on miss, is close with high probability.

Suppose G is sampled from $\mathcal{G}_{\text{normal}}$. By the definition of $P_2(G)$

$$\forall V\in P_2(G), \forall i\ge 0, \Pr_{U\leftarrow\mathcal{D}_G}[U\models \mathsf{Condition1} \text{ or } 2|U_E^{r,m+i}=V_E^{r,m+i}]\le 1-p/6.$$

Equivalently, we can describe the probability of the complementary event:

$$\forall V\in P_2(G), \forall i\ge 0,$$

$$\Pr_{U\leftarrow\mathcal{D}_G}\left[\underbrace{Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m})}_{\neg\mathsf{Condition1}}, \underbrace{q_{r,j}\in Q(U_A^{r-1})}_{\neg\mathsf{Condition2}}\middle| U_E^{r,m+i}=V_E^{r,m+i}\right]\ge p/6$$

Since Eve's view is ever expanding, we can change $Q(V_E^{r,m})$ to $Q(V_E^{r,m+i})$ in the above inequality, which is

$$\forall V \in P_2(G), \forall i \geq 0,$$
$$\Pr_{U \leftarrow \mathcal{D}_G}\left[Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i}), q_{r,j} \in Q(U_A^{r-1}) \middle| U_E^{r,m+i} = V_E^{r,m+i}\right] \geq p/6$$

this implies two weaker statements

$$\forall V \in P_2(G), \forall i \geq 0$$
$$\Pr_{U \leftarrow \mathcal{D}_G}\left[q_{r,j} \in Q(U_A^{r-1}) \middle| Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i}), U_E^{r,m+i} = V_E^{r,m+i}\right] \geq p/6 \tag{4}$$

and

$$\forall V \in P_2(G), \forall i \geq 0, \Pr_{U \leftarrow \mathcal{D}_G}\left[Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i}) \middle| U_E^{r,m+i} = V_E^{r,m+i}\right] \geq p/6 \tag{5}$$

We first make the following definition:

Definition 12. *Denote Φ to be the distribution over views generated by the following process: First sample $G \leftarrow \mathcal{G}_{normal}$, then sample $V \leftarrow P_2(G)$.*

The following claim is essential in our proof. It shows that Eve's algorithm indeed simulates Alice's view. It corresponds to [18, Claim 2], which is the major challenge in generalizing the technique of [18] to our case.

Claim 4 *Fix a segment number $m+i$. For a view $V \in \sup(\mathcal{D})$, define two distributions over partial views of Alice:*

$$\mathcal{S}_0(V) = \left\{U_A^{r-1} \middle| U \leftarrow \mathcal{D}_{G(V)}, U_E^{r,m+i} = V_E^{r,m+i}, Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})\right\}$$

and

$$\mathcal{S}_1(V) = \left\{\mathsf{Sim}_A \leftarrow \mathsf{SIM}(M^{r-1}(V), V_E^{r,m+i}) \middle| Q(\mathsf{Sim}_A) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})\right\}.$$

Then we have

$$\mathbb{E}_{V \leftarrow \Phi}[\Delta(\mathcal{S}_0(V), \mathcal{S}_1(V))] = \mathsf{negl}(\lambda).$$

Here, $\mathcal{S}_0(V)$ is the distribution of Alice's view conditioned on G and Eve has included all the intersection elements. $\mathcal{S}_1(V)$ is the distribution of Eve's simulated view of Alice in segment $i+1$ conditioned on "miss".

Proof. In the following, we write $\overline{V} = V^{r,m+i,j-1}$ for the sake of simplicity. Similarly, $\overline{\mathcal{D}} = \mathcal{D}^{r,m+i,j-1}, \overline{\mathcal{D}'} = \mathcal{D}'^{r,m+i,j-1}$.

We first observe that the distributions $\mathcal{S}_0(V), \mathcal{S}_1(V)$ only depend on $G(V)$ and $V_E^{r,m+i}$ (recall that $G = (M^{r-1}, V_E^{r,m}, V_B^{r,j-1})$), thus we denote $(G, V_E^{r,m+i})$ as a new variable G'. For clarity, we write $G'(V)$ to specify that G' is in the view V.

We have proved that $\Delta(\overline{\mathcal{D}}, \overline{\mathcal{D}'}) \leq \mathsf{negl}(\lambda)$. Suppose $\Delta(\overline{\mathcal{D}}, \overline{\mathcal{D}'}) = \varepsilon$. In Lemma 1, we substitute $\mathcal{S}_1, \mathcal{S}_2, f$ by $\overline{\mathcal{D}}, \overline{\mathcal{D}'}, G'$, we get

$$\mathbb{E}_{\overline{V} \leftarrow \overline{\mathcal{D}}}\Big[\Delta\big(\{\overline{U}\big|G'(\overline{U}) = G'(\overline{V}), \overline{U} \leftarrow \overline{\mathcal{D}}\},\\ \{\overline{U}\big|G'(\overline{U}) = G'(\overline{V}), \overline{U} \leftarrow \overline{\mathcal{D}'}\}\big)\Big] \leq 2\varepsilon$$

Instead of sampling $\overline{V} \leftarrow \overline{\mathcal{D}}$, we sample $V \leftarrow \mathcal{D}$, then output the partial view $\overline{V}$. As mentioned before, these two sampling ways are equivalent. So we have

$$\mathbb{E}_{V \leftarrow \mathcal{D}}\Big[\Delta\big(\{\overline{U}\big|G'(\overline{U}) = G'(\overline{V}), \overline{U} \leftarrow \overline{\mathcal{D}}\},\\ \{\overline{U}\big|G'(\overline{U}) = G'(\overline{V}), \overline{U} \leftarrow \overline{\mathcal{D}'}\}\big)\Big] \leq 2\varepsilon$$

Note that the following two sampling processes yield the same distribution:

1. Sample $V \leftarrow \mathcal{D}$.
2. First sample $G \leftarrow \mathcal{G}$, then sample $V \leftarrow \mathcal{D}_G$. (Recall that $\mathcal{G} = \{G(V)|V \leftarrow \mathcal{D}\}$.)

Observe that in the second process, with probability at least $p/6$, G is normal; if G is normal, then with probability at least $p/6$, $V \in \sup(P_2(G))$. Using the inequality $\mathbb{E}[X] \geq \Pr[A]\mathbb{E}[X|A]$, we can replace distribution $\mathcal{D}$ by distribution Φ and get:

$$\mathbb{E}_{V \leftarrow \Phi}\Big[\Delta\big(\{\overline{U}|G'(\overline{U}) = G'(\overline{V}), \overline{U} \leftarrow \overline{\mathcal{D}}\},\\ \{\overline{U}|G'(\overline{U}) = G'(\overline{V}), \overline{U} \leftarrow \overline{\mathcal{D}'}\}\big)\Big] \leq 2\varepsilon \cdot (\frac{6}{p} \cdot \frac{6}{p}) = \frac{72\varepsilon}{p^2}$$

For any $V \leftarrow \Phi$, by the definition of Φ, $V \in P_2(G(V))$. By Equation (5),

$$\Pr_{U \leftarrow \mathcal{D}_G}\Big[Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})\Big|U_E^{r,m+i} = V_E^{r,m+i}\Big] \geq p/6$$

Equivalently

$$\Pr_{U \leftarrow \mathcal{D}}\Big[Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})\Big|G'(U) = G'(V)\Big] \geq p/6$$

As we have stated before, we can change $U \leftarrow \mathcal{D}$ back to $\overline{U} \leftarrow \overline{\mathcal{D}}$, we get

$$\Pr_{\overline{U} \leftarrow \overline{\mathcal{D}}}\Big[Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})\Big|G'(\overline{U}) = G'(V)\Big] \geq p/6$$

In Lemma 2, let $\mathcal{S}_1, \mathcal{S}_2$ be the conditional distributions $\big\{\overline{U}\big|\overline{U} \leftarrow \overline{\mathcal{D}}, G'(\overline{U}) = G'(V)\big\}$ and $\Big\{\overline{U}\Big|G'(\overline{U}) = G'(V), \overline{U} \leftarrow \overline{\mathcal{D}'}\Big\}$. Let L in Lemma 2 be the set of views $\overline{U}$ such that $Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})$. Then, the above inequality translates to $\Pr_{\overline{U} \leftarrow \mathcal{S}_1}\big[\overline{U} \in L\big] \geq p/6$.

Then by Lemma 2

$$\begin{aligned}
&\Delta\Big(\big\{\overline{U}\big|\overline{U}\leftarrow\overline{\mathcal{D}},G'(\overline{U})=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\big\},\\
&\quad\big\{\overline{U}\big|\overline{U}\leftarrow\overline{\mathcal{D}'},G'(\overline{U})=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\big\}\Big)\\
\leq&\frac{12}{p}\Delta\Big(\big\{\overline{U}\big|\overline{U}\leftarrow\overline{\mathcal{D}},G'(\overline{U})=G'(V)\big\},\big\{\overline{U}\big|\overline{U}\leftarrow\overline{\mathcal{D}'},G'(\overline{U})=G'(V)\big\}\Big)
\end{aligned}$$

Take the expectation over $V\leftarrow\Phi$, by Equation (2) we have

$$\begin{aligned}
&\mathbb{E}_{V\leftarrow\Phi}\Big[\Delta\Big(\big\{\overline{U}\big|\overline{U}\leftarrow\overline{\mathcal{D}},G'(\overline{U})=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\big\},\\
&\qquad\big\{\overline{U}\big|\overline{U}\leftarrow\overline{\mathcal{D}'},G'(\overline{U})=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\big\}\Big)\Big]\\
&\qquad\leq\frac{12}{p}\cdot\frac{72\varepsilon}{p^2}=\frac{864\varepsilon}{p^3}
\end{aligned}$$

Note that U_A^{r-1} is a partial view of $\overline{U}(=U^{r,m+i,j-1})$, substituting $\overline{U}$ by U_A^{r-1} will not increase the statistical distance. So we have

$$\begin{aligned}
&\mathbb{E}_{V\leftarrow\Phi}\Big[\Delta\Big(\big\{U_A^{r-1}\big|\overline{U}\leftarrow\overline{\mathcal{D}},G'(\overline{U})=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\big\},\\
&\qquad\big\{U_A^{r-1}\big|\overline{U}\leftarrow\overline{\mathcal{D}'},G'(\overline{U})=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\big\}\Big)\Big]\leq\frac{864\varepsilon}{p^3}
\end{aligned}$$

In the above inequality, we argue that the the first distribution inside $\Delta(\cdot,\cdot)$ is exactly $\mathcal{S}_0(V)$. Recall the definition of $\mathcal{S}_0(V)$:

$$\mathcal{S}_0(V)=\Big\{U_A^{r-1}\Big|U\leftarrow\mathcal{D}_{G(V)},U_E^{r,m+i}=V_E^{r,m+i},Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\Big\}.$$

By the definition of $\mathcal{D}_{G(V)}$, we rewrite it as

$$\mathcal{S}_0(V)=\Big\{U_A^{r-1}\Big|U\leftarrow\mathcal{D},G(U)=G(V),U_E^{r,m+i}=V_E^{r,m+i},Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\Big\}.$$

Since $G'(V)=(G(V),V_E^{r,m+i})$, we rewrite it as

$$\mathcal{S}_0(V)=\Big\{U_A^{r-1}\Big|U\leftarrow\mathcal{D},G'(U)=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\Big\}.$$

Now we substitute $U\leftarrow\mathcal{D}$ by $\overline{U}\leftarrow\overline{\mathcal{D}}$, we get

$$\mathcal{S}_0(V)=\Big\{U_A^{r-1}\Big|\overline{U}\leftarrow\overline{\mathcal{D}},G'(U)=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\Big\}.$$

which is exactly the first distribution in $\Delta(\cdot,\cdot)$.

Now we denote the second distribution in $\Delta(\cdot,\cdot)$ as

$$\mathcal{S}_2(V)=\Big\{U_A^{r-1}\Big|\overline{U}\leftarrow\overline{\mathcal{D}'},G'(\overline{U})=G'(V),Q(U_A^{r-1})\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\Big\}.$$

So far, we have proved that

$$\mathbb{E}_{V\leftarrow\Phi}\left[\Delta\left(\mathcal{S}_0(V),\mathcal{S}_2(V)\right)\right]\leq\frac{864\varepsilon}{p^3}=\mathsf{negl}(\lambda).$$

Recall that

$$\mathcal{S}_1(V)=\left\{\mathsf{Sim}_A\leftarrow\mathsf{SIM}(M^{r-1}(V),V_E^{r,m+i})\middle|Q(\mathsf{Sim}_A)\cap Q(V_B^{r,j-1})\subseteq Q(V_E^{r,m+i})\right\}.$$

Now we only need to show the following to prove Claim 4

$$\mathbb{E}_{V\leftarrow\Phi}\left[\Delta\left(\mathcal{S}_1(V),\mathcal{S}_2(V)\right)\right]\leq\mathsf{negl}(\lambda).$$

Here we briefly explain the proof:

For any $V\leftarrow\Phi$, suppose $\mathsf{Sim}_A\in\sup(\mathcal{S}_1(V))$. Denote $U=\mathsf{Sim}_A\cup V_E^{r,m+i}\cup V_B^{r,j-1}$. Then we prove that

1. $U_{A,E},U_{B,E},U$ are valid views.
2. $U_{A,E},U_{B,E},U$ are good views and $w(U)>0$.
3. $w(U_{A,E})$ is proportional to $w(U)$.

Claim 4 follows from these three statements: since U is a valid view with $w(U)>0$, Sim_A should also be in $\sup(\mathcal{S}_2(V))$. In other words, $\sup(\mathcal{S}_1(V))\subseteq\sup(\mathcal{S}_2(V))$. The probability of Sim_A in $\mathcal{S}_1(V)$ is proportional to $w(U_{A,E})$ and the probability of Sim_A in $\mathcal{S}_2(V)$ is proportional to $w(U)$. Statement 3 implies that $\mathcal{S}_1(V)$ and $\mathcal{S}_2(V)$ has the same distribution over $\sup(\mathcal{S}_1(V))$. This means that

$$\Delta\left(\mathcal{S}_1(V),\mathcal{S}_2(V)\right)=\Pr\left[U_A^{r-1}\notin\mathcal{S}_1(V)\middle|U_A^{r-1}\leftarrow\mathcal{S}_2(V)\right]$$

If $U_A^{r-1}\in\sup(\mathcal{S}_2(V))$ and $U_A^{r-1}\notin\mathcal{S}_1(V)$, the only possibility is that $U_A^{r-1}\cup V_E^{r,m+i}$ contain collisions. This causes U_A^{r-1} not being able to be simulated by Eve. Now consider the elements in $\sup(\mathcal{S}_0(V))$. If $U_A^{r-1}\in\sup(\mathcal{S}_0(V))$, $U_A^{r-1}\cup V_E^{r,m+i}$ must not contain collisions. Thus, we have

$$\Pr\left[U_A^{r-1}\notin\mathcal{S}_1(V)\middle|U_A^{r-1}\leftarrow\mathcal{S}_2(V)\right]\leq\Delta\left(\mathcal{S}_0(V),\mathcal{S}_2(V)\right).$$

This means,

$$\Delta(\mathcal{S}_1(V),\mathcal{S}_2(V))\leq\Delta(\mathcal{S}_0(V),\mathcal{S}_2(V)).$$

Taking the expectation over $V\leftarrow\Phi$, we have

$$\mathbb{E}_{V\leftarrow\Phi}\left[\Delta(\mathcal{S}_1(V),\mathcal{S}_2(V))\right]\leq\mathbb{E}_{V\leftarrow\Phi}\left[\Delta(\mathcal{S}_0(V),\mathcal{S}_2(V))\right]\leq\mathsf{negl}(\lambda).$$

In the following, we formally prove these three statements.

Statement 1: $U_{A,E}, U_{B,E}, U$ are valid views Since $\mathsf{Sim}_A \in \sup(\mathcal{S}_1(V))$, $Q(U_A) \cap Q(U_B) \subseteq Q(U_E)$. Because of Eve's algorithm, $U_{A,E}$ is a collision-free valid view and $w(U_{A,E}) > 0$. We are going to prove the same result for U and $U_{B,E}$. Clearly, because V is sampled from Φ, $V \in \sup(\mathcal{D})$. This means that $U_{B,E} = V_E^{r,m+i} \cup V_B^{r,j-1}$ is a partial view of V, thus $U_{B,E}$ is valid and collision-free.

We now prove that U is also a valid view. Suppose U is not valid, U must contain complicated collisions mentioned in Section 6. We first show that self-collisions do not occur in U. Suppose $k[c_1] = k[c_2] = k'$ in U. Since $U_{A,E}$ and U_B are both collision-free views, the constrained relations $k[c_1] = k'$ and $k[c_2] = k'$ must be found on the opposite side. Suppose $k[c_1] = k'$ is in $U_{A,E}$ and $k[c_2] = k'$ is in U_B. This means, $k, k' \in Q(U_{A,E}) \cap Q(U_B) \subseteq Q(U_E)$. Then, Eve would have made the query $\mathsf{R}(k, k')$. The answer would be c_2 because it must be consistent with U_B. A contradiction because $k[c_1] = k'$ is found in $U_{A,E}$.

Before eliminating the other 3 type of complicated collisions, we first prove an intermediate result. Suppose $k_1[c_1] = k_2[c_2] = k', k_1 \neq k_2$ is a collision in U. Similarly, since $U_{A,E}$ and U_B are both collision-free views, the constrained relations $k_1[c_1] = k'$ and $k_2[c_2] = k'$ must be found on the opposite side. Without loss of generality, suppose $k_2[c_1] = k'$ is in $U_{A,E}$ and $k_2[c_2] = k'$ is in U_B. Then, $k' \in Q(U_{A,E}) \cap Q(U_B) \subseteq Q(U_E)$. Now, we argue that k' must be an isolated constrained key in U_E. If it is not an isolated constrained key in U_E, since $U_{B,E}$ is collision free, $k' = k_2[c_2]$ is in U_E. Similarly, since $U_{A,E}$ is collision free, $k' = k_1[c_1]$ in U_E. A contradiction. To sum up, we have prove that, if a collision happens at the constrained key k' in U, then k' must be an isolated constrained key in U_E.

With this result, we can easily eliminate the other 3 types of complicated collisions:

- Suppose a "W" collision exists in U, that is $k_1[c_1] = k_2[c_2] = k'_1, k_1[c'_1] = k_3[c_3] = k'_2$. Then, k'_1, k'_2 both appear in U_E as isolated constrained keys. This means that Eve would have queried $\mathsf{R}(k'_1, k'_2)$ and the answer is $\perp$. Then, $k_1[c_1] = k'_1$ and $k_1[c'_1] = k'_2$ cannot be both found in U_A because this is inconsistent with $\mathsf{R}(k'_1, k'_2) = \perp$. Similarly, they cannot be both found in U_B, since U_B must also be consistent with U_E. Without loss of generality, we assume $k_1[c_1] = k'_1$ in U_A and $k_1[c'_1] = k'_2$ in U_B. This means $k_1 \in Q(U_A) \cap Q(U_B) \subseteq Q(U_E)$. Eve must have made the query $\mathsf{R}(k_1, k'_2)$ and the answer would be c'_1, this contradicts to the fact that k'_1 is an isolated constrained key in U_E.
- Suppose a 2-to-2 collision exists in U: $k_1[c_1] = k_2[c_2] = k'_1, k_1[c'_1] = k_2[c'_2] = k'_2$. Then, k'_1 and k'_2 must all appear in U_E as isolated constrained keys. Then, $k_1[c_1] = k'_1$ and $k_1[c'_1] = k'_2$ cannot be both found in U_A because this is inconsistent with Eve's view. Similarly, they cannot be both found in U_B, since U_B must also be consistent with U_E. Without loss of generality, we assume $k_1[c_1] = k'_1$ in U_A and $k_1[c'_1] = k'_2$ in U_B. This means $k_1 \in Q(U_A) \cap Q(U_B) \subseteq Q(U_E)$. Eve must have made the query $\mathsf{R}(k_1, k'_2)$ and the answer would be c'_1, this contradicts to the fact that k'_1 is an isolated constrained key in U_E.

– Suppose 3 master keys collide at one constrained key in U: $k_1[c_1] = k_2[c_2] = k_3[c_3] = k'$. Then, at least one of $U_{A,E}$ and U_B contains a collision. However, we have proved that both of them are collision-free. A contradiction.

Thus, we have shown that there are no complicated collisions in U, which means that U is a valid view.

Statement 2: $U_{A,E}, U_{B,E}, U$ are good views Before proving statement 2, we first show more results about hybrid groups in U. Suppose $k_0[c_0] = k_1[c'_0] = k'$ is a collision in U. $CK_0 = \{k'_{0,1} = k_0[c_1], \cdots, k'_{0,s} = k_0[c_s]\}$ is the set of constrained keys generated from k_0 other than k'. $CK_1 = \{k'_{1,1} = k_1[c'_1], \cdots, k'_{1,t} = k_0[c'_t]\}$ is the set of constrained keys generated from k_1 other than k'. In the proof of statement 1, we have shown that k' must appear in U_E as an isolated constrained key. Note that any other keys in this hybrid group (including k_0, k_1, CK_0, CK_1) all have hidden relations with k', they cannot appear in U_E. Also, each of them cannot appear in both U_A and U_B, because $Q(U_A) \cup Q(U_B) \subseteq Q(U_E)$. In other words, each of these keys appears in exactly one of U_A and U_B. Since both $U_{A,E}$ and $U_{B,E}$ are collision free, the two relations $k_0[c_0] = k'$ and $k_1[c'_0] = k'$ must be on the opposite side. Without loss of generality, assume $k_0[c_0] = k'$ is in U_A and $k_1[c'_0] = k'$ is in U_B. Then, k_0 will not appear in U_B and k_1 will not appear in U_A. This means, the relations $k_0[c_i] = k'_{0,i}, i \in [s]$ must appear in U_A and $k_1[c_j] = k'_{0,j}, j \in [t]$ must appear in U_B. Thus, k_0, CK_0 only appears in U_A; k_1, CK_1 only appears in U_B.

With this result, we first prove that U is a good view. If U is not a good view, then there exists 2 F or f queries in U whose equivalency relation cannot be determined. We first consider the case in which the two keys of these two queries are in the same normal group in U. Then, there are only two possibilities:

1. $k \in K, k' \in K'$ both appeared in U and it can be observed in U that $k' = k[c]$ for some $c \in C$. Both $\mathsf{F}(k, x)$ and $\mathsf{f}(k', x)$ are queried in U. But the H queries made in U are not enough to evaluate $c(x)$.
2. $k \in K, k'_1, k'_2 \in K'$ both appeared in U and it can be observed in U that $k'_1 = k[c_1], k'_2 = k[c_2]$ for some c_1, c_2. Both $\mathsf{f}(k'_1, x)$ and $\mathsf{f}(k'_2, x)$ are queried in U. But the H queries made in V are not enough to evaluate $c_1(x)$.

Here, we only show the contradiction in the first case. The second case follows the same logic. First, if $\mathsf{F}(k, x)$ and $\mathsf{f}(k', x)$ both belong to U_A, one of them must be made after the other. Before querying the later, because Alice's algorithm is under restriction 4 of the normal form, Alice must make sure that there are no hidden relation between k and k'. This means, the relation $k[c] = k'$ would be found by Alice. Thus c appears in U_A before the later query. Because Alice's algorithm is also under restriction 2 of the normal form, before querying the later, Alice must make enough H queries to evaluate $c(x)$. This contradicts to the fact that $c(x)$ is not evaluable in U. Therefore, $\mathsf{F}(k, x)$ and $\mathsf{f}(k', x)$ cannot be both in U_A. Similarly, they cannot be both in U_B. Thus, either one of them is in U_E, or one of them is in U_A and the other is in U_B. In the later case, x is an intersection element. Thus, in either cases, $x \in U_E$. Then, because U_E

is complete, $c \notin U_E$. This means k, k' does not both appear in U_E. Without loss of generality, we assume $k \notin U_E$. Then $k[c] = k', \mathsf{F}(k,x)$ are not in U_E. Now consider the locations of $k[c] = k'$ and $\mathsf{F}(k,x)$. Because U_A is complete, they cannot be both in U_A, otherwise both c and x are in U_A, meaning that $c(x)$ is evaluable under U_A, a contradiction. Also, because $k \notin U_E$, k is not an intersection element. Thus, the only possibility is that both $k[c] = k', \mathsf{F}(k,x)$ are both in U_B. Since $c(x)$ is not evaluable under U_B, the only possibility is that $\mathsf{F}(k,x)$ happens in U_B before round r, c does not appear in U_B before round r, the first query of round r is an R query and the answer contains c. In this case, Charlie would send c to Eve in round r. Note that $U_B = V_B^{r,j-1}$ and $U_E = V_E^{r,m+i}$ are both partial views of the same view V, U_E should contain c. A contradiction.

We now consider the case that the two keys in the 2 F and f queries does not belong to the same normal group, denote them as query_1 and query_2. Because the equivalency relation between the two queries cannot be determined, these two keys must be in the same hybrid group, and they are on the opposite side of the colliding constrained key k'. Thus, one of these two queries is in U_A and the other is in U_B. This means that the second input of these two queries, denoted as x, is an intersection element. Thus, $x \in U_E$. Also, we have proved that $k' \in U_E$, so Eve must have made the query $\mathsf{f}(k', x)$. Then, either the equivalency relation between query_1 and $\mathsf{f}(k', x)$ cannot be determined, or the equivalency relation between query_2 and $\mathsf{f}(k', x)$ cannot be determined. Either cases reduce to a case that we have previously proved.

In the above proof, we have only used following fact about $U_A = \mathsf{Sim}_A$: (1), U_A is complete; (2), $U_{A,E}$ is a collision-free valid view; (3), the algorithm of Alice is under restriction 2,3,4; (4), $Q(U_A) \cap Q(U_B) \subseteq Q(U_E)$. Now, notice that an empty view (which does not contain any query answer pairs) also satisfies these conditions, we can substitute Sim_A by an empty view and show that $U_{B,E}$ is a good view.

On the other hand, since U_A and U_E are both complete views and the algorithm of Alice and Eve are under restriction 2,3,4, and $U_{A,E}$ are collision-free because of Eve's sampling algorithm, $U_{A,E}$ is a good view by Lemma 4.

Statement 3: $w(U)$ and $w(U_{A,E})$ are proportional We show that $w(U)/w(U_{A,E}) = w(U_{B,E})/w(U_E)$. Note that, the right side does not depend on $U_A = \mathsf{Sim}_A$, this is enough to show that $w(U)$ and $w(U_{A,E})$ are proportional.

We prove this in the following way: suppose U_A is initially empty. Then the initial value of $w(U)/w(U_{A,E})$ is exactly $w(U_{B,E})/w(U_E)$. We iteratively add the query-answer pairs in U_A back. Each time, we add one of the following 4 types: $\mathsf{H}(z) = b, \mathsf{Keygen}(k,c) = k', \mathsf{F}(k,x) = y, \mathsf{f}(k',x) = y$. We first add all the H queries, then Keygen queries, and finally F, f queries. An R query with non-empty response can be broken into 1 or 2 Keygen queries. Since we have proved that $U_{A,E}, U$ are good views, and we first add H queries in this expanding process, $U_{A,E}$ and U must remain good in this expanding process.

We argue that, each time U_A is expanded, the ratio between $w(U)$ and $w(U_{A,E})$ remains the same. Since U and $U_{A,E}$ remains good in the expanding process, their weights can be computed in the easier way.

If a query-answer pair is added to U_A but the answer can be deduced from $U_{A,E}$, then it can also be deduced from U. This means that the value of $w(U_{A,E})$ and $w(U)$ does not change. Naturally, the ratio between $w(U_{A,E})$ and $w(U)$ does not change.

Suppose a query-answer pair $\mathsf{H}(z) = b$ is going to be added to U_A, and it is not in $U_{A,E}$ so far. Because U_E contains all the intersection elements, the query answer pair $\mathsf{H}(z) = b$ is also not in U_B. Thus, both $w(U)$ and $w(U_{A,E})$ are multiplied by $1/2$, their ratio does not change.

Suppose a query-answer pair $\mathsf{Keygen}_\kappa(k, c) = k'$ is going to be added to U_A and $k[c] = k'$ cannot be deduced from $U_{A,E}$ so far. Then, we cannot deduce that $k[c] = k'$ from U_B. (Otherwise, k, c, k' are all intersection elements, thus contained in U_E. Therefore, we can deduce $k[c] = k'$ from U_E, a contradiction.) Then, both $w(U)$ and $w(U_{A,E})$ are multiplied by $|K'_\kappa|^{-1}$, which means that the ratio does not change.

Suppose an F_κ or f_κ query-answer pair $\mathsf{Query}_0 = y$ is going to be added to U_A and the answer cannot be deduced from $U_{A,E}$ so far. The first input of Query_0 is denoted as key, which can be either a master key or a constrained key. The second input of Query_0 is denoted as $x \in X_\kappa$. Then, the value of $w(U_{A,E})$ is multiplied by $|Y_\kappa|^{-1}$. If the value of $w(U)$ is also multiplied by $|Y_\kappa|^{-1}$, then the ratio of $w(U_{A,E})$ and $w(U)$ does not change. If not, the only possibilities are (1), $w(U)$ does not change; (2), $w(U)$ becomes 0. In both cases, another F or f query is made in U and this query is equivalent to Query_0 under U. Denote this query as Query_1. Then, the second input of Query_1 is naturally x. Denote the first input of Query_1 as key_*, which can be a master key or a constrained key as well. We now consider whether x is in U_E.

- **Case 1:** $x \notin U_E$. Since $x \in U_A$, we have $x \notin U_B$. Then, Query_1 can only be in U_A. Since Query_0 and Query_1 are both made in U_A, and they are equivalent in U, key and key_* are in the same normal group. Because of restriction 4 of the normal form, the hidden relation between key and key_* is also clear in U_A. Also because U_A is complete, Query_0 and Query_1 are equivalent under U_A. This contradicts to the fact that the answer of Query_0 cannot be determined by $U_{A,E}$.
- **Case 2:** $x \in U_E$. Since Query_0 and Query_1 are equivalent, key and key_* must be linked by constrained relations in U. Denote the linked path as $key = key_0 \overset{c_1}{-} key_1 \overset{c_2}{-} \cdots \overset{c_{t-1}}{-} key_{t-1} \overset{c_t}{-} key_t = key_*$. Each edge in this path $key_{i-1} \overset{c_i}{-} key_i$ represents a constrained relation $k[c_i] = k'$, here $\{k, k'\} = \{key_{i-1}, key_i\}$.(For example, $k'_6 \overset{c_4}{-} k_2 \overset{c_6}{-} k'_8 \overset{c_7}{-} k_3 \overset{c_8}{-} k'_9$ in Figure 3).
  First, since $x \in U_E$, but the answer of Query_0 cannot be deduced from $U_{A,E}$ so far, $key = key_0$ is not in U_E, otherwise Query_0 is also made in U_E. Then, $key_0 \notin U_B$, otherwise key_0 would be an intersection element. This means the constrained relation between key_0 and key_1 is also in U_A. Because U_A is complete, $c_1(x)$ can be evaluate to be 1 in U_A. Thus, $key_1 \notin U_E$. (Otherwise, since key_1 and x are both in U_E, Eve must have made the query $\mathsf{F}(key_1, x)$ or $\mathsf{f}(key_1, x)$. Combine with the fact that $c_1(x)$ can be evaluated to be 1 in

U_A, the answer of Query_0 can be deduced from $U_{A,E}$ so far, a contradiction.) Using the same argument, we can prove that $key_2, \cdots, key_t = key_*$ do not belong to $U_{B,E}$. The constrained relation $key_{i-1} \overset{c_i}{-} key_i$ appears in U_A for $i = 1, \cdots, t$. $c_1(x), c_2(x), \cdots, c_t(x)$ can be all evaluated to be 1. Since $key_* \notin U_{B,E}$, Query_1 can only be in U_A. Combine with all the results we have proved, Query_0 and Query_1 are equivalent under U_A. This means the answer of Query_0 can be deduced from $U_{A,E}$ so far. A contradiction. □

Claim 4 implies the following claim:

Claim 5 *Fix a segment number $m+i$, denote*

$$X_i(V) = \Pr_{U_A^{r-1} \leftarrow \mathcal{S}_0(V)} \left[q_{r,j}(V) \in Q(U_A^{r-1})\right]$$
$$= \Pr_{U \leftarrow \mathcal{D}_{G(V)}} \left[q_{r,j}(V) \in Q(U_A^{r-1}) \middle| U_E^{r,m+i} = V_E^{r,m+i}, Q(U_A^{r-1}) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})\right]$$

and

$$Y_i(V) = \Pr_{\mathsf{Sim}_A \leftarrow \mathcal{S}_1(V)} \left[q_{r,j}(V) \in Q(\mathsf{Sim}_A)\right]$$
$$= \Pr_{\mathsf{Sim}_A \leftarrow \mathsf{SIM}(M^r(V), V_E^{r,m+i})} \left[q_{r,j}(V) \in Q(\mathsf{Sim}_A) \middle| Q(\mathsf{Sim}_A) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})\right].$$

then we have

$$\mathbb{E}_{V \leftarrow \Phi} \left[|X_i(V) - Y_i(V)|\right] = \mathsf{negl}(\lambda).$$

Proof. If V is fixed, $q_{r,j}(V)$ is also fixed. Then, whether $q_{r,j}(V) \in Q(U_A^{r-1})$ can be seen as a binary function over U_A^{r-1}. Thus, we have

$$|X_i(V) - Y_i(V)| \leq \Delta(\mathcal{S}_0(V), \mathcal{S}_1(V))$$

Take the expectations over $V \leftarrow \Phi$, we prove Claim 5 from Claim 4. □

Proof of Claim 3 Now we are ready to prove Claim 3:

First consider the maximum size of $Q(V_B)$. In each round, at most 1 input of the first query is new, and the output can contain at most 3 new elements. Thus, the first query can bring at most 4 new elements to $Q(V_B)$. The following $2n\kappa_{\max}$ H queries can bring at most $2n\kappa_{\max}$ new elements to $Q(V_B)$. There are n rounds, so the maximum size of $Q(V_B)$ is $2n^2\kappa_{\max}+4n$. Denote $N = 2n^2\kappa_{\max}+4n$ which is also polynomial over λ. Naturally, since $V_B^{r,j-1}$ is a partial view of V_B, $|Q(V_B^{r,j-1})| \leq N$.

In round r, whenever Eve samples a view of Alice Sim_A s.t. $Q(\mathsf{Sim}_A) \cap Q(V_B^{r,j-1}) \not\subseteq Q(V_E^{r,m+i})$, (in other words, this simulation is a "hit") Eve knows at least 1 private element of Bob. Thus, among the m simulations, at least $m-N$ of them satisfies $Q(\mathsf{Sim}_A) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})$.

We first temporarily fix a normal G, then $Q(V_B^{r,j-1})$ is fixed for any $V \leftarrow \mathcal{D}_G$. We can think of a $V \leftarrow \mathcal{D}_G$ generated, segment by segment, as follows: In each segment, let b be the probability that a random $\mathsf{Sim}_A \leftarrow \mathsf{SIM}(M^r, V_E^{r,m+i})$ has no intersection with Bob's private elements, i.e. $Q(\mathsf{Sim}_A) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})$, in short words, the simulation is a "miss". Flip a coin which head comes up with probability b. If head, pick $\mathsf{Sim}_A \leftarrow \mathsf{SIM}(M^r, V_E^{r,m+i})$ conditioned on "miss", i.e. $Q(\mathsf{Sim}_A) \cap Q(V_B^{r,j-1}) \subseteq Q(V_E^{r,m+i})$. If tails, pick $\mathsf{Sim}_A \leftarrow \mathsf{SIM}(M^r, V_E^{r,m+i})$ conditioned on "hit", i.e. $Q(\mathsf{Sim}_A) \cap Q(V_B^{r,j-1}) \not\subseteq Q(V_E^{r,m+i})$. Note that this process does not generate the whole view V, but this is enough for our purpose.

Now, we view this process as a probability tree, denoted as $T(G)$. The depth of the tree is $2m$, each segment corresponds to 2 layers of this tree: one layer of coin flipping, and another layer of the rest (sampling Sim_A and the update phase). Then, the edges on odd layers correspond to coin flipping and th edges of even layers correspond to the rest. On layer $2i-1$, each edge represents one possible outcome of the coin flipping, i.e. head or tail. On layer $2i$, each edge represents one possible outcome of the simulation phase and the update phase. Therefore, each nodes of depth $2i$ represents one possible partial view $V_E^{r,m+i}$. The weight of the root is 1. The weight of an edge is defined as the probability of the event represented by this edge. The weight of a node d is defined as the product of the weight of d's father d' and the weight of edge (d', d).

We say an edge from an even layer is black if $q_{r,j}$ is in Sim_A of the simulation phase represented by this edge. Otherwise, we say this edge is white (color of edges is only defined for even layer edges). Note that $q_{r,j}$ may be found in the segment even if the corresponding edge in the tree is white, because Eve might find $q_{r,j}$ in the update phase.

We say a node d of odd depth (in other words, the nodes after coin flipping) is black if its father edge is head and the sum of weights of black edges among all the child edges of d is greater than $p/7$. We say a node d on of odd depth is white if its father edge is head and the proportion of black edges among all the child edges of d is less than or equal to $p/7$ (color of nodes is only defined for nodes whose father edges are heads).

All edges and nodes that are not defined to be black or white are considered colorless.

We say a leaf node is "residual" if, on the path from the root to this leaf, there are no black edges or white nodes. We first argue that the sum of the weights of residual leaves is small. We delete all black edges and white nodes in this tree, and when we delete an edge or node, we also delete the sub-tree that is connected to this edge or node. Then, we recursively delete all leaves that are not of depth $2m$ which is the depth of the original tree. Observe that residual leaves are never deleted from the tree and all leaves left are residual.

Consider a path from top to bottom in the resulting tree. Note that there are at least $m - N$ nodes of odd depth whose father edges are head. Then, all of these nodes are black because white nodes were all deleted. All the child edges of a black node are white since black edges were all deleted. By the definition

of black nodes, the sum of weights of edges from a black node is smaller than $1 - p/7$. To sum up, the resulting tree has the following properties:

- Each path from top to bottom has at least $m - N$ black nodes.
- The sum of weights of child edges from a black node is smaller than $1 - p/7$.
- The sum of weights of non-child edges from a (possibly non-black) node is smaller than 1.

From the above conditions, we can deduce that the sum of the weights of all leaves in the resulting tree is smaller than $(1 - p/7)^{m-N}$. In order to bound it, we set m to be the minimum integer such that $(1 - p/7)^{m-N} < p/7$, then $m = \lceil N + \log(p/7)/\log(1 - p/7) \rceil$ is polynomial over λ.

Let $X_i(V)$ and $Y_i(V)$ be the same as in Claim 5. Claim 5 states that, for any i,

$$\mathbb{E}_{V \leftarrow \Phi}\left[|X_i(V) - Y_i(V)|\right] = \mathsf{negl}(\lambda).$$

Applying Markov's inequality, we obtain

$$\forall i, \Pr_{V \leftarrow \Phi}\left[|X_i(V) - Y_i(V)| \geq p/42\right] \leq \frac{42}{p}\mathsf{negl}(\lambda) = \mathsf{negl}(\lambda). \tag{6}$$

Then apply a union bound, we obtain

$$\Pr_{V \leftarrow \Phi}\left[\exists i, |X_i(V) - Y_i(V)| \geq p/42\right] \leq m\mathsf{negl}(\lambda) = \mathsf{negl}(\lambda). \tag{7}$$

For a fixed G, consider V such that $V \in P_2(G)$, we have $X_i(V) \geq p/6$ by Equation (4). If $V \not\models \mathsf{Condition3}$, there are two possible cases:

1. There exists some i s.t. $Y_i(V) \leq p/7$.
2. For all i, $Y_i(V) > p/7$.

In the first case, $|X_i(V) - Y_i(V)| \geq p/6 - p/7 = p/42$ for some i. In the second case, suppose the leaf v in $T(G(V))$ corresponds to $V_E^{r,2m}$. Consider the path from the root of $T(G)$ to v. Since $V \not\models \mathsf{Condition3}$, there are no black edges on this path. Since $Y_i(V) > p/7$ for all i, there are no white nodes on this path. Therefore, v is residual.

Hence for any normal G, we have

$$\begin{aligned}
&\Pr_{V \leftarrow \mathcal{D}_G}\left[V \in P_2(G), V \not\models \mathsf{Condition3}\right] \\
\leq &\Pr_{V \leftarrow \mathcal{D}_G}\left[V \text{ is a residue leaf in } T(G)\right] \\
&+ \Pr_{V \leftarrow \mathcal{D}_G}\left[V \in P_2(G), V \not\models \mathsf{Condition3}, \exists i, |X_i(V) - Y_i(V)| \geq p/42\right].
\end{aligned} \tag{8}$$

We have proved that the first term is bounded by $p/7$, and we can also bound the second term:

$$\begin{aligned}
&\Pr_{V\leftarrow\mathcal{D}_G}[V\in P_2(G), V\not\models \mathsf{Condition3}, \exists i, |X_i(V)-Y_i(V)|\geq p/42]\\
\leq &\Pr_{V\leftarrow\mathcal{D}_G}[V\in P_2(G), \exists i, |X_i(V)-Y_i(V)|\geq p/42]\\
\leq &\Pr_{V\leftarrow\mathcal{D}_G}[\exists i, |X_i(V)-Y_i(V)|\geq p/42 | V\in P_2(G)]\\
= &\Pr_{V\leftarrow P_2(G)}[\exists i, |X_i(V)-Y_i(V)|\geq p/42].
\end{aligned}$$

Take the expectation over $G\leftarrow\mathcal{G}_{\text{normal}}$, the above inequality implies

$$\begin{aligned}
&\mathbb{E}_{G\leftarrow\mathcal{G}_{\text{normal}}}[\Pr_{V\leftarrow\mathcal{D}_G}[V\in P_2(G), V\not\models \mathsf{Condition3}, \exists i, |X_i(V)-Y_i(V)|\geq p/42]]\\
\leq &\mathbb{E}_{G\leftarrow\mathcal{G}_{\text{normal}}}[\Pr_{V\leftarrow P_2(G)}[\exists i, |X_i(V)-Y_i(V)|\geq p/42]]\\
= &\Pr_{V\leftarrow\Phi}[\exists i, |X_i(V)-Y_i(V)|\geq p/42]\\
= &\mathsf{negl}(\lambda)
\end{aligned}$$

where the last equation is exactly Equation (7).

In Equation (8), take the expectation over $G\leftarrow\mathcal{G}_{\text{normal}}$, we have

$$\begin{aligned}
&\mathbb{E}_{G\leftarrow\mathcal{G}_{\text{normal}}}\left[\Pr_{V\leftarrow\mathcal{D}_G}[V\in P_2(G), V\not\models \mathsf{Condition3}]\right]\\
\leq &p/7+\mathsf{negl}(\lambda)\\
< &p/6
\end{aligned}$$

for sufficiently large λ. Thus, Claim 3 is proved.

C.2 The difference between the proofs of Lemma 6 and Lemma 7

Lemma 6 and Lemma 7 are almost the same: Lemma 7 states that the last m segments of Eve in round r can bound the probability that $q_{r,j}$ $(j\geq 2)$ is a type 1 first failure; Lemma 6 states that the first m segments of Eve in round r can bound the probability that $q_{r,1}$ is a type 1 first failure. Thus, we can see that the proof technique can be easily applied to prove Lemma 6. Actually, the proof of Lemma 6 is simpler. Recall that, in the proof of Lemma 7, the view of Bob is $V_B^{r,j-1}$ which is not a complete view. Analogously, Bob's view in the proof of Lemma 6 would be $V_B^{r,0}$ which is the view of Bob right before the first query of round r. Unlike $V_B^{r,j-1}$, $V_B^{r,0}$ is a complete view. In the proof of Lemma 7, the issue that Bob's view not being complete is addressed by introducing Charlie. However, in the proof of Lemma 6, we have no such issue, since Bob's view is always complete.