Private Information Retrieval Does Not Imply One-way Permutations

Hsiao Chun-Yun
Department of Computer Science and Information Engineering
National Taiwan University

To my companions,

those whom I love and those who love me.

Acknowledgements.

I would like to thank Professor Yuh-Dauh Lyuu for bringing me into the fascinating field - theoretical computer science. In addition to the fundamental complexity theory, he also introduced me several exciting topics such as randomization, parallel computing and game theory. Most important of all, I learned from him the spirit of seeking knowledge.

I would like to thank Professor Chi-Jen Lu. He, patiently, tolerated my laziness and showed me the active attitude of doing research. If I am able to look into problems any deeper than before, or if I am able to appreciate the enjoyment of research, it is all due to his invaluable guidance.

As far as this thesis is concerned, I owe Professor Lu and Yan-Cheng Chang everything. Chang kindly shared all his ingenious ideas with me. And without Professor. Lu's help, the proofs in this work would be nowhere close to being rigorous. I also would like to thank Professor Yuh-Dauh Lyuu and Professor Tsan-Sheng Hsu for many helpful comments.

Special thanks to many many friends that, during the writing of this work, gave me support.

Finally, I am extremely grateful to my parents for providing such a wonderful environment to study in during the past years.

Abstract

We study the relationship between the protocol *Private Information Retrieval* and the primitive *One-way Permutations*. As shown in [20] that the existence of one-way trapdoor permutation implies the existence of private information retrieval, we follow the methodology suggested in [14] and provide strong evidence that the converse is not true. Namely, the existence of private information retrieval is not likely to be a sufficient condition for the existence of one-way permutations, hence not a sufficient condition for the existence of one-way trapdoor permutations.

Contents

Chapter 1

Introduction

Modern Cryptography

Cryptography was of interest long before the arrival of computer, and was considered as an exclusive domain of the military for many years. However, public academic research during the past 30 years has transformed this secret art from a semi-scientific discipline to a respectable field in theoretical computer science. The notion of security is now well defined in terms of *information theory* and of *complexity theory*. Modern cryptography focuses mostly on the latter. That is, the security of a protocol is based on the *infeasibility*, rather than the *impossibility*, of extracting secrets. More precisely, the adversary is assumed to have only limited resources rather than being all-powereful in a reasonable sense.

Despite enormous efforts on the study in computational complexity, computer scientists have been frustrated in finding out many of the relationships among complexity classes. One of the fundamental relationship that is unknown, unfortunately, is the relationship between $\mathcal{P}$ and $\mathcal{NP}$. Therefore in modern cryptography, most theorems are unavoidably based on unproven assumptions such as the existence of one-way functions [13], which in particular implies $\mathcal{P} \neq \mathcal{NP}$.

Similarly, finding out the relationships among primitives is one of the major issues of the study in modern cryptography. This is done with much success in the last few decades, that most primitives fall into two categories: either one-way functions are sufficient and necessary, or stronger properties like trapdoor and/or injective are required.[1] For example, pseudorandom generators [12], signature schemes [24, 26], commitment schems [12, 23] and zero-knowledge proofs for $\mathcal{NP}$ [11, 25] are all shown to exist if and only if one-way functions do. In the somewhat more versatile category including public key encryption

[1] We sometimes refer to the former as "private cryptography" and to the latter as "public cryptography."

[9, 2, 7], oblivious transfer, secure function evaluation [27, 17, 3, 7], key agreement [6, 14, 29], and trapdoor permutations [30, 20], however, primitives are not all equivalent to each other. Some of them are even *incomparable* (e.g. PKE and OT [7]), and there is no simple hierarchy of assumptions in this world.

A standard way to prove that the existence of a primitive Q implies the existence of a primitive P is to find a reduction from P to Q. However, to prove that the existence of primitive Q is not a sufficient condition for the existence of primitive P is not as straightforward, especially when both primitives are commonly believed to exist; a reduction itself can just ignore the source primitive Q and build the target primitive P from scratch. While it is indeed very difficult to prove the failure of *all* reductions, Impagliazzo and Rudich [14] gave a method for separating primitves under a restricted but important subclass of reductions, namely, *black-box* reductions. Informally, a black-box reduction from primitive P to primitive Q is a construction of P out of Q that views Q as a black-box (a subroutine), rather than using the internal structure of the implementation of Q. More formally, the reduction is to construct an *oracle Turing machine* that, given *oracle access* to an implementation of Q, implements P. A black-box reduction is also known as a *relativizing* reduction because the reduction *relativizes*. That is, if there exists a black-box reduction from primitive P to primitive Q, then relative to *any* oracle the existence of Q implies the existence of P.[2] Therefore, in order to show that there are no black-box reductions from P to Q, it suffices to find an oracle relative to which Q exists but P does not. Ruling out the possibility of black-box reductions from primitive P to primitive Q is a very strong evidence that the existence of Q is not a sufficient condition for the existence of P, in the sense that almost all known implications among primitives are proved in a black-box way (hold relative to any oracle), with [10][3] being one of the few exceptions. Thus, it is quite "safe" to say that "primitive Q does not imply primitve P" if we can show that there are no black-box reductions from P to Q. Using this powerful methodology (the oracle separation paradigm), Impagliazzo and Rudich showed that one-way permutations do not imply key agreement (under black-box reduction). And in [7], Gertner et al. showed a two-sided separation of PKE from OT. This result is rather surprising and inspired us for further study.

[2] SeeLemma 2.1 for more detail.

[3] In [10], the proof of "the existence of one-way permutations implies the existence of zero-knowledge protocols for all languages in $\mathcal{NP}$" does not relativize.

Our work

Although the positions of almost all the important primitives are resolved, the relationships between the more recently introduced protocol *Private Information Retrieval* [4] and other primitives are not well understood. Informally, a PIR protocol allows a user to retrieve information from a database while maintaining the query private from the database manager. The strongest sufficient (resp. necessary) condition for the existence of PIR known so far, is the existence of one-way trapdoor permutations [20] (resp. the existence of OT [5]). In this work we follow the oracle separation paradigm and show that PIR does not imply one-way permutations under black-box reduction. Specifically, we construct an oracle that is "almost" random and relative to which PIR exists, based on the result Rudich showed in [28] that one-way permutations do not exist relative to an oracle consisting of a random function and a $\mathcal{PSPACE}$-complete oracle, we then show that one-way permutations do not exist relative to our oracle union a $\mathcal{PSAPCE}$-complete one.

As pointed out in [14], non-black-box (non-relativizing) reductions from one-way permutations to PIR, though believed to be difficult to find, are still possible. Nevertheless, we hope that providing such a strong evidence that "PIR does not imply one-way permutations" will help to better understand the structure of PIR and it's cryptographic significance.

Outline

We give notation and definitions as well as some basic probability theorems in Chapter 2. A brief introduction to the oracle separation paradigm is also included. And the main result "private information retrieval does not imply onw-way permutations" is proved in Chapter 3.

Chapter 2

Preliminaries

2.1 Notation and Definitions

In this section we give formal definitions of the primitive *One-way Permutations* and the protocol *Private Information Retrieval*, as well as other notation and conventions.

We abbreviate probabilistic polynomial time Turing machine with the notation PPTM, and use U_n to denote the random variable uniformly distributed over the set of n-bit strings. Namely, $\Pr[U_n = u]$ equals 2^{-n} if $u \in \{0,1\}^n$ and equals 0 otherwise. And use k to denote the *security parameter* of cryptographic primitives and protocols.[1] Let x be a bit string, we use $x[i]$ to denote the i^{th} bit of x.

Security

Intuitively, a primitive or a protocol is *secure* if the "secret" can not be efficiently computed. An efficient computation is one which can be carried out by a PPTM, thus by secure we mean that every adversary PPTM outputs the secret successfully with only a negligible probability. Formally, we call a function *negligible* if it vanishes faster than the inverse of any polynomial. Namely,

Definition 1 (Negligible Functions). *A function* $\nu : \mathbb{N} \to \mathbb{R}$ *is* negligible *if for every polynomial* $p(\cdot)$ *there exists an integer* n_0 *such that for all* $n > n_0$

$$\nu(n) < \frac{1}{p(n)}$$

[1] In general, the input length and the running time of all PPTMs involved when discussing a primitive or a protocol, will be polynomial in k. However, in a PIR protocol, the input length will be polynomial in $k + |db|$, where db is the database.

For example, the function 2^{-n} is negligible. Note that a function remains negligible when multiplied by any fixed polynomial. It follows that an event which occurs with negligible probability is highly unlikely to occur even if we repeat the experiment polynomially many times.

One-way Permutations (OWPs)

A function is called *one-way* if it is easy to compute but hard to invert, and is called a *permutation* if it is injective (one-to-one) and onto. Formally,

Definition 2 (One-way Permutations). *A function* $f : \{0,1\}^* \to \{0,1\}^*$ *is called a* one-way permutation (OWP) *if the following three conditions hold*

1. *Permutation: For each* $k \in \mathbb{N}$, f *is an injective and onto map from k-bit strings to k-bit strings.*
2. *Easy to compute: There exists a deterministic polynomial-time Turing machine M, so that on input x the turing machine M outputs* $f(x)$.
3. *Hard to invert: For every PPTM* M'*, every polynomial* $p(\cdot)$*, there exists an integer* k_0 *such that for all* $k > k_0$

$$\Pr\left[M'\left(f(U_k)\right) = f^{-1}\left(f(U_k)\right)\right] < \frac{1}{p(k)}$$

Note that there is in fact a collection of permutations, each on the set $\{0,1\}^k$. A more general and common definition is to have each permutation defined on some set $D_k \subseteq \{0,1\}^{l(k)}$, for some fixed polynomial $l(\cdot)$. We adopt this simplified version as in [28].

Emsembles and Indistinguishability

Definition 3 (Ensembles). *Let I be a countable index set. An* ensemble *indexed by I is a* sequence *of random variables indexed by I. Namely,* $X = \{X_i\}_{i \in I}$*, where the* X_i*'s are random variables, is an ensemble indexed by I.*

Typically, $\mathbb{N}$ is the index set, and each X_i ranges over bit strings of length i. Here in our applications, we will use $\mathbb{N}$ as the index set as usual, but X_i will be ranging over bit strings of length $l(i)$, for some fixed polynomial $l(\cdot)$. For example, ensemble of the form $\{F(U_i)\}_{i \in \mathbb{N}}$, where F is a length-tripling function, will be used.

Definition 4 (Computational Indistinguishability). *Two ensembles,* $\{X_n\}_{n\in\mathbb{N}}$ *and* $\{Y_n\}_{n\in\mathbb{N}}$*, are* indistinguishable in polynomial-time *if for every PPTM* M*, every polynomial* $p(\cdot)$*, there exists an integer* n_0 *such that for all* $n > n_0$

$$\left|\Pr[M(X_n, 1^n) = 1] - \Pr[M(Y_n, 1^n) = 1]\right| < \frac{1}{p(n)}$$

That is, two ensembles are indistinguishable if for every PPTM, the distance between the above two probabilities is a negligible function in n. The probabilities are taken over the corresponding random variables and the internal coin tosses of the PPTM.

Private Information Retrieval (PIR)

Informally, A PIR protocol allows a user to retrieve information from a database while maintaining the query private from the database manager.

Two-party Protocols A two-party protocol is a probabilistic process where two parties (two PPTMs), exchange messages (bit strings) in turns. Each message sent by a party is a function of its input, its random string and previous messages exchanged. A *pass* of the protocol consists of a *single* message sent from one party to the other. And the total messages exchanged during the process is called the *conversation*.

PIR protocols We model the database as an n-bit string x and model the query as a $(\log n)$-bit string i, called the index. The user is interested in the i^{th} bit of x, denoted by $x[i]$. So a PIR protocol is a two-party protocol with two PPTMs M_S (the server) and M_U (the user). The input to M_S is x and the input to M_U is i and 1^n (the length of the database written in unary). At the end of the protocol M_U outputs $x[i]$, whereas i is "unknown" to M_U. That is, there is no PPTM can distinguish the conversation generated by the two parties when the user is interested in $x[i]$ and the conversation generated when the user is interested in $x[j]$. Furthermore, we require that the total number of bits sent from M_S to M_U is strictly smaller than n. Otherwise, sending the entire database to the user would be a trivial way to accomplish this task. Formally,

Definition 5 (Private Information Retrieval). *A two-party protocol between* M_S *and* M_U *is called a* private information retrieval *protocol if the following three conditions hold*

1. *Correctness: If both parties follow the protocol, then* $M_U(i, 1^n)$ *outputs* $x[i]$ *at the end of the protocol.*

2. *Security: For each pair of index sequences* $\{i_n\}_{n\in\mathbb{N}}$ *and* $\{j_n\}_{n\in\mathbb{N}}$*, where each* $i_n, j_n \in \{0,1,\ldots,n-1\}$*, the conversation ensemble* $\{\alpha_n\}$ *(when* M_U*'s index is* $\{i_n\}$*) and the conversation ensemble* $\{\beta_n\}$ *(when* M_U*'s index is* $\{j_n\}$*) are indistinguishable. Namely, for every PPTM* M*, every polynomial* $p(\cdot)$ *there exists an integer* n_0 *such that for all* $n > n_0$

$$\left|\Pr[M(\alpha_n, 1^n) = 1] - \Pr[M(\beta_n, 1^n) = 1]\right| < \frac{1}{p(n)}$$

3. *Communication Complexity: The total number of bits* M_S *sent to* M_U *is strictly smaller than* n*.*

Condition 1 is sometimes relaxed to allowing a negligible probability of error. Here in our application, the user always gets the correct answer.

We say that a server is *honest-but-curious* if the server follows the protocol and is *malicious* if it does not. A malicious server may behave in an arbitrary way, for example, it may alter the contents of x. Thus there are two versions of security in condition 2. Namely, security against honest-but-curious server and security against malicious server. We will show that even malicious-PIR "does not imply" OWPs.

Note that the security parameter k does not appear in the definition at all. For our purpose, we set $k = n^{1/\tau}$ for some constant $\tau \geq 2$.

2.2 Black-box Reductions

Oracle Machines An oracle PPTM, denoted by M^Γ, is a PPTM M that has access to a given oracle Γ, such that in one time step M may receive the answer to a single query to Γ. When discussing a primitive or a protocol relative to an oracle, we assume that all the machines that are involved (including the adversary that tries to break the primitive) are in fact oracle machines with access to the same fixed oracle. An oracle Γ may be the union of two oracles Ψ and Λ. We use the notation $M^{\Psi,\Lambda}$ to denote M^Γ.

Black-box Reductions A black-box reduction from a primitive P to a primitive Q is a construction of P out of Q that ignores the internal structure of Q. More precisely, this is a construction of two oracle PPTMs M and A_Q such that, if N is an implementation of Q then M^N is an implementation of P and for any adversary A_P that breaks M^N (as an implementation of P), A_Q^{N,A_P} breaks N (as an implementation of Q).

For example, assume that (M_S, M_U) is an implementation of a PIR protocol, then a black-box reduction from OWPs to PIR is a construction of two oracle PPTMs M and A_{PIR}

such that M^{M_S,M_U} computes OWPs and for any adversary A_{OWP} that break M^{M_S,M_U} (inverts the permutation with a non-negligible probability when the input is chosen randomly), $A_{PIR}^{M_S,M_U,A_{OWP}}$ breaks (M_S, M_U) (outputs the user's index with a non-negligible probability, given only the conversation). Our goal is to show that there are no black-box reductions from OWPs to PIR. With the following lemma, we only need to find an oracle relative to which PIR exists but OWPs do not.

Lemma 2.1. *Assume that there is a black-box reduction from a primitive P to primitive Q, then relative to any fixed oracle, the existence of Q implies the existence of P.*

Proof. Assume that there is a black-box reduction from primitive P to primitive Q, then there exists two oracle PPTMs M and A_Q such that M^N implements P whenever N implements Q and A_Q^{N,A_P} breaks N (as an implementation of Q) whenever A_P breaks M^N (as an implementation of P).

Let Γ be any oracle relative to which primitive Q exists. Thus there exists an oracle PPTM N^Γ that implements Q and there is no oracle PPTM with oracle access to Γ breaks N^Γ. In addition M^{N^Γ} is an implementation of P relative to Γ, by the assumption that black-box reduction from P to Q exists.

Now assume for contradiction that relative to Γ primitive P does not exists. That is, relative to Γ any implementation of P can be broken. Thus, there is an oracle PPTM A_P^Γ that breaks M^{N^Γ} (as an implementation of P relative to Γ). So $A_Q^{N^\Gamma,A_P^\Gamma}$ breaks N^Γ (as an implementation of Q relative to Γ) by the assumption that black-box reduction from P to Q exists. Clearly, $A_Q^{N^\Gamma,A_P^\Gamma}$ can be simulated by an oracle PPTM with oracle access only to Γ. This is a contradiction to the assumption that Q exists relative to Γ. □

Black-box Constructions A black-box construction from a primitive P to a primitive Q is like a black-box reduction, except that the adversary oracle PPTM A_Q may use the internal structure of the adversary oracle PPTM A_P. That is, for every machine A_P, there exists a machine A_Q such that, if A_P with oracle access to N breaks M^N (as an implementation of P) then A_Q with oracle access to N breaks N itself (as an implementation of Q).

The Oracle Separation Paradigm

Let P and Q be two cryptographic primitives. To separate P and Q with respect to black-box reductions, we construct an oracle Γ such that relative to Γ the primitive Q exists but P does not. This in itself is enough to conclude that there is no black-box reduction from P to Q. In fact, Impagliazzo and Rudich [14] suggested a more powerful methodology

which we follow in our separation. In [14], Γ is constructed as a union of an oracle O and a $\mathcal{PSPACE}$-complete oracle. The first step is to show that there exists an implementation of Q using only O, which is secure with respect to Γ. That is, the adversary trying to break Q has oracle access to both O and the $\mathcal{PSPACE}$-complete oracle. Second, prove that if $\mathcal{P} = \mathcal{NP}$ then relative to O there is no secure implementation of P. Since relative to a $\mathcal{PSPACE}$-complete oracle $\mathcal{P} = \mathcal{NP}$, we conclude that

- There is no black-box reduction from P to Q. Furthermore,
- "$\mathcal{P} = \mathcal{NP}$" implies that there is no black-box construction from P to Q. That is, to provide such a black-box construction is at least as hard as proving $\mathcal{P} \neq \mathcal{NP}$.

Our goal is to construct an oracle O such that an implementation of PIR using only O is secure respect to Γ (union of O and a $\mathcal{PSPACE}$-complete oracle). And show that if $\mathcal{P} = \mathcal{NP}$ there are no OWPs relative to O. Based on the result proven in [28, 21] that relative to a random oracle no OWPs exist, we construct the oracle O such that it is "almost" random and the non-random parts can help to build the protocol PIR.

For simplicity, we only show the first conclusion that no black-box reductions from OWPs to PIR exist, but the theorem also implicitly leads to the second conclusion mentioned above.

2.3 Mathematics Background

In this section we give the pigeonhole principle and some basic probability theorems that will be used later.

Theorem 2.2 (Pigeonhole Principle). *Let A be a Boolean matrix with a $1-\alpha\beta$ proportion of 1's. Then a $1-\alpha$ portion of the columns have at least a $1-\beta$ portion of 1's.*

Proof. It suffices to note that the worst case is when the 0's are concentrated in a α by β rectangle. □

Theorem 2.3 (Markov Inequality). *Let X be a nonnegative random variable and t be a positive real number. Then*

$$\Pr[X \geq t] \leq \frac{\mathrm{E}[X]}{t}$$

Proof.

$$\begin{aligned} \mathrm{E}[X] &= \sum_x \Pr[X = x] \cdot x \\ &\geq \sum_{x<t} \Pr[X = x] \cdot 0 + \sum_{x \geq t} \Pr[X = x] \cdot t \\ &= \Pr[X \geq t] \cdot t \end{aligned}$$

□

Theorem 2.4 (Chebyshev Inequality). *Let X be a random variable, and $\delta > 0$. Then*

$$\Pr\big[\big|X - \mathrm{E}[X]\big| > \delta\big] \leq \frac{\mathrm{Var}[X]}{\delta^2}$$

Proof.

$$\begin{aligned} \Pr[|X - \mathrm{E}[X]| > \delta] &= \Pr[(X - \mathrm{E}[X])^2 > \delta^2] \\ &\leq \frac{\mathrm{E}[(X - \mathrm{E}[X])^2]}{\delta^2} \end{aligned}$$

□

Theorem 2.5 (Borel-Cantelli Lemma). *Let $B_1, B_2, \ldots$ be events on the same probability space. Then $\sum_{n=1}^{\infty} \Pr[B_n] < \infty$ implies that $\Pr[\bigcap_{k=1}^{\infty} \bigcup_{n \geq k} B_n] = 0$.*

Proof. Let $B = \bigcap_{k=1}^{\infty} \bigcup_{n \geq k} B_n$, and $A_k = \bigcup_{n \geq k} B_n$. So that $B = \bigcap_{k=1}^{\infty} A_k$; in particular, $B \subseteq A_k$ for all k. If $\sum_{n=1}^{\infty} \Pr[B_n] < \infty$, then for every $\varepsilon > 0$ there exists a $k > 0$ such that $\sum_{n \geq k} \Pr[B_n] < \varepsilon$. We have

$$\Pr[B] \leq \Pr[A_k] \leq \sum_{n \geq k} \Pr[B_n] < \varepsilon$$

and since ε can be arbitrarily small, we must have $\Pr[B] = 0$. □

Lemma 2.6. *A uniformly-distributed, length-tripling function $f : \{0,1\}^* \to \{0,1\}^*$ is injective for sufficiently long inputs with probability one.*

Proof. The probability that f is not injective on the domain $\{x : x \in \{0,1\}^n\}$ is at most $2^n \cdot (2^n - 1)/2 \cdot 1/2^{3n} \leq 1/2^n$. Since $\sum_{n=1}^{\infty} 1/2^n < \infty$, by the *Borel-Cantelli* Lemma f is injective for sufficiently long inputs with probability one. □

Chapter 3

PIR Does Not Imply One-way Permutations

In this chapter we construct an oracle Γ relative to which there is a 2-pass PIR protocol but no one-way permutations. Let k be the security parameter. The oracle Γ consists of the following parts.

- A $\mathcal{PSPACE}$-complete oracle.
- An injective, uniformly-distributed, length-tripling function $F(\cdot,\cdot)$.
- A uniformly-distributed function $S : \{0,1\}^* \to \{0,1\}^k$.
- A uniformly-distributed function $T : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$.
- A function G defined as follows.

$$G(x,y) \equiv \begin{cases} x[i] \oplus T\big(S(x),m\big) & \text{if } |x| \text{ is power of 2 and } \exists i,m \in \{0,1\}^* \\ & \text{such that } |i| = \log|x| \text{ and } y = F(i,m) \\ & \text{(call such } (x,y) \textit{ valid}) \\ 0 & \text{otherwise} \\ & \text{(call such } (x,y) \textit{ invalid}) \end{cases}$$

Note that G is well defined as long as F is injective, and that we can essentially ignore the restriction on F being injective since, with probability one, a random length-tripling function is injective for sufficiently long inputs. And note that S and G are actually two families of functions $\{S_k\}_{k\in\mathbb{N}}$ and $\{G_k\}_{k\in\mathbb{N}}$.

The design of G is to allow the user, who had chosen m, to compute $x[i]$ whereas from the server's point of view (without the knowledge of m), G is the value of a random function

on some unknown input, which is a meaningless random bit. Furthermore, it is very hard to query G on a valid input without first querying F. However, $G\big(x,F(i,m)\big)$ can be computed without actually querying the oracle function if i and m are known, which is the case if F is queried first. So it is unlikely that G is "useful" to any one-party primitive.

3.1 2-pass PIR Using the Oracle

By padding redundant 0's, we may assume that the length of the database x is power of 2. For example, if $|x| = 513$, we pad 511 0's at the end of x. This increases the length by a factor of two at most. The following is a 2-pass PIR using the oracle Γ.

The user selects $m \in \{0,1\}^k$ uniformly and sends $\alpha_F = F(i,m)$ to the server. The server then sends $\alpha_S = S(x)$ and $\alpha_G = G(x,\alpha_F)$ back to the user. At the end of the protocol, the user outputs $\alpha_G \oplus T(\alpha_S, m)$.

Server		*User*
	$\alpha_F = F(i,m)$ $\longleftarrow$	Select $m \in \{0,1\}^k$ uniformly
	$\alpha_S = S(x),\ \alpha_G = G(x,\alpha_F)$ $\longrightarrow$	$x[i] = \alpha_G \oplus T(\alpha_S, m)$

Correctness If both the server and the user follow the protocol, the user gets $S(x)$ and $G\big(x,F(i,m)\big)$ at the end of conversation. Recall that $G\big(x,F(i,m)\big) = x[i] \oplus T\big(S(x),m\big)$. So with the konwledge of m and $S(x)$, the user can compute $x[i]$.

Communication Complexity The original condition requires that the total number of bits sent from the server to the user is strickly smaller than the length of the datatbase. However, the length of the database may be doubled since we padded some redundant 0's to make the length power of 2. So we need to make sure that $|\alpha_S| + |\alpha_G| = k+1$ is strickly smaller than $|x|/2$. We could, for example, set $k = |x|^{1/\tau}$ for any constant $\tau \geq 2$.

Security As for the security of the protocol, we shall show that the server is unable to distinguish α_F from α'_F, the only information sent by the user when it is interested in $x[i]$ and $x[j]$, respectively. If the server does not query the function G, the proof is standard using the fact that the rest of the oracle consists of merely random functions. However, unless the server queries G on some distinct (x_1,α_F) and (x_2,α_F) such that $S(x_1) = S(x_2)$, which is unlikely to happen since the range of S is large, G acts like a random function also. Note that the function S (as a random hash) in the definition of G is essential; if we define $G\big(x,F(i,m)\big) = x[i] \oplus T(m)$ instead of $x[i] \oplus T\big(S(x),m\big)$, a single $G\big(x,F(i,U_k)\big)$

may be a uniform distribution but for any x_1 and x_2, $G\big(x_1, F(i,U_k)\big)$ and $G\big(x_2, F(i,U_k)\big)$ are correlated.

Since n and k are polynomially related, we prove the security (indistinguishability of α and α') of the protocol in terms of k. We state, without proof, a lemma needed for the main theorem first, and then prove it after the theorem. Since every PPTM can be simutated by (non-uniform) polynomial-size circuits, we prove the stronger statement that there are no polynomial-size circuits can distinguish the two conversations.

Lemma 3.1. *Let $\{C_k\}_{k\in\mathbb{N}}$ be a family of polynomial-size oracle Boolean circuits and let $\{i_k\}_{k\in\mathbb{N}}$, $\{j_k\}_{k\in\mathbb{N}}$ be two sequences of indices with each $|i_k| = |j_k| \leq k^c$ for some constant c. Define a sequence of random variables $X_k(\Gamma, m) \equiv C_k^\Gamma\big(F(i_k,m)\big) - C_k^\Gamma\big(F(j_k,m)\big)$, where $m \in \{0,1\}^k$, then both $\left|\mathop{\mathrm{E}}_{\Gamma,m}\big[X_k\big]\right|$ and $\mathop{\mathrm{E}}_{\Gamma}\left[\Big(\mathop{\mathrm{E}}_{m}\big[X_k\big]\Big)^2\right]$ are bounded by $poly(k)/2^k$ for some polynomial $poly(\cdot)$.*

We now prove the main theorem.

Theorem 3.2. *With probability one over random Γ, for any constant c, any fixed pair of sequences of indices $\{i_k\}_{k\in\mathbb{N}}$ and $\{j_k\}_{k\in\mathbb{N}}$ with each $|i_k| = |j_k| \leq k^c$, the two ensembles $\big\{F(i_k,U_k)\big\}_{k\in\mathbb{N}}$ and $\big\{F(j_k,U_k)\big\}_{k\in\mathbb{N}}$ are indistinguishable, even by non-uniform circuits.*

Proof. We first show that for any pair of sequences $\{i_k\}_{k\in\mathbb{N}}$, $\{j_k\}_{k\in\mathbb{N}}$ and any polynomial-size oracle circuits $\{C_k\}_{k\in\mathbb{N}}$,

$$\Pr_{\Gamma}\left[\left|\Pr_{m\in\{0,1\}^k}\big[C_k^\Gamma\big(F(i_k,m)\big)=1\big] \quad - \quad \Pr_{m\in\{0,1\}^k}\big[C_k^\Gamma\big(F(j_k,m)\big)=1\big]\right| \right.$$
$$\left. \text{is negligible for all sufficiently large } k\text{'s}\right] = 1$$

Let $X_k(\Gamma,m) = C_k^\Gamma\big(F(i_k,m)\big) - C_k^\Gamma\big(F(j_k,m)\big)$ be a sequence of random variables, and

$\delta(k)$ be some function that we will determine later, then

$$\begin{aligned}
&\Pr_{\Gamma}\left[\left|\Pr_{m}\left[C_k^{\Gamma}\left(F(i_k,m)\right)=1\right]-\Pr_{m}\left[C_k^{\Gamma}\left(F(j_k,m)\right)=1\right]\right|>\delta(k)\right]\\
&=\Pr_{\Gamma}\left[\left|\mathop{\mathrm{E}}_{m}\left[X_k\right]\right|>\delta(k)\right]\\
&\leq\Pr_{\Gamma}\left[\left|\mathop{\mathrm{E}}_{m}\left[X_k\right]-\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|\right|+\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|>\delta(k)\right]\\
&=\Pr_{\Gamma}\left[\left|\mathop{\mathrm{E}}_{m}\left[X_k\right]-\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|\right|>\delta(k)-\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|\right]\\
&\leq\frac{\mathop{\mathrm{E}}_{\Gamma}\left[\left(\mathop{\mathrm{E}}_{m}\left[X_k\right]-\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|\right)^2\right]}{\left(\delta(k)-\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|\right)^2}\\
&=\frac{\mathop{\mathrm{E}}_{\Gamma}\left[\left(\mathop{\mathrm{E}}_{m}\left[X_k\right]\right)^2\right]-\left(\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right)^2}{\left(\delta(k)-\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|\right)^2}\\
&\leq\frac{\mathop{\mathrm{E}}_{\Gamma}\left[\left(\mathop{\mathrm{E}}_{m}\left[X_k\right]\right)^2\right]}{\left(\delta(k)-\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|\right)^2}
\end{aligned}$$

By Lemma 3.1 we know that both $\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|$ and $\mathop{\mathrm{E}}_{\Gamma}\left[\left(\mathop{\mathrm{E}}_{m}\left[X_k\right]\right)^2\right]$ are bounded by a negligible function $\varepsilon(k)$ ($>0\ \forall k\in\mathbb{N}$). Set $\delta(k)$ to be the negligible function $\sqrt[4]{\varepsilon(k)}+\varepsilon(k)$, we have

$$\begin{aligned}
&\Pr_{\Gamma}\left[\left|\Pr_{m}\left[C_k^{\Gamma}\left(F(i_k,m)\right)=1\right]-\Pr_{m}\left[C_k^{\Gamma}\left(F(j_k,m)\right)=1\right]\right|>\sqrt[4]{\varepsilon(k)}+\varepsilon(k)\right]\\
&\leq\frac{\varepsilon(k)}{\left(\sqrt[4]{\varepsilon(k)}+\varepsilon(k)-\varepsilon(k)\right)^2}\\
&=\sqrt{\varepsilon(k)}
\end{aligned}$$

By the definition of negligible function, there exists a k_0 such that $\varepsilon(k)<1/k^4$ for all $k>k_0$; $\sqrt{\varepsilon(k)}<1/k^2$ for all $k>k_0$. And since $\sum_{k=1}^{\infty}1/k^2$ converges, $\sum_{k=1}^{\infty}\sqrt{\varepsilon(k)}$ also converges. By the *Borel-Cantelli* Lemma, the probability over random Γ that $\left|\Pr_{m}\left[C_k^{\Gamma}\left(F(i_k,m)\right)=1\right]-\Pr_{m}\left[C_k^{\Gamma}\left(F(j_k,m)\right)=1\right]\right|$ is greater than the negligible function $\delta(k)$ for infinitely many k's is

zero. Thus with probability one over random Γ, $\left|\Pr_m\left[C_k^\Gamma\left(F(i_k,m)\right)=1\right]-\Pr_m\left[C_k^\Gamma\left(F(j_k,m)\right)=1\right]\right|$ is negligible for all sufficiently large k's.

For every combination of $\{i_k\}_{k\in\mathbb{N}}$, $\{j_k\}_{k\in\mathbb{N}}$ and $\{C_k\}_{k\in\mathbb{N}}$, we throw out the measure zero of Γ's relative to which $\left|\Pr_m\left[C_k^\Gamma\left(F(i_k,m)\right)=1\right]-\Pr_m\left[C_k^\Gamma\left(F(j_k,m)\right)=1\right]\right|$ is greater than $\delta(k)$ for infinitely many k's. Since there are only countably many $\{i_k\}$'s, $\{j_k\}$'s and $\{C_k\}$'s, we have thrown out measure zero in all. Thus the two ensembles $\left\{F(i_k,U_k)\right\}_{k\in\mathbb{N}}$ and $\left\{F(j_k,U_k)\right\}_{k\in\mathbb{N}}$ are indistinguishable relative to the remaining measure one of the oracles. □

We now prove Lemma 3.1.

Claim 3.3. $\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right|$ *in Lemma 3.1 is bounded by* $poly(k)/2^k$ *for some polynomial* $poly(\cdot)$.

Proof. Define the following probability events

- $B_1 : C_k^\Gamma$ makes oracle query to F or to T on (z,m) for some $z\in\{0,1\}^*$.
- $B_2 : C_k^\Gamma$ makes oracle queries to G on (x_1,y) and on (x_2,y), where $x_1,x_2,y\in\{0,1\}^*$ such that $x_1\neq x_2$ and $S(x_1)=S(x_2)\in\{0,1\}^k$.
- $B : B_1\cup B_2$.

and use the notation $w\models B$ to denote that B happens when the input to C_k^Γ is w.

Let $p=p(k)$ be a polynomial that bounds the size of the circuits C_k,[1] and note that on input $F(i_k,m)$ all "useful" information (anything that is different when the input is $F(i_k,m)$ and when the input is $F(j_k,m)$) that C_k^Γ can get is the function values $F(i_k,m)$, $G\big(x_1,F(i_k,m)\big)=x_1[i_k]\oplus T\big(S(x_1),m\big)$, …, $G\big(x_p,F(i_k,m)\big)=x_p[i_k]\oplus T\big(S(x_p),m\big)$. Similary on input $F(j_k,m)$, C_k^Γ gets $F(j_k,m)$, $G\big(x'_1,F(j_k,m)\big)=x'_1[j_k]\oplus T(S(x'_1),m)$, …, $G\big(x'_p,F(j_k,m)\big)=x'_p[j_k]\oplus T(S(x'_p),m)$. Observe that if C_k^Γ does not make oracle query to F nor to T on (z,m) for some $z\in\{0,1\}^*$, the information is just the values of random oracle on some unknown locations. Furthermore if none of $S(x_1)$, …, $S(x_p)$ ($S(x'_1)$, …, $S(x'_p)$) are equal, these locations are all distinct. So these two distributions, $\Big(F(i_k,m),G\big(x_1,F(i_k,m)\big),\ldots,G\big(x_p,F(i_k,m)\big)\Big)$ and $\Big(F(j_k,m),G\big(x'_1,F(j_k,m)\big),\ldots,G\big(x'_p,F(j_k,m)\big)\Big)$, are identical when we fix everything else but vary over $F(i_k,m)$, $F(j_k,m)$ and $T(*,m)$, where $T(*,m)=\{T(z,m):z\in\{0,1\}^*\}$. Thus conditioned on $F(i_k,m)\not\models B$ and $F(j_k,m)\not\models B$,

$$\mathop{\mathrm{E}}_{F(i_k,m),F(j_k,m),T(*,m)}\left[C_k^\Gamma\left(F(i_k,m)\right)-C_k^\Gamma\left(F(j_k,m)\right)\right]=0$$

[1] Note that the input length of C_k is $3(|i_k|+k)$ instead of k. Such polynomial exists, however, if $|i_k|\leq k^c$ for some constant c.

for any m and $\Gamma\backslash\{F(i_k,m)\cup F(j_k,m)\cup T(*,m)\}$. We have

$$\begin{aligned}
&\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[X_k\right]\right| \\
&= \left|\mathop{\mathrm{E}}_{\Gamma,m}\left[C_k^{\Gamma}\big(F(i_k,m)\big)-C_k^{\Gamma}\big(F(j_k,m)\big)\right]\right| \\
&\le \Pr_{\Gamma,m}\left[F(i_k,m)\models B \text{ or } F(j_k,m)\models B\right] \\
&\qquad\times\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[C_k^{\Gamma}\big(F(i_k,m)\big)-C_k^{\Gamma}\big(F(j_k,m)\big)\,\middle|\,F(i_k,m)\models B \text{ or } F(j_k,m)\models B\right]\right| \\
&\quad+\Pr_{\Gamma,m}\left[F(i_k,m)\not\models B \text{ and } F(j_k,m)\not\models B\right] \\
&\qquad\times\left|\mathop{\mathrm{E}}_{\Gamma,m}\left[C_k^{\Gamma}\big(F(i_k,m)\big)-C_k^{\Gamma}\big(F(j_k,m)\big)\,\middle|\,F(i_k,m)\not\models B \text{ and } F(j_k,m)\not\models B\right]\right| \\
&\le \Pr_{\Gamma,m}\left[F(i_k,m)\models B \text{ or } F(j_k,m)\models B\right]\cdot 1+1\cdot 0 \\
&\le 2\Big(\Pr_{\Gamma,m}\left[F(i_k,m)\models B_1\right]+\Pr_{\Gamma,m}\left[F(i_k,m)\models B_2\right]\Big) \\
&\le 2\Big(p(k)/2^k+\frac{p(k)(p(k)-1)}{2}\cdot 1/2^k\Big) \\
&= p(k)\big(p(k)+1\big)/2^k
\end{aligned}$$

□

Claim 3.4. $\mathop{\mathrm{E}}_{\Gamma}\left[\left(\mathop{\mathrm{E}}_{m}\left[X_k\right]\right)^2\right]$ *in Lemma 3.1 is bounded by* $poly(k)/2^k$ *for some polynomial* $poly(\cdot)$.

Proof. Events B_1, B_2 and B are defined as in Claim 3.3, and define in addition

- B' : C_k^{Γ} makes oracle query to G on $\big(z_1,F(z_2,m)\big)$ for some $z_1,z_2\in\{0,1\}^*$.

Let $m'\in\{0,1\}^k$. Recall that $G\big(x,F(i,m)\big)=x[i]\oplus T\left(S(x),m\right)$ and observe that $F(i_k,m)$, $F(j_k,m)$ and $T(*,m)$ do not affect $C_k^{\Gamma}\big(F(i_k,m')\big)$ if C_k does not make oracle query to F nor to T on (z_1,m), nor to G on $\big(z_2,F(z_3,m)\big)$, for some $z_1,z_2,z_3\in\{0,1\}^*$. Thus conditioned on "$m\neq m'$ and $F(i_k,m')\not\models B\cup B'$ and $F(j_k,m')\not\models B\cup B'$," both $C_k^{\Gamma}\big(F(i_k,m')\big)$ and $C_k^{\Gamma}\big(F(j_k,m')\big)$ are constant when we vary over $F(i_k,m)$, $F(j_k,m)$ and $T(*,m)$. Thus conditioned on "$m\neq m'$ and $F(i_k,m)\not\models B$ and $F(j_k,m)\not\models B$ and $F(i_k,m')\not\models B\cup B'$ and

$F(j_k,m') \not\models B \cup B'$,"

$$\begin{aligned}
&\mathop{\mathrm{E}}_{\substack{F(i_k,m),F(j_k,m)\\ T(*,m)}} \Big[\big[C_k^\Gamma\big(F(i_k,m)\big) - C_k^\Gamma\big(F(j_k,m)\big)\big] \cdot \big[C_k^\Gamma\big(F(i_k,m')\big) - C_k^\Gamma\big(F(j_k,m')\big)\big] \Big] \\
&= \mathop{\mathrm{E}}_{F(i_k,m),F(j_k,m),T(*,m)} \Big[C_k^\Gamma\big(F(i_k,m)\big) - C_k^\Gamma\big(F(j_k,m)\big)\Big] \cdot \text{ some constant} \\
&= 0
\end{aligned}$$

for any m, m' and $\Gamma \backslash \{F(i_k,m) \cup F(j_k,m) \cup T(*,m)\}$. We have

$$\begin{aligned}
&\mathop{\mathrm{E}}_{\Gamma}\Big[\big(\mathop{\mathrm{E}}_{m}\big[X_k\big]\big)^2\Big] \\
&= \mathop{\mathrm{E}}_{\Gamma}\Big[\mathop{\mathrm{E}}_{m}\big[C_k^\Gamma\big(F(i_k,m)\big) - C_k^\Gamma\big(F(j_k,m)\big)\big] \cdot \mathop{\mathrm{E}}_{m'}\big[C_k^\Gamma\big(F(i_k,m')\big) - C_k^\Gamma\big(F(j_k,m')\big)\big]\Big] \\
&\le \Pr_{\Gamma,m,m'}\big[m = m'\big] \cdot 1 + 1 \cdot \mathop{\mathrm{E}}_{\Gamma,m,m'}\Big[\big[C_k^\Gamma\big(F(i_k,m)\big) - C_k^\Gamma\big(F(j_k,m)\big)\big] \\
&\qquad\qquad \cdot \big[C_k^\Gamma\big(F(i_k,m')\big) - C_k^\Gamma\big(F(j_k,m')\big)\big] \,\Big|\, m \neq m'\Big] \\
&\le 1/2^k + \Pr_{\Gamma,m,m'}\big[F(i_k,m) \models B \,\big|\, m \neq m'\big] + \Pr_{\Gamma,m,m'}\big[F(j_k,m) \models B \,\big|\, m \neq m'\big] \\
&\quad + \Pr_{\Gamma,m,m'}\big[F(i_k,m') \models B \cup B' \,\big|\, m \neq m'\big] + \Pr_{\Gamma,m,m'}\big[F(j_k,m') \models B \cup B' \,\big|\, m \neq m'\big] \\
&\quad + 1 \cdot 0 \\
&\le 1/2^k + 4\big(p(k)/2^k + \frac{p(k)(p(k)-1)}{2} \cdot 1/2^k\big) \\
&\quad + 2 \Pr_{\Gamma,m,m'}\big[F(i_k,m') \models B' \backslash B \,\big|\, m \neq m'\big] \\
&\le 1/2^k + 2p(k)\big(p(k)+1\big)/2^k + 2 \cdot p(k)/2^{3k} \\
&\le \big(2(p(k))^2 + 4p(k) + 1\big)/2^k
\end{aligned}$$

□

3.2 No OWPs Relative to the Oracle

In this section we show that no OWP exists relative to Γ. It was shown in [28, 21] that no OWPs exist relative to a random oracle union a $\mathcal{PSPACE}$-complete oracle. We proceed by showing that the function G does not help building OWPs either.[2]

[2]Note that we can aasociate several random functions with a single random oracle, and that with probability one a random length-tripling function is injective for sufficiently long inputs.

A first attempt is to see if the function G can be constructed by using a random oracle union a $\mathcal{PSPACE}$-complete oracle. Unfortunately, this is not the case since it is proved in [14] that relative to a random oracle union a $\mathcal{PSPACE}$-complete oracle, no key agreement protocol exists. And by [5] we know that the existence of PIR implies the existence of oblivious transfer, which by [17, 7] implies the existence of key agreenment. These implication hold relative to any oracle because they are proved by black-box reduction. So if G can be constructed by using a random oracle union a $\mathcal{PSPACE}$-complete oracle, we have PIR protocol exists relative to a random oracle union a $\mathcal{PSPACE}$-complete oracle, which imlies the existence of key agreement protocol in such world.

However, in the one-party situation, valid inputs are hard to sample without first querying F since F is length-tripling. And $G\big(x,F(i,m)\big)$ can be computed without actually querying G if i, m and x are known, which is the case if F is queried first. So any oracle PPTM with oracle access to Γ can be "approximately" simulated by another oracle PPTM that does not query G. With a minor change to only the statement, but not the proof, of Theorem 9.2 and Theorem 9.3 in [28], we can show that no OWPs relative to Γ. Detail follows.

Assume for contradiction that there is an oracle PPTM M' computes a OWP on some positive measure δ' of oracles in Γ.[3] By the *Lebesgue Density* Theorem, there exists an oracle PPTM M, with a finite number of oracle answers hardwired, that computes a OWP on measure $1-\delta$ of oracles in Γ (for any constant $\delta > 0$ we choose). Let Γ' denote this subset of oracles relative to which M computes a OWP. We construct from M another oracle PPTM N that does not query G but outputs differently from M only on a small fraction of inputs. Then to invert M, we invert N instead.

Now consider inputs from $\{0,1\}^n$ and suppose that the running time of M is bounded by n^c, for some constant $c \geq 2$. The oracle PPTM N simulates M step by step, keeps track of the queries to F, e.g. a list of 2-tuples (i,m)'s, and replaces any query to G, say on (x_0,α_0), by the following. If $|\alpha_0| \leq 3c\log n$, N checks every $(i,m) \in \{0,1\}^{|\alpha_0|/3}$ to see if any of them satisfies $F(i,m) = \alpha_0$. This takes at most $2^{3c\log n/3} = 2^c \cdot n$ time, which is polynomial in n. If $|\alpha_0| > 3c\log n$, N then checks the list (the history of queries to F) to see if any (i,m) on which satifies $F(i,m) = \alpha_0$. This again takes at most polynomial time since the list is at most polynomially long. If the pair (i_0,m_0) such that $F(i_0,m_0) = \alpha_0$ can be found, by a brute-force way or by serching the list mentioned above, N replaces $G(x_0,\alpha_0)$ by $x_0[i_0] \oplus T(S(x_0),m_0)$. Otherwise N assumes (x_0,α_0) is an invalid inputs and replaces

[3] Such a δ' must exist since there are only countably many oracle PPTMs but uncountably many oracles in Γ. Otherwise for measure one of the oracles in Γ would have no machine computing a OWP. Here we regard Γ as a set of the oracles.

$G(x_0, \alpha_0)$ by 0.

For any input $x \in \{0,1\}^n$, $N(x) \neq M(x)$ only if M ever queries G on some valid (x,α) and that $|\alpha| > 3c\log n$ is not obtained previously by querying F. Then for any fixed random choice of M, $N(x) \neq M(x)$ for at most $n^c 2^{c\log n}/2^{3c\log n} = 1/n^c \leq 1/n^2$ of oracles in Γ, and hence for at most $1/((1-\delta)n^2) \leq 2/n^2$ of oracles in Γ', for $\delta \leq 1/2$.

Lemma 3.5. *There are less than $2/n$ fraction of n-bit strings y such that $N^{-1}(y) \neq M^{-1}(y)$ for more than $2/n$ of oracles in Γ'.*

Proof. Consider the Boolean matrix A with rows indexed by $y \in \{0,1\}^n$ and columns indexed $\gamma \in \Gamma'$, such that $A_{y,\gamma} = 1$ iff $N^{-1}(y) \neq M^{-1}(y)$ relative to γ. For each $x \in \{0,1\}^n$, $N(x) \neq M(x)$ for at most $2/n^2$ of oracles in Γ', and this contributes at most $2^{-n}4/n^2$ fraction of 1's to A. As there are 2^n different x's, the total fraction of 1's in A is at most $4/n^2$. By the *pigeonhole principle*, less than $2/n$ of rows in A have more than $2/n$ of columns of 1's. □

For any y, $M^{-1}(y)$ is unique relative to any oracle in Γ' since it is a permutation. So by Lemma 3.5, for less than $2/n$ fraction of n-bit strings, $N^{-1}(y)$ is not unique for more than $2/n$ of oracles in Γ'. In other words, there are more than $1-2/n$ fraction of n-bit strings y such that $N^{-1}(y)$ is unique for more than $1-2/n$ of oracles in Γ', and hence for more than $1-2/n-\delta > 1-\varepsilon$ of oracles in Γ, for any constant $\varepsilon > \delta$ and sufficiently large n's. Based on [21], Theorem 9.3 in [28] states that $\mathcal{P} = \mathcal{NP}$ implies the following: For any permutation Π on $1-\varepsilon$ random oracles, and for each y, $\Pi^{-1}(y)$ can be computed on $1-\sqrt{\varepsilon}$ random oracles in polynomial time. Observe that the proofs of Theorem 9.2 and 9.3 in [28] actually yield a stronger statement that unique $N^{-1}(y)$'s can be computed. We have

Lemma 3.6. *Suppose $\mathcal{P} = \mathcal{NP}$. There is a constant λ such that for every oracle PPTM N, there exists an oracle PPTM N' with the following property. For any $\varepsilon < \lambda$ and for any y, if $N^{-1}(y)$ is unique for $1-\varepsilon$ of random oracles, then $N'(y) = N^{-1}(y)$ for $1-\sqrt{\varepsilon}$ of random oracles.*

And we have the main theorem by slightly modifying Theorem 9.4 in [28] as follows.

Theorem 3.7. *For measure one of the oracles in Γ, relative to which there is no oracle PPTM that computes a one-way permutation.*

Proof. Choose $\delta < \lambda$ such that there exists ε with $\delta < \varepsilon < \lambda$ and $\varepsilon + \sqrt[4]{\varepsilon} < 1$. As oracles in Γ contain a $\mathcal{PSPACE}$-complete problem, relative to which we have $\mathcal{P} = \mathcal{NP}$. So for any n, there are more than $1-2/n$ of n-bit string y such that we can find $N^{-1}(y) = M^{-1}(y)$ for more than $1-\sqrt{\varepsilon}$ of oracles in Γ. By the *pigeionhole principle*, there are more than

$1-\sqrt[4]{\varepsilon}$ of oracles in Γ for which there are infinitely many n where we can computer $M^{-1}(y)$ for more than $1-2/n-\sqrt[4]{\varepsilon}$ fraction of n-bit strings y, violating the definition of one-way permutations. That is, M is one-way relative to less than $\sqrt[4]{\varepsilon} < 1-\varepsilon < 1-\delta$ fraction of oracles in Γ, a contradiction. □

Bibliography

[1] A. Beimel, Y. Ishai, E. Kushilevitz and T. Malkin. One-way functions are essential for single-server private information retrieval. STOC 1999.

[2] M. Bellare, S. Halevi, A. Sahai and S. Vadhan. Many-to-one trapdoor functions and their relations to public-key cryptosystems. CRYPTO 1998.

[3] A. Beimel, T. Malkin, and S. Micali. The all-or-nothing nature of two-party secure computation. CRYPTO 1999, 80-97.

[4] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan. Private information retrieval. FOCS 1995, 41-51.

[5] G. Crescenzo, T. Malkin and R. Ostrovsky. Single database private information retrieval implies oblivious transfer. EUROCRYPT 2000, 122-138.

[6] W. Diffie and M. E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory 1976, 22(6) 644-654.

[7] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. FOCS 2000.

[8] O. Goldreich, S. Goldwasser and S. Micali. How to construct random functions. Journal of the ACM 1986, 33(4) 792-807. Preliminary version in FOCS 1984, 464-479.

[9] S. Goldwasser and S. Micali. Probabilistic encryption. Journal of Computer and System Science 1984, 28 270-299. Preliminary version in STOC 1982, 218-229.

[10] O. Goldreich, S. Micali, A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. STOC 1987, 218-229.

[11] O. Goldreich, S. Micali, A. Wigderson. Proofs that yield nothing but their validity, and a methodology of cryptographic protocol design. Journal of the ACM 1991, 38 691-729.

[12] J. Håstad, R. Impagliazzo, L. Levin and M. Luby. Construction of a pseudo-random generator from any one-way function. SIAM Journal on Computing 1999, 28(4) 1364-1396.

[13] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. FOCS 1989, 230-235.

[14] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. STOC 1989, 44-61.

[15] R. Impagliazzo and D. Zuckerman. How to recycle random bits. FOCS 1989, 248-253.

[16] M. Jerrum, L. Valiant and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. Theoretical Computer Science 1986, 43 169-188.

[17] J. Kilian. Founding cryptography on oblivious transfer. STOC 1988, 20-31.

[18] S. Kurtz, S. Mahaney and J. Royer. The isomorphism conjecture fails relative to a random oracle (extended abstract). STOC 1989, 157-166.

[19] E. Kushilevitz and R. Ostrovsky. Replication is not needed: single-database computationally private information retrieval. FOCS 1997.

[20] E. Kushilevitz and R. Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. EUROCRYPT 2000, 104-121. 157-166.

[21] J. Kahn, M.Saks and C. Smyth. A dual version of Reimer's inequality and a proof of Rudich's conjecture. IEEE Conference on Computational Complexity 2000.

[22] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal on Computing 1988, 17(2), 373-386. Preliminary version in STOC 1986.

[23] M. Naor, Bit commitment using pseudorandom generators. Journal of cryptology 1991, 4:151-158.

[24] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. STOC 1989.

[25] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. Proceedings of the second Israel Symposium on Theory of Computing and Systems 1993, 1-10.

[26] J. Rompel. One-way functions are necessary and sufficient for secure signatures. STOC 1990.

[27] M. O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[28] S. Rudich. Ph.d. thesis.

[29] S. Rudich. The use of interaction in public cryptosystems. CRYPTO 1991, 242-251.

[30] A. C. Yao. Theory and applications of trapdoor functions. FOCS 1982, 80-91.

[31] A. C. Yao. Protocols for secure computations (extended abstract). FOCS 1982, 160-164.

Appendix A: Detail Proofs of Lemma 3.3 and Lemma 3.4.

Recall the probability events,

- $B_1 : C_k^\Gamma$ makes oracle query to F or to T on (z,m) for some $z \in \{0,1\}^*$.
- $B_2 : C_k^\Gamma$ makes oracle queries to G on (x_1,y) and on (x_2,y), where $x_1,x_2,y \in \{0,1\}^*$ such that $x_1 \neq x_2$ and $S(x_1) = S(x_2) \in \{0,1\}^k$.

We have for every $m' \in \{0,1\}^k$,

$$\begin{aligned}
\Pr_{\Gamma,m}\left[F(i_k,m') \models B_1 \middle| m \neq m'\right] &= \mathop{\mathrm{E}}_{\Gamma}\left[\Pr_{m}\left[F(i_k,m') \models B_1\right] \middle| m \neq m'\right] \\
&\leq \mathop{\mathrm{E}}_{\Gamma}\left[p(k)/(2^k-1)\right] \\
&\leq (p(k)+1)/2^k
\end{aligned}$$

And

$$\begin{aligned}
\Pr_{\Gamma,m}\left[F(i_k,m) \models B_1\right] &= \mathop{\mathrm{E}}_{m,\Gamma'}\left[\Pr_{F(i_k,m),T(*,m)}\left[F(i_k,m) \models B_1\right]\right] \\
&= \mathop{\mathrm{E}}_{\Gamma}\mathop{\mathrm{E}}_{m}\left[\Pr_{u,v}\left[u \models B_1\right]\right] \\
&\leq \mathop{\mathrm{E}}_{\Gamma}\left[p(k)/2^k\right] \\
&= p(k)/2^k
\end{aligned}$$

where $\Gamma' = \Gamma \backslash \{F(i_k,m) \cup T(*,m)\}$ and $\mathop{\mathrm{E}}_{\Gamma}\mathop{\mathrm{E}}_{m}\left[\Pr_{u,v}\left[u \models B_1\right]\right]$ means that we fix Γ (including $F(i_k,m)$ and $T(*,m)$) first, and then later replace $F(i_k,m)$ and $T(*,m)$ in Γ by u and v respectively when sampling them. This term makes sense because whether B happens does

not depends on the value of $F(i_k, m)$ nor on $T(*, m)$. Whether B happens or not depends on the input u and $G(*, u)$, and note that $G(*, u)$ is determined by v.

And

$$\begin{aligned} \Pr_{\Gamma, m}\left[F(i_k, m) \models B_2\right] &\leq 1/2^k + 2/2^k + \cdots + (p(k)-1)/2^k \\ &= p(k)(p(k)-1)/2^k \end{aligned}$$