**Historical Cryptographic Paper Study** **Crypto & Complexity Lab**

**Notes on Relativization & Black-Box Separation**

**August 27, 2015** **Scribe: Chi Ning Chou**

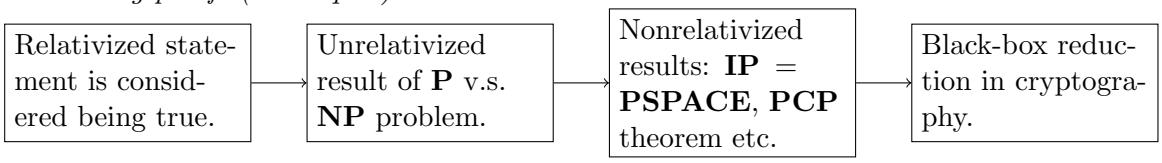
> Oracles do not relativize complexity classes, they only relativize the machines.
>
> *[HCC+92]*

# 1 The History and Intuition of Relativization

When it comes to relativization, people will come up with the idea about *oracles*. However, what is the importance of relativization in complexity theory and cryptography? Most theorist regards problem (or statement) that is **unrelativizable** as a hard one. And this was conjectured in the form of *Random Oracle Hypothesis*. [BG81]

Around 1985, the idea of *Interactive proofs* were invented [Bab85] [GMR85]. And in 1992, Shamir [Sha92] and Lund *et al* [LFKN92] showed that $\mathbf{IP} = \mathbf{PSPACE}$, which was a truly non-relativizing theorem! (While in 1994, [CCG+94] showed that for almost all oracles $A$, $\mathbf{IP}^A \neq \mathbf{PSPACE}^A$). Moreover, in 1992, Arora showed the powerful theorem *PCP Theorem* , which shows $\mathbf{NP} = \mathbf{PCP}(\log n, 1)$ and reveals the inapproximability of **PCP**.

In this section, I'll briefly introduce the history mentioned above and their implication. Most important of all, I hope everyone can get the intuition of what is *relativization*, *unrelativizable*, and *nonrelativizing proofs (techniques)*.




## 1.1 Oracles

Usually, we treat oracles in computational complexity theory and cryptography as a *black-box*, which means that an oracle will give the output correspond to the given input in constant time without leaking information. However, oracles are not just that simple, the reasons why theorists like to use it and its intuitions are important. The following summarize some of my aspects about oracles.

1. Black-box: Hide the description.
2. Additional resource: In a computational complexity point of view.
3. Additional information: In a information theoretic point of view.

4. A hash function: Random oracle in cryptography.

5. A set of strings (or string pairs): An analogy of a language, or a decision function.

### 1.2 Relativization

**Definition 1 (Relativizable)** *A statement $S$ is* **relativizable** *if $\forall$ oracle $A$, the statement always hold true. Similarly, if $\forall$ oracle $A$, statement $S$ always fails to hold true, we also say $S$ is relativizable to hold false.*

*On the other hand, once $S$ can hold true for some oracle $A$ but does not hold true for other oracle $B$, then we call $S$* **unrelativizable** *(also known as* **contradictory relativization***).*

In 1975, Baker, Gill and Solovay published the famous paper [BGS75] showing the following results:

**Theorem 2 (Baker-Gill-Solovay)** *Consider the* **P** *versus* **NP** *problem, there exists two oracle $A$ and $B$ such that*

- $\mathbf{P}^A = \mathbf{NP}^A$
- $\mathbf{P}^B \neq \mathbf{NP}^B$

This is a shocking result for that time since people in that day believed most of the statements are like the theorems in recursion theory that are all relativizable. Facing such weired unrelativizable result, theorists believed unrelativizable statements are hard to prove (e.g. Meta-Theorem [Hop84]) or event unprovable (Some believed maybe such problems is like Gödel's incompleteness theorems).

Such unrelativized statements $S$: $\exists$ oracles $A$ and $B$ such that $S^A$ holds true while $S^B$ doesn't hold true are also called *contradictory relativization*.

### 1.3 Hardness of Unrelativizable Results

After the $\mathbf{P}^A = \mathbf{NP}^A$, $\mathbf{P}^B \neq \mathbf{NP}^B$ result, people soon found out that almost every known techniques to prove the complexity results are useless to solve this kind of problems. And theorists view such phenomenon in the following aspects:

1. Oracles: People view oracles as *additional resources*. As a result, we can view the relativizing (or unrelativizing) results (suppose in the $S : C_1^A = C_2^A$ form) as the two complexity classes behave the same (or different) with the same additional supports.

2. Proof structure [Ver94]: The proofs of results having form $\exists$ *oracle $A$ s.t.* $S_1^A \neq S_2^A$ consists of two parts:

   (a) Diagonal: constructing the oracle steps by steps

   (b) Combinatorial: prove that every step can be made

   That is, the to show two complexity classes are different with the help of same oracle $A$, the structures of the two classes are inherently distinct such that the amount of required time or the depth of decision tree might grow in different level in the combinatorial part.

3. Restricted model[BLS84] [Boo87]: The *oracle access pattern* also matters. For example, just as 2. said, the number of access to the oracle in **NP** might be exponentially many more than the number in **P**. Therefore, some theorists had been devoted into this *oracle restricted model* and showed that once the oracle access mechanism for different classes are made roughly the same (comparable), then relativization separation would also results in the separation in nonrelativization. For example, if the oracle access for **P** and **NP** are the same and someone show $\mathbf{P}^A \neq \mathbf{NP}^A$ under such restriction, then we have $\mathbf{P} \neq \mathbf{NP}$.

4. Random Oracle: Bennett and Gill [BG81] proposed the *random oracle* approach in order to avoid highly structured oracle (which exploit the differences in the access mechanism) w.h.p. Since random oracles are intuitively **unbiased** and **unstructured**, they conjectures the famous *Random Oracle Hypothesis* stating that

   **Conjecture 3 (Random Oracle Hypothesis)** *A relativized result which holds true with probability 1 for a random oracle should also holds true in the absence of an oracle. In other words, random oracle separates the complexity classes with either probability 0 or 1.*

   In other words, once a statement being found contradictory unrelativizable, it is generally consider impossible to solve.

   An important observation about the random oracle result for $\mathbf{P}^A \neq \mathbf{NP}^A$ is that: It's difficult for people to find an oracle which relativize $\mathbf{P} \neq \mathbf{NP}$ but if we randomly choose an oracle, w.h.p the statement will be relativize! Intuitively, we can find out the limitation of the proof skill of human is weaker than randomness in some sense!

### 1.4 Nonrelativizing Results

As in the introduction of this section mentioned, with the invention of interactive proof, the random oracle hypothesis was seem to be broken. The following is a brief history line about what had happened:

- [LFKN92] $\mathbf{PH} \subseteq \mathbf{IP}$ v.s.[FS88] $\mathbf{co\text{-}NP}^A \not\subseteq \mathbf{IP}^A$
- [Sha92] $\mathbf{IP} = \mathbf{PSPACE}$ v.s. [CCG$^+$94] $\mathbf{IP}^A \neq \mathbf{PSPACE}^A$
- [AS98] $\mathbf{NP} = \mathbf{PCP}(\log n)$ v.s. $\mathbf{NP}^A \neq \mathbf{PCP}^A(\log n)$

### 1.5 Intuition of Relativization and Nonrelativization

In this subsection, I will give a summarization for the brief introduction to the history and intuition of relativization:

1. [HCC$^+$92] *Oracles do not relativize complexity classes, they only relativize the machines.*
2. [HCC$^+$92] Any proofs which uses **direct simulation** and **diagonalization** must relativize.
3. [HPV77] Space bounded computations are **strictly** more powerful than time bounded computations.

4. [HCC+92] The difference between relativized computation and nonrelativized computation is that

   - The step size of the nonrelativized computation are much smaller because of the restricted of a finite transition table.
   - The configurations of the nonrelativized computation are based more on local information.

   Intuitively, the nonrelativized computation is more **coherent**.

# 2 Limits of the Provable Consequences of OWP

In this section, I will introduce the famous work done by Russell Impagliazzo and Steven Rudich: *Limits on the provable consequences of one-way permutations* [IR89]. The work was inspired by the following question:

*Which assumptions are too weak to yield a proof that a secure protocol for task* **P** *is possible*?

The paper showed that OWP (One-Way Permutation) is too weak to guarantee SKA (Secret Key Agreement) in a sense of there is a contradictory relativization given oracle access to OWP. In other words, they showed that

**Theorem 4** *If* $\mathbf{P} = \mathbf{NP}$ *and all parties have access to OWP, then it is impossible to build secure SKA. Conversely, To prove a SKA with access to OWP is secure is as hard as proving* $\mathbf{P} \neq \mathbf{NP}$.

The paper has the following three important results (they are tightly related):

1. Showing the relationship between *cryptographic assumptions* and *cryptographic tasks*. For example, $\mathbf{P} = \mathbf{NP} + \mathbf{OWP}$ *exists* $\not\rightarrow$ *secure* **SKA**
2. Propose a framework to separate the power of cryptographic assumption such as OWP, OWP with trapdoor etc.
3. Black-box reduction: $A$ is black-box reducible to $B$, $A \leq_b B$, means that if $B$ holds true relative to an oracle $O$ then $A$ also holds true relative to $O$. Note that here $A$ and $B$ are statement or assumption. For example $A =$ "$\mathbf{P} = \mathbf{NP} + \mathbf{OWP}$ *exists*", $B =$ "*secure* **SKA**".

## 2.1 More Intuition on Black-Box Separation

First, since the *reduction* and *separation* mentioned here all involve so called *cryptographic assumptions* and *cryptographic tasks*, let's see some examples of them:

Before we go deeper into this paper, we should get some basic understanding about *Black-Box reduction* and *Black-Box separation*. Here, I'll give formal definitions [Fis12] and some intuitions:

**Definition 5 (Black-Box reduction)** *A Black-Box reduction from statement A to statement B is a process that only looks at the input/output behaviors of the oracle. Which means that it doesn't use any internal information or structure of the oracle. Here, you can think the oracle as the adversary or some given primitives.*

| Cryptographic Assumptions | Cryptographic Tasks |
|---|---|
| OWP | private-key encryption |
| PRG | bit-commitment |
| trap-door permutations | oblivious transfer |
| two-to-one collision functions | secret agreement |
| | electronic voting |
| | identification |
| | electronic signatures |
| | coin flipping by telephone |

Table 1: Cryptographic assumptions and tasks

For example, the following cryptographic primitives can be derived to each others in a black-box ways [Fis12]: OWF (one-way functions), PRG, PRF. However, once there is a positive direction, there must be some geek proposed another direction also works. And in this paper, Impagliazzo and Rudich came up with the concept of *Black-Box Separation*:

**Definition 6 (Black-Box separation)** *A black-box separation between two statements is the process that it is shown that there exists no black-box reduction among them.*

Take the OWP and SKA example, they showed that there is an oracle unrelativized the existence of OWP and the possibility of SKA, which indicates that there is no black-box reduction between them. As a result, this ended up with a black-box separation among the existence of OWP and the possibility of SKA.

And in this paper, they give more separation results as listed in the following table (see Table: 2.1):

- There is an oracle $O$ relative to which all $A$'s hold, but all $B$'s do not.
- If we use an assumption $\mathcal{A}$ to prove the tasks in $B$, then it is impossible to prove $\mathcal{A}$ through all of $A$ via black-box reduction.
- For example, secret exchange protocol (SKA) cannot be constructed from a one-way permutations (OWP).

| A | B |
|---|---|
| OWP exists | secure SKA is possible |
| Signature scheme exists | Oblivious transfer is possible |
| PRG exists | Trapdoor functions exists |
| Private-key encryption exists | Voting schemes exists |
| $\vdots$ | |

Table 2: Black-Box separation

The intuition of black-box separation is that it in some sense (at least in every possible relativized techniques) the two assumptions (ot tasks) are in the different level of hardness. For example, the

paper said that to show SKA is secure with OWP (of course, in a nonrelativized way) is as hard as showing $\mathbf{P} \neq \mathbf{NP}$. That is, we can intuitively think of the negative result is guaranteed by the hardness of showing $\mathbf{P} \neq \mathbf{NP}$.

**Remark 7** As a little remark, the common black-box separation techniques are:

- Relativizing techniques: by [IR89]
- Fully black-box reductions: The two-oracle techniques by [HR04]
- The Meta-Reduction technique: by [BV98] (Breaking RSA may not be equivalent to factoring)

## 2.2 Uniform Generation

**Intuition** Think of the following three questions:

Q: Can we generate a relation over a PPTM?

Q: Given an input $I$, an output $O$ and a PPTM $M$, can we find a computation (transcript) from $I$ to $O$ over $M$ with non-negligible probability?

Q: Given the conversation between two PPTM $M$ and $N$, can we find the computation of $M$?

First, we need to have the concept of *polynomial relation* and *uniform generation*. To save our time, I'll give a informal definition as follow:

**Definition 8 (polynomial relation, uniform generation)**
*Polynomial relation: A relation xRy that can be decided in poly-time w.r.t the size of x and y.*
*Uniform generation: Given x and relation R, pick y uniformly at random such that xRy.*

We can think of a polynomial relation $xRy$ analogous to the intuition above in the following way:

$x$: Input $I$, output $O$

$R$: PPTM $M$

$y$: A valid computation from $I$ to $O$ over $M$

Once we think in this way, the following theorem will be awesome:

**Theorem 9 ([JVV86])** *For any polynomial-time relation, there exists a PPTM equipped with a $\Sigma_2^p$ oracle that uniform generates it.*

The reason why the theorem is awesome is because that we can derived the following corollary:

**Corollary 10** $\mathbf{P} = \mathbf{NP} \Rightarrow$ *given a conversation between two PPTMs $M$ and $N$, we can uniformly generate a possible computation of $M$.*

## 2.3 Random Oracles, Random Function Oracles and Random Permutation Oracles

Let's start with the definition of the three important primitives:

**Definition 11 (random oracles, random function oracles, random permutation oracles)** *Define:*

1. *Let $r \leftarrow [0,1]$ uniformly, then a random oracle $R$ is the set induced from $r$:*

$$\{x : \textit{the } x\textit{th binary digit of } r \textit{ is } 1\}$$

2. *We can associate a random oracle $R$ to a function $f : \{0,1\}^n \to \{0,1\}^n$ in the following way:*

$$(f(i))_j := \mathbf{1}_{\{(2i+1)2^j \in R\}}, \forall i \in \{0,1\}^n$$

   *, where $(f(i))_j$ denotes the $j$th bits of $f(i)$. Then we call such function $f$ as random function oracle.*

3. *Once we add the constraint that $f$ must be 1-1 onto in 2., $f$ becomes a random permutation oracle.*

**Remark 12**

- We can think of *random oracles* $\Leftrightarrow$ *random function oracles* since over all possible random oracles, we get $f$ uniformly random.
- Suppose PPTM $T$ wants to invert $f$. Denote the output distribution of $T$ with access to oracle $f$ and given input $x$ as $T^f(x)$.
- For PPTM, there is no different between random function oracles and random permutation oracles.

What really matters to cryptographer about random oracles is just as mentioned in the first section of this notes: random oracle model [Gre11]:

Basically, random oracle model provides a provable security scheme such that we can build cryptographic construction under such scheme and argue its security. Intuitively, random oracle model allows all parties to access a random oracle, which can also be think of as accessing a large hash function.

Most importantly, In this model, no one can compute the hash function alone, they **must** ask the oracle! That is, all their access to the oracle can in some sense being revealed to everyone, which is totally different then hiding the hash function in their own separated pockets. And this is what we benefit from random oracle model. However, everything has some downside, just as Impagliazzo and Rudich showed that when we use OWP in our random oracle model, we can not guarantee the security of SKA. That is, provable security of OWP under the random oracle model scheme is limited.

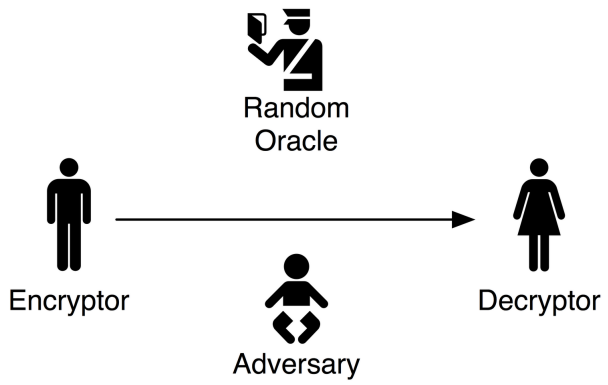



Figure 1: The random oracle model: all parties can have access to the oracle.

### 2.4 Cryptographic Lower Bounds

**Goal** Show that the existence of OWP is not an assumption for proving SKA is secure.

**Theorem 13** **P = NP** $\Rightarrow$ *relative to a random permutation oracle, any secret key agreement scheme can be broken.*

**Corollary 14** *There exists an oracle relative to which a strongly one-way permutation exists, but secure secret agreement is impossible.*

**Intuition**

1. The secret is an *intersection query*.
2. By Corollary 10, Eve can find the valid computation of Alice and Bob with non-negligible probability.
3. Eve can find all the intersection queries.
4. Eve will have a polynomial list containing the secret.

## 3 Conclusions

To sum up, a black-box separation tries to show the impossibility of certain proof techniques, for example, the proof using the existence of one-way permutation. Thus, it is important for cryptography since it provides a sense of impossibility. However, for computational complexity theorist, what they care are the separations between complexity classes (or computational models and resources). One focus on the relationship between cryptographic assumptions and tasks, the other emphasize on the resources and structures. Both are fundamental problems, but with different tastes.

## References


[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM (JACM)*, 45(1):70–122, 1998.

[Bab85] László Babai. Trading group theory for randomness. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 421–429. ACM, 1985.

[BG81] Charles H Bennett and John Gill. Relative to a random oracle a, \bfpˆa≠\bfnpˆa≠co-\bfnpˆa with probability 1. *SIAM Journal on Computing*, 10(1):96–113, 1981.

[BGS75] Theodore Baker, John Gill, and Robert Solovay. Relativizations of the p=?np question. *SIAM Journal on computing*, 4(4):431–442, 1975.

[BLS84] Ronald V Book, Timothy J Long, and Alan L Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13(3):461–487, 1984.

[Boo87] Ronald V Book. Towards a theory of relativizations: Positive relativizations. In *STACS 87*, pages 1–21. Springer, 1987.

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking rsa may not be equivalent to factoring. In *Advances in Cryptology—EUROCRYPT'98*, pages 59–71. Springer, 1998.

[CCG+94] Richard Chang, Benny Chor, Oded Goldreich, Juris Hartmanis, Johan Håstad, Desh Ranjan, and Pankaj Rohatgi. The random oracle hypothesis is false. *Journal of Computer and System Sciences*, 49(1):24–39, 1994.

[Fis12] Marc Fischlin. Black-box reductions and separations in cryptography. In *Progress in Cryptology-AFRICACRYPT 2012*, pages 413–422. Springer, 2012.

[FS88] Lance Fortnow and Michael Sipser. Are there interactive protocols for co-np languages? *Information Processing Letters*, 28(5):249–251, 1988.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 291–304. ACM, 1985.

[Gre11] Matthew Green. What is the random oracle model and why should you care? 2011.

[HCC+92] Juris Hartmanis, Richard Chang, Suresh Chari, Desh Ranjan, and Pankaj Rohatgi. Relativization: A revisionistic retrospective. In *Bulletin of the EATCS*. Citeseer, 1992.

[Hop84] John E Hopcroft. Turing machines. *Scientific American*, 250(5):86–107, 1984.

[HPV77] John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. *Journal of the ACM (JACM)*, 24(2):332–337, 1977.

[HR04] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In *Advances in Cryptology–CRYPTO 2004*, pages 92–105. Springer, 2004.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 44–61. ACM, 1989.

[JVV86] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM (JACM)*, 39(4):859–868, 1992.

[Sha92] Adi Shamir. Ip= pspace. *Journal of the ACM (JACM)*, 39(4):869–877, 1992.

[Ver94] Niekolai K Vereshchagin. Relativizable and nonrelativizable theorems in the polynomial theory of algorithms. *Izvestiya: Mathematics*, 42(2):261–298, 1994.