NOTIONS OF EFFICIENCY IN SIMULATION PARADIGM

TZER-JEN WEI

Department Of Applied Mathematics, National Dong Hwa University, Hualien 974, Taiwan
tjw@mail.ndhu.edu.tw

Abstract. There are some well-known conceptional and technical issues related to a common setting of simulation paradigm, i.e., EPT (expected polynomial time) simulator versus SPT (strict polynomial time) adversary. In fact, it has been shown that this setting is essential for achieving constant-round black-box zero-knowledge protocols. Many suggestions and results have been proposed to deal with these issues. In this paper, we propose an alternative solution. We study a new class of machines, MPT (Markov polynomial time), which is a cryptographic adaption of Levin's average polynomial-time. Since MPT has good compatibility to SPT and intuitive composition properties, we can use it as a drop-in replacement of SPT. Moreover, it is easy to construct simulators in MPT.

Key words. Markov polynomial-time, Average polynomial-time, Expected polynomial-time, Zero-knowledge, Simulation Paradigm

1. Introduction

1.1. Motivation.

1.1.1. *MPT.* In this paper, we study an alternative model of adversary, MPT (Markov polynomial time) machine. It can be used as a drop-in replacement of the standard model of efficient adversary in cryptography, namely SPT (strict polynomial time) machines.

MPT can be considered as one of many variations of Levin's average polynomial-time (see [5, 9] for more information on average polynomial time). There are many different design choices, which makes MPT suitable for cryptography.

However, the first question is, why even bother to consider an alternative model of efficient adversary at all?

1.1.2. *Why SPT standard?* The standard model of efficient adversary is SPT. One of the biggest advantage of SPT is convenient. SPT has many nice properties and easy to deal with in many context.

SPT captures many of our intuitive ideas of what an efficient machine should be. But more often than not, SPT is too loose or too restrict compare to the practical ideal of efficiency. For example, an n-bit RSA cracker that halt in $n^{\log\log n}$ seconds for the probability 99.99% is practically much more efficient than the one that has a uniform running time $n^{10^{100}}$ seconds. The real advantage of SPT is it's nice theoretical properties.

1.1.3. *Is SPT convenient for us?* SPT has issues when ZK (zero-knowledge) is involved in. ZK proofs and arguments have become essential and powerful constructing blocks of cryptographic protocols. The definition of ZK is based on simulation paradigm. Being ZK guarantees that whatever the adversary sees in the protocol is no different than the output generated by a simulator without knowing the secret. However, in many contexts, SPT adversary with EPT (expected polynomial time) simulator has become the default one for simulation paradigm. Obviously, the difference between the computation power of simulator and adversary is aesthetically unsatisfactory. However, it has been shown that this setting is some what essential. It is impossible to find a SPT black box simulator for constant round negligible soundness-error ZK argument (see [1, 2, 3]). Most practical ZK protocols falls in this category. This makes the standard definition of ZK literally EPT $-$ SPT $\neq 0$-knowledge. So there are many issues of the standard ZK definition needed to be take care of when we are proving the security of protocols. In most simple cases, these issues may not be that serious and can usually find a way to by pass them. After all, if the protocol is really secure, then one way or another we should be able to prove it. However, for more complex real world protocols, this issue may become annoying and even troublesome. In fact, our interest of this study was aroused while proving the security of our mental poker protocol.

1.2. **A good notion of efficient adversary.**

1.2.1. *Modeling simulator and adversary.* One solution of this issue is "require simulators to be no more powerful than adversaries" (See [10] Section 1.2). As we described in 1.1.3, Barak ([4]) shows that it is difficult to find simulators in SPT. The setting of EPT adversary with EPT simulator is also problematic. One reason is that EPT is not a robust class for composition like SPT. Readers can refer to [10, 8] for detail discussions on this topic and some interesting way to handle this issue.

Our solution falls in the category too. However, instead of using the usual class of machines SPT or EPT, we study an alternative one, MPT. In this paper, we argue that MPT is a better choice for modeling efficient adversary. When we say better, we mean MPT is intuitively no worse than SPT as a notion of efficiency and it is easier to handle when study theoretical properties.

In this subsection, we discuss what a good notion of efficient adversary should be.

1.2.2. *Probabilistic time bound.* As the example in 1.1.2, an n-bit RSA cracker does not need to be SPT to be harmful. Let us consider a cracker D which halts in polynomial time except a small probability, say for example negligible, $\frac{1}{n^2}$ or 1%. D is dangerous enough though it is not necessary SPT or even EPT.

One way of study of this algorithm D is that instead of study D itself, we study another algorithm D^*, which is a restriction of D on a polynomial time bound. Therefore D^* can decipher RSA except a small probability and is SPT. But one can easily argue that D is a better crack than D^*. After all, it is easier to get D^* by restricting D than to get D by extending D^*.

Therefore it is fair to say that setting efficient adversaries with probabilistic time bounds is intuitively no worse than setting them with deterministic time bounds.

1.2.3. *Probabilistic input.* On the other hand, we should also consider the input of the machine as a sample from a random distribution. Adversaries are typically modeled as probabilistic machines. The input of one machine is often the output of another. So it is straightforward to consider the input to be probabilistic.

In simulation paradigm, the simulator often query the black box adversary. These queries are indistinguishable (but may not be same) from those of honest parties. However, an efficient adversary should only be required to be efficient when interacting with honest parties (see [10] Section 1.2 for more discussion).

Therefore, it would be nice if a machine is efficient on an input source X, then it should be efficient on another input source Y that is indistinguishable from X. Indeed, efficiency ought to be a property concrete enough so that we can tell the difference. If X and Y are practically same to us, then no machine can practically separate X from Y , not even the running time.

1.2.4. *Backward compatibility.* Another consideration of a new notion of efficiency is whether it provides some compatibility to SPT. Many cryptographic results and assumptions are based on impossibility or possibility results of SPT algorithms. The benefit of a new notion of efficiency may not be worth if we have to rebuild the cryptography from scratch.

1.2.5. *Nice for simulation paradigm.* Finally, we would like both simulator and adversary to have same efficiency. Moreover, this class of efficient machines should have nice properties, especially composition properties. If an infeasible problem can be divided and conquered efficiently, then it is not really infeasible. Therefore, the composition of two efficient algorithms should be also considered as efficient.

1.3. **Our results.** In this paper, we prove several results to support that MPT fulfilled the above expectations in 1.2.

1.3.1. *Design choices.* To find the right definition, we made several design choices, namely

- We treat the input as a family of distributions.
- The time bound is asymptotic, not total.
- We choose the computational version of the definition.
- We use a more generalized and abstract way to define the input and machines.

Although the idea of average polynomial computing is not new, our approach has several advantages. Our definition captures the intuitive ideas of what a efficient adversary is. Other variations and some similar ideas can be found in [9, **?**]. The flexible probabilistic efficiency in [**?**] is very similar to our definition of perfect Markov polynomial-time, except our definition is more abstract and asymptotic. The same paper also mentions Vadhan's notion of efficient computation, which is also similar to our notion.

Our notions is a little unconventional. However, we think we have the right level of abstraction so that our definition is both fairly intuitive and flexible. Under suitable interpretations, it can be applied to various applications (see 3.3,5 for example).

1.3.2. *Intuitive meaning.* To summary, we claim that MPT has following properties

1. It is hardware model independent and closed under composition. (Section 3.3)
2. Stable to indistinguishable input. If two inputs X, Y are indistinguishable, then the running time a machine over X, Y should be indistinguishable. Otherwise, we can practically separate X from Y by their running time. If the running time of two machine are indistinguishable, they are supposed to be both efficient or both inefficient. (Section 3.5)
3. There exists efficient simulators for simulation paradigm and constructing such simulators is fairly easy. (Section 4)
4. Has compatibility to widely used standard definition of efficiency, so that we can still use standard cryptography assumptions. (Section 3.2)

In the rest of this paper, we provide several results to support our claim.

2. Basic Definitions

2.1. **Ensembles.** Recall that the usual definition of distribution ensemble is a series of random distributions. In this paper, we use a generalized definition as following.

Definition 1. Let I be a set. An *ensemble* X indexed by I is a family of random variables indexed by I. That is, $X=\{X_i\}_{i\in I}$, where the X_i are random variables. Call I the index set of X.

On other words, ensemble is simply a mapping $i\mapsto X_i$ from domain I into some collection of random variables. When there is no danger of confusion, we use the notation X_i, to represent both the random variable and its distribution μ_i. Moreover, if all X_i are the same random variable, we often omit the subscribe, hence X means either $\{X_i\}$ or a X_i depends on context.

Since a deterministic object can be viewed as a constant random variable, any function defined on I can be viewed as an ensemble with index I.

Throughout this paper, we implicitly assume following.

1. All probability measures are defined on all computable events. Therefore, every computable event has a probability.
2. The index I is equipped with a norm$|\cdot|:I\to\mathbb{N}$.
3. Moreover, denote by I_n the set $\{i\,|\,|i|=n\}$. There are infinitely many nonempty I_n.

We model the input of machine as ensembles. Our definition is more abstract than the usual one. Thus with creative interpretations, it can apply to a wide range of applications and can use for different purposes.

Typical examples of I are

- $I=\mathbb{N}$. $|i|=i$.
- $I=2^{<\mathbb{N}}$=$\{0,1\}^{<\mathbb{N}}$. $|i|$ is the length of i.
- More generally, for an arbitrary polynomial p, $I_n:=\left\{i\in 2^{<\mathbb{N}}|\text{ length of } i \text{ is } |p(n)|\right\}$.
- Let J be an arbitrary index set. Denote by $I=\bigcup J$ the set that $I_n=\{(j,n)\,|j\in J_m\wedge m\leq n\}$. I is an index set.
- Let I,J be index sets. $I\times J:=\{(i,j)\,|\,|i|=|j|\}$ and $|(i,j)|:=|i|=|j|$.
- Let I,J be index sets. $I\sqcup J:=\{(i,0)\,|i\in I\}\cup\{(j,1)\,|j\in J\}$ and $|(i,s)|:=|i|$.

We write $\forall^\infty i$ in short of $\forall^\infty n \forall |i| = n$, and $\exists^\infty i$ in short of $\exists^\infty n \exists |i| = n$.

Unless otherwise mentioned, we do pretty much all operations on ensembles component wise. For example, $X + Y$ means a new ensemble Z, such that $Z_i = X_i + Y_i$. $\Pr(X = 1)$ means a number ensemble $z = \{z_i\}$, where $z_i = \Pr(X_i = 1)$. $(X, Y, |X - Y|)$ means an ensemble Z, such that

$$Z_i = (X_i, Y_i, |X_i - Y_i|) .$$

Notable exceptions are

- Let X be an ensemble indexed by I and Y an ensemble indexed by J. $X \times Y$ is defined as
$$\{(X_i, Y_j) \mid (i, j) \in I \times J\} ,$$
and $Z = X \sqcup Y$ is defined as $Z_{(i,0)} = X_i$, $Z_{(j,1)} = Y_i$ and indexed by $I \sqcup J$.
- Let X be an ensemble indexed by I. $Y = \bigcup X$ is defined as $Y_{(i,n)} = X_i$ for all $(i, n) \in \bigcup I$.

Since $\times$, $\sqcup$ and $\cup$ usually make little or no sense when apply to random variables, the above exceptions won't cause much confusion. Most statements, if make sense, apply to ensembles component wisely too.

Call an ensemble of real numbers $\{f_i\}$ negligible if it is non-negative and

$$n \mapsto \sup_{|i|=n} f_i$$

is negligible. Call it noticeable if it is non-negative and

$$n \mapsto \inf_{|i|=n} f_i$$

is noticeable.

Therefore, a function $\varepsilon : I \to \mathbb{R}$ is negligible if the ensemble $\{\varepsilon(i)\}$ is negligible, noticeable if the ensemble $\{\varepsilon(i)\}$ is noticeable.

2.2. **Model of computation.** In this paper, *a machine* M=$\{M_i\}$ is a family of deterministic multi-tape Turing machines with some index set I. Inputs and outputs are put on tapes and may have non-finite support. The input $X = \{X_i\}$, output $M(X)$, and the running time $T = \{T_i \in \mathbb{N}^+\}$ are viewed as ensembles with the same index set I. For the sake of argument, we consider only the uniform family of machines unless otherwise mentioned. Therefore, we use M to denote both the family and the Turing machine. We use a Turing machine with a family of auxiliary-input to model non-uniform computation model, which we describe below. Followings are interpretations of various common computation models.

- For uniform computation model, $I = \mathbb{N}$. X_i typically has the uniform distribution on $2^{|i|} = \{0, 1\}^{|i|}$.
- For probabilistic computation model, we can use a tape R to provide the results of random coin tossing. Therefore, R is uniformly random on $2^{\mathbb{N}}$. We also use R to denote the ensemble $\{R_i = R\}$, when there is no danger of confusion.The input $X = \{(R, W_i)\}$, where $W_i$is uniformly random on $2^{|i|}$.
- More general, for probabilistic machine with auxiliary input. We fix an index set I. The input $X_i = \left(Z_i, R, W_{|i|}\right)$, where z_i is the auxiliary-input, R is the results of coin tossing, and $W_{|i|}$ is the random variable of other "regular" inputs. Let $Z = \{Z_i\}$, the ensemble of auxiliary inputs. Denote

by $M * Z_i$ the Turing machine M with the auxiliary-input Z_i wired in. Denote by $M * Z$ the ensemble $\{M * Z_i\}$ indexed by I.
- To execute an oracle querying, let the machine first write the query on a special tape. Then in a single step, the oracle returns the answer on another special tape, erases the input.
- Interactive machines are similar to oracle machines. Interaction with other machines can be simulated by oracle access to other machines' responses.
- Extended instructions. An extended instruction is works almost like an oracle, only it has full access to all tapes and states of the machine. Every extended instruction execution counts as one step too. Let M be s machine and S be a submachine of M. We allow submachines to have multiple start states. Let s be a start state of S. Whenever the state of M become s and let y denote the data on all tapes, we say that M calls S with parameter y. Denote by M/S the machine same as M except that S is replaced by an extended instruction with same functionality.

Our notion is a little bit unconventional. We use this notion so that we can study common models of computation and some uncommon models in a uniform way. Our notion allows the data on a tape to have infinite support or to be sampled from a nonuniform distribution. Since these relaxation makes little difference in our argument, this notion allows potential creative use of our results.

2.3. **Markov Polynomial Time.**

Definition 2. An ensemble $T = \{T_i\}$ is said to be *computational Markov polynomial growth* if there are non-constant polynomials f, g and a negligible function ε, such that

$$\forall^\infty i \forall s > 0 \left(\Pr\left(T_i > f(s, |i|)\right) < \frac{1}{s} + |g(s, |i|)|\, \varepsilon(|i|) \right).$$

If $\varepsilon = 0$, T is said to be *perfect Markov polynomial growth.*

The definition is inspired by Markov's inequality and should not be confused with other terminologies with similar names.

Definition 3. A machine respect to input (X, μ) is said to be *$\mu-$computational Markov polynomial time* ($\mu-$CMPT or simply CMPT), if its running time T is computational Markov polynomial growth. It is said to be *$\mu-$perfect Markov polynomial time* ($\mu-$PMPT or simply PMPT) if T is perfect Markov polynomial growth.

We use the term MPT in a statement when the statement is true for both CMPT and PMPT.

We say Γ is a class of machines, we mean Γ is a collection of machine and input pair $(M, X) = \{(M_i, X_i)\}$ that satisfies certain property.For example,

$$\{(M, X) : (M, X) \text{ is CMPT}\}$$

is a class of machines. We write $\mu-\Gamma$ or $\Gamma(X)$ or $\Gamma|X$ to denote the class Γ restricts to the input distribution ensemble (X, μ).

Recall that M is SPT if $\forall^\infty i\, (T_i \leq p(|i|))$ for some polynomial p and is EPT if $\forall^\infty i\, (E\,[T_i] \leq p(|i|))$ for some polynomial p. All EPT and SPT machines are MPT.

Let Σ be a class of auxiliary-inputs and Γa class of machines. Denote by $\Gamma * \Sigma$ the class of machine in Γ with auxiliary-input in Σ, i.e.,

$$\Gamma * \Sigma = \{(M * Z, X) \,|z \in \Sigma \wedge (M, Z \times X) \in \Gamma\} .$$

Thus $M * z \in \Gamma * \Sigma(X)$ means $(M, Z \times X) \in \Gamma$ and $M * Z(X)$ means $M\,(Z \times X)$. If M is in $\Gamma(X)$ when interacting with B, or replace submachine B by extended instruction, we write $M \in \Gamma(X)/B$.

The $*$ notation can be easily extended to a class of machines with a class of *randomized* inputs, e.g. random oracle, in the obvious way.

3. Properties

3.1. **Separable ensemble.**

Definition 4. An ensemble X is *separable* if there is an $M \in \mathrm{SPT}\,(X)$ such that $\Pr\,(M(X_i) = |i|) = 1$.

X is *semiseparable* if there are an $M \in \mathrm{SPT}\,(X)$ and a polynomial p such that $\Pr\,(M(X_i) \le p\,(|i|)) = 1$ for every i and $\Pr\,(M(X_i) < |i|)$ is negligible. Call the output of $M\,(X_i)$ the seminorm of X_i or in symbol $\langle X_i \rangle$.

The following proposition is straightforward.

Proposition 1. *Let X, Y be ensembles.*

1. *If X is separable, then $X \times Y$ is separable.*
2. *If X is semiseparable, then $X \times Y$ is semiseparable.*
3. *If X and Y are both separable, then $X \sqcup Y$ is separable.*
4. *If X and Y are both semiseparable, then $X \sqcup Y$ is semiseparable.*
5. *In particular, $\mathbb{N} \times X$ is separable, where $\mathbb{N}$ works as an ensemble the norm $|| : n \mapsto n$.*

In cryptography, the security parameter are not considered as a secret and usually understood by all participators. Therefore, the input ensemble can be modeled in the form $\mathbb{N} \times X$, hence separable.

3.2. **Relations with SPT.**

In this subsection, we study some basic properties that makes MPT compatible to SPT.

Theorem 1. *Let Σ be an arbitrary class of input and X a semiseparable input ensemble. Suppose $M \in MPT{*}\Sigma(X)$. For every $k \in \mathbb{N}$, there is a $M^* \in SPT{*}\Sigma(X)$, such that*

$$\forall^\infty i \left(\Pr(M^*(X_i) \neq M(X_i)) < \frac{1}{|i|^k} \right) .$$

Proof. Let f, g, ε be the witness of M in $\mathrm{MPT}(X)$. Let M^* be the machine that simulates M up to $f(2 \langle X_i \rangle^k, \langle X_i \rangle)$ steps. When the running steps of M exceeds the limit, M^* simply output something and halt. Clearly, M^* is in $\mathrm{SPT}(X)$. Since M is in $\mathrm{MPT}\,(X)$ and X is semi-separable, for large enough i, the probability that M doesn't halt in the time restriction is less than

$$\frac{1}{2\,|i|^k} + \varepsilon'(|i|),$$

where ε' is some negligible function.

Therefore, for large enough $|i|$,

$$\Pr(M^*(X_i) \neq M(X_i)) < \frac{1}{|i|^k}.$$

□

The following corollary is straightforward but useful.

Corollary 1. *Let X be a semiseparable input ensemble and $M \in MPT(X)$. For every set S, there is $M^* \in SPT(X)$, such that*

1. *If $\Pr(M(X) \in S)$ is noticeable, then $\Pr(M^*(X) \in S)$ is noticeable;*
2. *If $\Pr(M(X) \in S)$ is not negligible, then $\Pr(M^*(X) \in S)$ is not negligible;*
3. *If $\Pr(M(X) = 1 \iff X \in S)$ is noticeable, then $\Pr(M^*(X) = 1 \iff X \in S)$ is noticeable;*
4. *If $\Pr(M(X) = 1 \iff X \in S)$ is not negligible, then $\Pr(M^*(X) = 1 \iff X \in S)$ is not negligible.*

Let Γ be a class of machines with auxiliary input. Recall that we say that two ensembles X, Y are $\Gamma-$*indistinguishable* (we mean *computational indistinguishable*) if for every machine $M * z \in \Gamma\,(X \sqcup Y)$,

$$|\Pr((M * z)\,(X) = 1) - \Pr((M * z)\,(Y) = 1)|$$

is negligible.

Corollary 2. *Let Σ be a class of input indexed by J. If two semiseparable ensembles X, Y indexed by I are $SPT * \Sigma$-indistinguishable, then X, Y are also $MPT * \Sigma$-indistinguishable.*

Proof. We prove "not MPT$*\Sigma$-indistinguishable" $\Rightarrow$ "not SPT$*\Sigma$-indistinguishable".

Suppose X, Y are not MPT $* \Sigma$-indistinguishable, there is an $M * Z \in \text{MPT} * \Sigma\,(X \sqcup Y)$ and a $k \in \mathbb{N}^+$ such that,

$$\exists^\infty\,(j, i) \in J \times I \left(|\Pr(M * z_j(X_i) = 1) - \Pr(M * Z_j(Y_i) = 1)| \geq \frac{1}{|i|^k} \right)$$

By Theorem 1, there is a

$$M^* \in \text{SPT}\,(Z \times (X \sqcup Y))$$

such that

$$\forall^\infty\,(j, i) \in J \times I \left(\Pr(M^* * Z_j(X_i) \neq M * Z_j(X_i)) < \frac{1}{|i|^{2k}} \right)$$

and

$$\forall^\infty\,(j, i) \in J \times I \left(\Pr(M^* * Z_j(Y_i) \neq M * Z_j(Y_i)) < \frac{1}{|i|^{2k}} \right).$$

Therefore, there exist infinitely many $n = |i| = |j|$,

$$\begin{aligned} & |\Pr(M^* * Z_j(X_i) = 1) - \Pr(M^* * Z_j(Y_i) = 1)| \\ \geq\ & |\Pr(M * Z_j(X_i) = 1) - \Pr(M * Z_j(Y_i) = 1)| - \frac{2}{|i|^{2k}} \\ \geq\ & \frac{1}{|i|^{2k}}. \end{aligned}$$

Thus X, Y are not SPT $*$ Σ-indistinguishable. □

Moreover, let X_i be the initial input data on tapes and p be a polynomial. Denote by X_i^p the same initial input data as X_i except all squares on tapes that can not read by any machine in time $p(|i|)$ are blank. So for example, if the Turing machine has only one tape, then X_i^p represent an input that the distance between the head and every none blank square on tape is less than $p(|i|)$.

Let X be an input ensemble and p a polynomial. Denote by X^p the ensemble $\{X_i^p\}$. Let Σ^{PL} be the class

$$\{X^p : X \in \Sigma \text{ and } p \text{ is a polynomial}\}.$$

If $\Sigma^{PL} \subseteq \Sigma$, then clearly Σ-indistinguishablility implies Σ^{PL}-indistinguishability. On the other hand, if $M*Z \in$ SPT$*\Sigma$ distinguish X, Y, then $M*Z^p$ also distinguish X, Y, where p is the polynomial time bound of $M * Z$. Therefore, we have

Corollary 3. *Let Σ be a class of input such that $\Sigma^{PL} \subseteq \Sigma$. If two semiseparable ensembles X, Y are SPT $*$ Σ^{PL}-indistinguishable, then they are also MPT $*$ Σ-indistinguishable.*

When X, Y are SPT$*\Sigma$-indistinguishable, we say that they are Σ-indistinguishable.

3.3. Closeness under Compositions.

In this section, we show several composition properties of MPT.

Definition 5. Let M be a machine with input (X, μ) and S is a submachine of M. Let $N(x, y)$ be the number of calls of S with parameter y when M has input x, and

$$N(x) = \sum_y N(y, x).$$

The *joint distribution X, Y of (S, M, X)-submachine input* is defined by the density function $p_i(x, y) = \frac{N(x,y)}{N(x)}\mu_i'(x)$, where when $N(x) = \infty$, we artificially define

$$\frac{N(x, y)}{N(x)} = \begin{cases} 1 & y = 1^\infty \\ 0 & \text{otherwise} \end{cases}.$$

In other words,

$$\Pr(A|p_i) = E\left[\sum_{(X_i, y)\in A} \frac{N(X_i, y)}{N(X_i)}\right]$$

for any event A.

The *distribution ν of (S, M, X)-submachine input* is defined by the density

$$\nu_i'(y) = \int \frac{N(x, y)}{N(x)} d\mu_i(x) = E\left[\frac{N(X_i, y)}{N(X_i)}\right].$$

Although our definition of the distribution of submachine input looks artificial and technical, it actually captures the intuitive idea of what is the distribution of input of submachine. Informally, the definition basically says $\nu'(y) = E[\Pr(y|X)]$.

Another naive definition would be something like

$$\nu(y) = \frac{E[\text{total number of calling } S(y)]}{E[\text{total number of calling } S]}.$$

However, this definition can have little sense when M calls some $S(y)$ infinite times, and when it has sense, the composition theory is trivial and useless (EPT would have composition in this sense). An alternative definition is to consider the distribution on each different running steps of M. We discuss some corollaries related to this approach at the end of this subsection.

Theorem 2. *(Composition Theorem) Let M be a machine with input X and S be a submachine of M. Consider a new extend instruction that has the functionality of S and let $M^* = M/S$. If M^* is MPT and S is MPT respect to the distribution of (S, M, X)-submachine input, then M is in MPT.*

Proof. Assume

$$\forall^\infty i \forall s > 0 \left(\Pr\left(T^*_{|i|} > f(s, |i|)\right) < \frac{1}{s} + |g(s, |i|)|\, \varepsilon(|i|) \right),$$

and

$$\forall^\infty i \forall s > 0 \left(\Pr\left(T'_{|i|} > f(s, |i|)\right) < \frac{1}{s} + |g(s, |i|)|\, \varepsilon(|i|) \right),$$

where T^*, T' is the running time of M^*, S.

Let T be the running time of M. For an arbitrary given input x, $T > L \cdot K$ can only happens when either $T^*(x) > L$ or the T' of one of the submachine calls to S takes longer than K.

Let $B = \{y | T'_i(y) > K\}$ and A be the subset of input x of M that the at least one of the submachine calls to S takes longer than K. Assume $|i|$ is large enough, compute the following probability of for random variable X, Y over the joint distribution of subroutine call ,

$$\begin{aligned}
\Pr(T'_i(Y) > K \wedge T^*_i(X) \le L) &= \Pr(Y \in B \cap T^*_i(X) \le L) \\
&= \int_{T^*_i(x) \le L} \sum_{y \in B} \frac{N(x, y)}{N(x)} d\mu_i(x) \\
&\ge \int_{x \in A \wedge T^*_i(x) \le L} \frac{1}{N(x)} d\mu_i(x) \\
&\ge \int_{x \in A \wedge T^*_n(x) \le L} \frac{1}{L} d\mu_i(x) \\
&= \frac{1}{L} \Pr(X \in A \wedge T^*_i(X) \le L).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\Pr(T_i > LK) &\le \Pr(X \in A \cap T^*_i(X) \le L) + \Pr(T^*_i > L) \\
&\le L \Pr(T'_i(Y) > K) + \Pr(T^*_i > L).
\end{aligned}$$

Let $L = f(2s, |i|)$, $K = f(sf(2s, |i|), |i|)$ we have

$$\begin{aligned}
& \Pr\left(T_i > 2sf(f(2s, |i|), |i|)f(2s, |i|)\right) \\
\le\ & f(2s, |i|) \left(\frac{1}{2sf(2s, |i|)} + g\left(2sf(2s, |i|), |i|\right) \varepsilon(|i|) \right) + \frac{1}{2s} + g(2s, |i|)\varepsilon(|i|) \\
=\ & \frac{1}{s} + \left(f(2s, |i|)g\left(2sf(2s, |i|), |i|\right) + g(2s, |i|)\right) \varepsilon(|i|).
\end{aligned}$$

Therefore, M is MPT. □

Following is a typical usage of Theorem 2.

Let us consider the usual setting of a Turing machine with security parameter $n \in \mathbb{N}$, a working tape X_n, a random tape R and a read only tape z_n for auxiliary input. Call the distance n between the head and the far most non-blank symbol on a tape the *length* of data on the tape.

Let $\mathbb{I}_n$ be the collection of all input with length $n-1$, and I_0 contains the blank tape. Assume X_n is a random variable and $\Pr(X_n \in I_n) = 1$. There fore, the input of M is $Z_n \times X_n \times R$, which includes the auxiliary input, the regular input and random bits.

Let S be a submachine of M and Y_n be the uniform random variable on I_n. Assume $S \in \mathrm{SPT}\left(\bigcup Z \times \bigcup Y \times R\right)$, i.e., S has a strict polynomial total bound. Since M may call S with various length of parameter on working tape, the polynomial bound of S has to be total rather than an asymptotic bound like $S \in \mathrm{SPT}\left(Z \times Y \times R\right)$. Otherwise, M may call S with a short parameter and S doesn't even need to halt.

In many situation, S can be polynomially accelerated by a special purpose hardware. We can model this new hardware as an extended instruction.

We have the following corollary.

Corollary 4. *Let M be a Turing machine, z be an auxiliary S be a submachine of M, and $M^* = M/S$. Assume Y_n as uniformly random on $\mathbb{I}_n \times 2^{\mathbb{N}}$ and $Y = \{Y_n\}$ indexed by $\mathbb{N}$. If $M^* \in MPT(Z \times X \times R)$ and S is $SPT(\bigcup Z \times \bigcup Y \times R)$, then $M \in MPT(Z \times X \times R)$.*

Proof. Suppose M calls S at time t. Since M is a Turing machine, the length of data on the working tapes is bounded by $t + n$. It is easy to check that the distribution of $(S, M, Z \times X \times R)$-submachine input is MPT by the assumption that $M^* \in \mathrm{MPT}(Z \times X \times R)$. □

If the length of auxiliary input z_n growths polynomially, then S is total strict polynomial time in the usual sense, i.e., respect to the length of auxiliary input plus the length of data on working tape.

Remark. Thus unlike EPT, the MPT is intrinsic to model of computation. In fact, PMPT is the smallest such class that contains EPT (see 6.1).

In many applications, submachines are MPT in some sense and the calling parameters of given length are fairly independent to X. Corollary 4 can be easily generalized for these situations.

Corollary 5. *Let M be a machine with input X indexed by I, S be a submachine of M, Let $Y_{i,t}$ be the parameter of S when M/S calls S (as an instruction) at step t. Let Y be the ensemble indexed by $\{(i,t)\}_{i \in I, t \in \mathbb{N}}$ and $|(i,t)| = (i+t)^2 + t$.*

If $S \in MPT(Y)$ and $M/S \in MPT(X)$, then $M \in MPT(X)$.

Optionally, we can let S to simulate all instructions. Since regular instructions can be simulated in constant time, S is easier to be $\mathrm{MPT}(Y)$ if M calls S infrequently.

Results in 3.5 also help to verify the conditions of Theorem 2.

3.4. **Equivalence Conditions.** In this subsection, we proves many equivalence condition of MPT. Many notions based on the ideas in 1.2 are equivalent to MPT. Some others can be found in Section 5.

Theorem 3. *Let T be the running time of a machine M with input X. The followings are equivalent,*

1. *M is CMPT.*
2. There is a negligible ε, such that
$$\exists d > 0 \forall^{\infty} i \forall t > 0 \left(\Pr\left(T_i > t\right) < |i|\, t^{-d} + t\varepsilon(|i|) \right).$$
3. There is a negligible ε, such that
$$\exists d > 0 \forall^{\infty} i \forall t > 0 \left(\Pr\left(T_i > t\right) < |i|\, t^{-d} + \varepsilon(|i|) \right).$$
4. *There are a non-constant polynomial f and a super-polynomial S, such that*
$$\forall^{\infty} i \forall s \in (0, S(|i|)) \left(\Pr\left(T_i > f(s,|i|)\right) < \frac{1}{s} \right).$$
5. *There are a $d > 0$ and a super-polynomial S , such that*
$$\forall^{\infty} i \forall t \in (0, S(|i|)) \left(\Pr\left(T_i > t\right) < |i|\, t^{-d} \right).$$
6. *There are a non-constant polynomial f and a function $\epsilon : \mathbb{R}^{+} \times \mathbb{N} \to \mathbb{R}$, such that $n \mapsto \sup_{x<n^c} \epsilon(s,n)$ is negligible for every $c \geq 1$ and*
$$\forall^{\infty} i \forall s > 0 \left(\Pr\left(T_i > f(s,|i|)\right) < \frac{1}{s} + \epsilon(s,|i|) \right).$$
7. *There is a super-polynomial S and $d > 0$, such that*
$$\forall^{\infty} i \left(E\left[T_i^d | T_i < S\left(|i|\right)\right] \leq |i| \right).$$

Proof. (1⇒2) Since
$$\Pr\left(T_i > f(s,|i|)\right) < \frac{1}{s} + |g(s,|i|)|\, \varepsilon(|i|)$$
always holds when $0 < s \leq 1$, it is easy to see that we can assume $f(s,n) = s^l n^k$ and $g = s^l$, $l, k \geq 1$ in definition 2. Furthermore, we can choose $l = k = \max(l,k)$ and let $t = s^k n^k$.Thus the inequality in definition 2 become
$$\Pr(T_i > t) < \frac{|i|}{t^{\frac{1}{k}}} + t\varepsilon(|i|)$$
for some large enough k and negligible ε. Let $d = \frac{1}{k}$, we have 1⇒2.

(2 ⇒1) Assume 2 holds. 1 holds by letting $k = \lceil \frac{1}{d} \rceil$, $t = f(s,n) = s^k n^k$, $g(s,n) = s^k$.

(1 ⇒6) We can assume $g(s,n) = s^k$ and let $\epsilon(s,n) = g(s,n)\varepsilon(n) = s^k \varepsilon(n)$.

(6 ⇒4) Assume 6 holds. We can assume ϵ is increasing respect to s. Let $u = \sqrt{\frac{s}{2}}$. Let $h_n(u)$=$\frac{1}{u} - \frac{1}{2u^2} - \epsilon(2u^2, n)$. Observe that $h_n(u)$ is decreasing for $u \geq 1$ and $\forall^{\infty} n \left(h_n(1) > 0\right)$.

Assume $u \geq 1$ and n is large enough so that $h_n(1) > 0$. It is easy to check that $\frac{1}{u} \geq \frac{1}{s} + \epsilon(s,n)$ when $y < \frac{1}{2\epsilon(2u^2,n)}$. Let $S(n) = \sup_u \{h(u) > 0\}$. Let c be an arbitrary integer. We have
$$\forall^{\infty} n \left(n^c < \frac{1}{2\epsilon(n^{3c}, n)} \leq \frac{1}{2\epsilon(2n^{2c}, n)} \right),$$
hence $\forall^{\infty} n \left(n^c \leq S(n)\right)$. Therefore $S(n)$ is super polynomial and easy to check that statement 4 holds.

(4 ⟺ 5) Similar to 1⇒2 and 2⇒1.

(5 ⇒3) Assume 5 holds. Let $\varepsilon(n) = nS(n)^{-d}$. Clearly, condition 3 holds.

(3 $\Rightarrow$2) Observe that we can assume $t \geq 1$. Hence 3$\Rightarrow$2.
(7 $\iff$ 5)Obvious. □

Remark. The statements "for a super-polynomial S" can be replaced by "for all polynomial S".

Similarly, we have the following theorem of PMPT.

Theorem 4. *Let T be the ruining time of a machine M. The followings are equivalent,*

1. *M is PMPT.*
2. $\exists d > 0 \forall^\infty i \forall t > 0 \left(\Pr\left(T_i > t\right) < |i|\, t^{-d}\right)$.
3. $\exists d > 0 \forall^\infty i \left(E\left[T_i^d\right] \leq |i|\right)$.

These results are similar to well-known results in standard average-polynomial context. We omit the proof here.

3.5. **Stability.** MPT is invariant when the input is restricted on a noticeable subset.

Theorem 5. *If $M \in MPT(X, \mu)$ and Y is a noticeable subset ensemble of (X, μ), then $M \in MPT(Y, \mu|Y)$.*

Proof. Assume $\mu_i(Y_i) \geq |i|^{-k}$ for some k. Let f, g, ε be the witness of M being $\mu-$MPT, then $f(s\,|i|^k, |i|)$, $g(s\,|i|^k, |i|)$, ε are the witness of M being $\mu|Y-$MPT. □

If M is MPT and the change of input distribution has a "polynomial bound" for all "polynomial-time decidable subsets", then M is still MPT.

Theorem 6. *Let Let $\mu = \{\mu_i\}$ and $\nu = \{\nu_i\}$ are two series of probabilities defined on $X = \{X_i\}$ and M is $\mu-$CMPT. Assume there exist $d > 0$, e, such that for all $A \in SPT(\mathbb{N}, X)$ and $Y_i = \{x \in X_i : A(|i|, x) = 1\}$, we have*

$$\forall^\infty i \left(\nu_i(Y_i) \leq (\mu_i(Y_i))^d\, |i|^e\right).$$

Then M is $\nu-$CMPT.

Proof. (Sketch of the proof) We can assume $\frac{1}{d} = e = k \geq 1$.

We prove the special case that M is $\mu-$PMPT. let f be a witness of M being μ-PMPT. Then since for infinitely many i,

$$\begin{aligned}
\nu\left(T > f\left(s^k\,|i|^{k^2}, |i|\right)\right) &\leq \left(\mu\left(T > f\left(s^k\,|i|^{k^2}, |i|\right)\right)\right)^{\frac{1}{k}} |i|^k \\
&< \left(s^{-k}\,|i|^{-k^2}\right)^{\frac{1}{k}} |i|^k \\
&= \frac{1}{s},
\end{aligned}$$

$s, n \mapsto f(s^k n^{k^2}, n)$ is the witness of M being $\nu-$PMPT.

The CMPT case is similar but with slightly more computation. One way to prove it is using Theorem 3 (4). □

Fix an index set I and let U be the universal collection of all possible input on tape of machines. Denote by PolyDecidable the class of ensemble Y such that $Y_i = \{x \in U : A(|i|, i, x) = 1\}$ for an $A \in \mathrm{SPT}(\mathbb{N}, I, \{U_i = U\})$.

CMPT is also closed under some computational small change of the input.

Theorem 7. *Let X, Y be ensembles. If there exist a negligible function ε, such that*

$$\forall^{\infty} i \forall S \in \mathit{PolyDecidable}\left(\left|\Pr\left(X_i \in S_i\right) - \Pr\left(Y_i \in S_i\right)\right| \le \varepsilon(|i|)\right),$$

then $CMPT(X) = CMPT(Y)$.,

Proof. Let M be a machine and $T(x)$ be the running time of $M(x)$. Let $S_i = \{x \in U : T(x) > f(|i|, X)\}$ for some polynomial f. It is easy to verify that $\mathrm{CMPT}(X) = \mathrm{CMPT}(Y)$. □

Theorem 8. *If X, Y are semiseparable and $SPT * \Sigma$-indistinguishable ensembles, then $CMPT * \Sigma(X) = CMPT * \Sigma(Y)$.*

Proof. Let X, Y be semiseparable and $\mathrm{SPT} * \Sigma$-indistinguishable ensembles and $M * Z \in \mathrm{CMPT} * \Sigma(X)$.

Since X, Y are semiseparable, we can assume that $\langle X_i \rangle \le |i|^K$ and $\langle Y_i \rangle \le |i|^K$ for large enough i and

$$\Pr\left(\langle X_i \rangle < |i|\right) + \Pr\left(\langle Y_i \rangle < |i|\right) < \varepsilon(i)$$

for some negligible ε.

By Theorem 3(4), for large enough i,

$$\forall s < S(|i|) \left(\Pr\left(T_i^X > (s|i|)^k\right) < \frac{1}{s}\right)$$

for some super-polynomial S and k, where T^X is the running time of $M * z(X)$.

Define

$$U(n) = \inf\left\{s : \exists i \in I_n \left(\Pr\left(T_i^Y > \left(s^2 |i|^K\right)^k\right) \ge \frac{1}{s}\right)\right\},$$

where T^Y is the running time of $M * z(Y)$.

If U is super-polynomial, then By Theorem 3 (4), $(M, Z \times Y) \in \mathrm{CMPT}$. We are done.

Now assume U is not super-polynomial, we show that X, Y are not $\mathrm{SPT} * \Sigma$-indistinguishable.

Since U is not super-polynomial, we can find an $l \ge 0$, such that for infinitely many i, there is an

$$s \le |i|^l$$

and

$$\Pr\left(T_i^Y > \left(s^2 \langle Y_i \rangle\right)^k\right) \ge \Pr\left(T_i^Y > \left(s^2 |i|^K\right)^k\right) \ge \frac{1}{s}.$$

However, for large enough i, and for all

$$s \le \langle X_i \rangle^l,$$

we have

$$\Pr\left(T_i^X > \left(s^2 \langle X_i \rangle\right)^k\right) < \Pr\left(T_i^X > \left(s^2 |i|\right)^k\right) + \varepsilon(i) < \frac{1}{s^2} + \varepsilon(i).$$

Let $W \in \{X, Y\}$. Consider a machine D first takes auxiliary-input $Z \in \Sigma$ and input W. D sample $w_j \leftarrow W$ N times, where $N = 9L^3$, $L = \max(v+1, 2ku)$, and $v = \min\{\langle w_j\rangle\}, u = \max\{\langle w_j\rangle\}$. Since $\langle w_j\rangle \leq |i|^K$, N, L, v, u are all bounded by a polynomial.

Then let D simulates $M * Z(w_j)$ for all $1 \leq j \leq N$ and records the running steps t_j up to $u^k v^{2lk}$.

Fix some $1 < s < v^l$ and let T be the running time of $M * Z(W)$, by Hoeffding's inequality,

$$\Pr\left(\left|\Pr\left(T > \left(s^2 \langle W\rangle\right)^k\right) - \frac{\#\left\{t_j > \left(s^2 \langle w_j\rangle\right)^k\right\}}{N}\right| > c\right) < 2\exp\left(-2Nc^2\right).$$

Note that $\Pr(v \leq |i|) < N\varepsilon$Therefore, if $W = X$, we have

$$\Pr\left(\exists s \in \left(1, v^l\right) \frac{\#\left\{t_j > \left(s^2 \langle w_j\rangle\right)^k\right\}}{N} \geq \frac{1}{s^2} + \varepsilon + \frac{1}{3L}\right) < 2\exp(-2L) + N\varepsilon,$$

and if $W = Y$, then

$$\Pr\left(\exists s \in \left(1, v^l\right) \frac{\#\left\{t_j > \left(s^2 \langle w_j\rangle\right)^k\right\}}{N} < \frac{1}{s} - \frac{1}{3L}\right) < 2\exp(-2L) + N\varepsilon.$$

Since T is a positive integer, so there at most $u^k v^{2lk} N$ values of s need to check and the least one need to check is

$$\frac{\left(u^k+1\right)^{\frac{1}{2k}}}{u^{\frac{1}{2}}} = \left(1 + u^{-k}\right)^{\frac{1}{2k}}.$$

It is easy to check that

$$\frac{1}{s} - \frac{1}{s^2} \geq \frac{1}{L}$$

for all $s \in \left[\left(1+u^{-k}\right)^{\frac{1}{2k}}, v^l\right]$.

Let D output 1 iff

$$\exists s \in \left(1, v^l\right) \frac{\#\left\{t_j > \left(s^2 \langle w_j\rangle\right)^k\right\}}{N} \geq \frac{1}{s} - \frac{1}{3L}.$$

Clearly $D \in \mathrm{SPT}(Z \times W)$ and $\Pr(D * Z(X) = 1)$ is negligible while for infinitely many i, $\Pr(D * Z(Y) = 1) > \frac{1}{2}$. This contradicts to the assumption that X, Y are $\mathrm{SPT} * \Sigma$-indistinguishable. □

Let X be an ensemble and Σ a class of auxiliary input. Say that X is Σ-semiseparable if there is an $M \in \mathrm{SPT}(X \times \Sigma)$ and a polynomial p and negligible ε, such that for all $Z \in \Sigma$, we have $\Pr(M(X_i, Z_j) \leq p(|i|)) = 1$ and $\Pr(M(X_i, Z_j) < |i|) = \varepsilon(i)$.

Corollary 6. *Assume X, Y are Σ-semiseparable and $SPT * \Sigma$-indistinguishable ensembles. Then $CMPT * \Sigma(X) = CMPT * \Sigma(Y)$. In particular, $CMPT(X) = CMPT(Y)$.*

Proof. Assume $M \in \text{CMPT}(X \times Z)$ and let p, ε be a common witness of X, Y being Σ-semiseparable. Since $X \times Z$ and $Y \times Z$ are semiseparable and SPT-indistinguishable, by Theorem 8, we have $M \in \text{CMPT}(Y \times Z)$. Therefore, $\text{CMPT} * \Sigma(X) = \text{CMPT} * \Sigma(Y)$.

If we consider only the machine that ignore the information from auxiliary input, we have $\text{CMPT}(X) = \text{CMPT}(Y)$. □

Indistinguishabiliy is often study in a context that separability is usually not an issue.For example, theorem 8 implies that if X, Y are $\text{SPT} \times \mathbb{N}$-indistinguishable, then $\text{CMPT}\,(X) = \text{CMPT}\,(Y)$. In cryptography, the security parameter are usually not considered as a secret. Therefore, we are usually deal with separable input.

PMPT is not stable for indistinguishable change of input, see 4 for an example.

4. Zero-Knowledge And Simulation

4.1. **Definition.** Zero-knowledge proofs and arguments are powerful and widely used in cryptography. See [7] for a good survey on zero-knowledge. Let us first define the interactive argument system in our notion.

Definition 6. Let Γ be a class of machine, U be a set and $S \subset U$. Consider ensembles $X = \{X_i\}_{i \in S}$, $Y = \{Y_i\}_{i \in U \backslash S}$. A $\Gamma-$*interactive argument system* for X, Y is a two-party game between a machine *verifier* V, and another machine *prover* P, satisfying

- Efficiency: V is in $\Gamma(X)$ when interacting with P.
- Completeness: $\Pr(V \text{ accept } X) = 1$, when V interact with P.
- Soundness: for every P^* in $\Gamma(Y)/V$, $\Pr(V \text{ accept } Y)$ is negligible, when V interacts with P^*.

Above definition is a straightforward generalization of standard definition of (negligible soundness error) interactive argument system.

The standard definition can be interpreted as following. Let U be the universal set and $S \subseteq U$. There is a norm $|\cdot| : U \to \mathbb{N}$ defined on U. X is indexed by S and $X_s = (s, R)$, where R is the random tape. Similarly, Y is indexed by $U \backslash S$ and $Y_u = (u, R)$.

The prover has unbounded computation power, but in practice, it is usually a auxiliary-input machine $P = M * Z \in \Gamma * \Sigma$ where the secret Z_s helps $M \in \Gamma$ to convince V that $.s \in S$.

Ideally, V should learn nothing extra but the fact that $s \in S$. This concept, which is called zero-knowledge, can be formally defined as following.

Definition 7. A strategy A is $(\Gamma, \Gamma', \Sigma)-$*zero-knowledge from input* $X = \{X_i\}$ if for every interactive machine with auxiliary-input $B^* * Z \in \Gamma * \Sigma\,(X)\,/A$, there exists a simulator $C^* \in \Gamma'(Z \times X)$, such that the following two probability ensembles are Σ-indistinguishable:

(1) $(A(X), B^* * Z(X)) :=$ the output of $B^* * Z$ and interacting with A on common input X_i.
(2) $C^*(Z \times X)$:=the output of C^* on input $Z \times X$.

Call this $\Gamma * \Sigma$-zero-knowledge when $\Gamma = \Gamma'$.

An interactive argument system for X, Y is called *(auxiliary-input) zero-knowledge* if the prover strategy is auxiliary-input zero-knowledge on input from X.

This definition adapts the standard definition in our context with slightly generalization. Denote by PL the class of auxiliary-input with polynomial length of data. the standard definition of auxiliary-input constant-round zero-knowledge is equivalent to $(\mathrm{SPT}, \mathrm{EPT}, \mathrm{PL})$-zero-knowledge. There is a difference a priori between standard definition and $(\mathrm{SPT}, \mathrm{EPT}, \mathrm{PL})$-zero-knowledge. The standard definition considers only B^* such that

$$\forall Z \in \mathrm{PL}\,(B^* \in \mathrm{SPT} * Z(X))$$

for all $Z \in$ PL, while in our definition, we consider all $B^* * Z \in$ SPT*PL(X). However, it is easy to prove that these two definitions are equivalent. Moreover, similar to Corollary 3, because there is a polynomial bound of $B^* * Z$, there is no need to restrict the length of input in PL.

Since simulator can have more computing power than the adversary, it is arguable that we probably should call this definition $\Gamma' \backslash \Gamma$-knowledge, instead of zero-knowledge.

In the rest of this section, we consider the case $\Gamma = \Gamma' =$ CMPT.

4.2. **CMPT Simulator.** A common technique used in constructing simulators is called *rewind*. Simulators often use this method to trick the blackbox B^*, in order to get some information so that the simulator can use this information to construct a fake but valid transaction script.

The typical use of rewind works like the following. The simulator C^* makes a random query Y to B^* based on auxiliary-input Z. Y is generally *fake* (different than the real interaction with honest parties) but indistinguishable from a *genuine* one. If B^*does not response properly, then the simulation stops here. If B^* response properly, which usually reveals some important information, then C^* rewind B^* and use the newly learned information to make a *better* query X to B^* . This time, B^* may or may not return a valid response, the C^* will have to keep rewinding and querying B^* until it returns a valid response.

The running time of above algorithm depends on the probability of B^*returning a proper response. Let q_r be the probability of B^* having proper response with the fake query Y and p_z be the probability of B^*having proper response to the better query X, with auxiliary input z. The expectation of running time T_z is

$$E[T_z] = 1 + \frac{q_z}{p_z}.$$

The expected running time T is

$$E[T] = 1 + E[\frac{q_Z}{p_Z}].$$

If $p_z = q_z$, then $E[T] = 2$, thus the simulator is in PMPT. However, (Z, X) and (Z, Y) usually have different distributions. Fortunately, they are usually indistinguishable.

We have the following theorem.

Theorem 9. *Let Z be a semiseparable ensemble, R the random tape, and Σ a class of auxiliary-input. Let Q and $Q^{'}$ be two oracle ensembles and $B * \sigma$ a machine B with auxiliary input σ.*

Consider a machine $C_{B*\sigma}$ with input z from Z, which do the following algorithm,

(1) *Read an r from tape R.*
(2) *If $B * \sigma\,(Q(z, r))$=0, halt.*

(3) *Read next r from tape R.*

(4) *If $B*\sigma\left(Q^{'}(z,r)\right)=0$, go to 3. Otherwise, halt.*

If $(Z, B*\sigma(Q(Z\times R)))$and $(Z, B*\sigma(Q'(Z\times R)))$ are SPT-indistinguishable and

$$B*\sigma\in CMPT*\Sigma\left(Q(Z\times R)\sqcup Q^{'}(Z\times R)\right),$$

then $C_{B*\sigma}\in$CMPT$(Z\times R)$.

In particular, if $(Z, Q(Z\times R))$and $(Z, Q'(Z\times R))$ are Σ-indistinguishable for some Σ, then

$$C_{B*\sigma}\in CMPT(Z\times R)$$

for all

$$B*\sigma\in CMPT*\Sigma\left(Q(Z\times R)\sqcup Q^{'}(Z\times R)\right).$$

Proof. Assume $Q=Q^{'}$, then it is easy to check that $C_{B*\sigma}/B$ is in EPT, hence PMPT. Since

$$(Z\times R, B*\sigma(Q(Z\times R)), B*\sigma(Q'(Z\times R)))$$

and

$$(Z\times R, B*\sigma(Q(Z\times R)), B*\sigma(Q(Z\times R)))$$

are SPT-indistinguishable, by Theorem 8, $C_{B*\sigma}/B\in$ CMPT$(Z\times R)$.

By Theorem 2, it is easy to check that $C_{B*\sigma}\in$ CMPT$(Z\times R)$. □

Since $\frac{q_z}{p_z}$ can be quite arbitrary when they are noticeable, it is easy to construct examples so that there is no C in PMPT.

4.3. **An Example.** Following we propose a $CMPT$-zero-knowledge argument for equality of logarithm based on DDH (decisional Diffie-Hellman see [6]) assumption.

Protocol 1. *$CMPT$-zero-knowledge argument for equality of logarithm*

Common input: *$a, b, c=a^w, d=b^w\in G$, where G is a group.*

Private auxiliary input for the prover P**:** *$w=\log_a c=\log_b d$.*

V1 (Verifier's Step 1): *$l, m\underset{R}{\leftarrow}|G|$. Send $e=a^l b^m$ to P.*

P1 (Prover's Step 1): *$k\underset{R}{\leftarrow}|G|$. Send $f=a^k, g=e^{wk}$ to V.*

V2: *Send l, m to P.*

P2: *If $e\neq a^l b^m$, then stop. Otherwise, send k to V.*

V3: *If $a^k\neq f$ stop. If $g=\left(c^l d^m\right)^k$, then accept that $\log_a c=\log_b d$; otherwise, reject.*

The honest V and P run in SPT. The completeness of this argument is obvious. Since there are super polynomial many solutions l, m of $e=a^l b^m$, so even the P^* with unlimited computational power can not cheat. Thus the soundness also holds.

Let $G=\{G_i\}$ and $X_i=\left(g, g^a, g^b, g^{ab}\right)$ and $Y_i=\left(g, g^a, g^b, g^c\right)$ where $g\underset{R}{\leftarrow}G_i$ and $a, b, c\leftarrow|G_i|$. Recall that DDH assumption asserts that X, Y are SPT $*$ Σ-indistinguishable, where $\Sigma=$ PL or Σ is trivial, depends on context. By Theorem 2, DDH implies that X, Y are also CMPT$*\Sigma$-indistinguishable. Construct a simulator C^* with auxiliary input V^**z as following:

Algorithm 1. *Simulator for Protocol 1*

(1) *Simulate V1 of V^**z to get e.*

(2) *Send random* $k, f = a^k, g$ *to* $V^* * z$ *and simulate V2 to get* l, m.
(3) *If* $e \neq a^l b^m$, *then stop.*
(4) *Rewind to step P1 and send* $k', f = a^{k'}, g = \left(c^l d^m\right)^{k'}$ *to* $V^* * z$ *for a random* k'.
(5) *Simulate V2 and get* l, m *from* $V^* * z$.
(6) *If* $e \neq a^l b^m$, *then go to 4.*

By Theorem 9 and DDH assumption, C^* runs in CMPT whenever $V^* * z \in \mathrm{CMPT} * \Sigma$. It is easy to check that this simulator generates Σ-indistinguishable transcripts by DDH assumption.

In addition, it is very efficient and parallel-composition zero-knowledge.

4.4. **Indistinguishability and ZK.** Let J be a machine that can take unlimited sampling from X, Y for given security number n, each sampling count as one step. Denote by $J(n, \{X\}, \{Y\})$ the output of J.

In following theorem, we build a MPT machine which can "judge" whether two input ensembles X, Y are indistinguishable.

Theorem 10. *There is a* $J \in MPT$ *so that all ensembles* X, Y, $\Pr(J(n, \{X\}, \{Y\}) = 1)$ *is negligible iff* X, Y *is indistinguishable*

Proof. Let M_s be an enumeration of SPT machines so that the running time of M_s is bounded by n^s. Let $e : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ be a bijection that can be compute in SPT.

Define J do the following work.

```
(1) randomly select $S$ so that $\Pr(S \geq N) = \frac{1}{N}$.
(2) $i \leftarrow 0$.
(3) $(s, m) \leftarrow e(i)$
(4) if $n^s > S$ or $n^m > S$, go to 8.
(5) Let $\{x_j\}$ $\{y_j\}$ be $2n^{2m}n$ samples from $X_n$ and $Y_n$.
(6) Let $u = \frac{\#\{M_s(x_j)=1\}}{2n^m}$ and $v = \frac{\#\{M_s(y_j)=1\}}{2n^m}$.
(7) If $|u - v| > \frac{1}{n^m} - \frac{1}{n^{m+1}}$, output 1, halt.
(8) $i \leftarrow i + 1$
(9) if $i = n$, output 0, halt.
(10) Go to 3
```

It is easy to check that $\Pr\left(T_J > K_0 S^2 + K_1\right) < \frac{1}{S}$ for some constant K_0, K_1, hence $J \in \mathrm{MPT}$.

By Hoeffding inequality, it is easy to see that $\Pr\left(|u - v| > \frac{1}{n^m} - \frac{1}{n^{m+1}}\right)$ is negligible if

$$|P(M_s(X) = 1) - P(M_s(Y) = 1)| < \frac{1}{n^m}.$$

Therefore, if X, Y are indistinguishable, J outputs 0 except negligible probability.

Assume X, Y are not indistinguishable. Let $e^{-1}(s_0, m_0)$ be the smallest number so that

$$\exists^\infty n \, |P(M_{s_0}(X) = 1) - P(M_{s_0}(Y) = 1)| \geq \frac{1}{n^{m_0}}.$$

By Hoeffding inequality, $|u - v| > \frac{1}{n^{m_0}} - \frac{1}{n^{m_0+1}}$ when $i = (m_0, s_0)$ except negligible probability for infinite n. Since $\Pr(S \geq n^{s_0+m_0}) = \frac{1}{n^{s_0+m_0}}$, it is easy to check that $\Pr(J(n, \{X\}, \{Y\}) = 1) \geq \frac{1}{n^{s_0+m_0}} - \varepsilon$ for some negligible ε and infinite n. □

Theorem 1 ant its corollaries basically says that a hard problems is for SPT are also hard for MPT. One may wonder why it is easy to find a simulator in MPT but hard in SPT?

Roughly speaking, ZK is equivalent to the existence of a simulator which can generate indistinguishable transcript. However, there is no SPT "judge" for indistinguishibility. Therefore, we are unable to find an SPT machine J which can "judge" whether a machine S is a ZK-simulator for given verifier V^*. Therefore, the existence of simulator is not considered as a "problem" in the sense of Theorem 1 ant its corollaries.

5. Variations

In this section, we discuss several variations of our notion that can also be considered as efficient adversary in some sense.

5.1. **Total bound vs. asymptotic bound.** The polynomial bound in our notion is asymptotic. An obvious variation is to make the bound total. As shown in the setting of 4, $M \in \text{TOTAL-SPT}(X)$ is equivalent to $M \in \text{SPT}(\bigcup X)$. Similarly, $M \in$ TOTAL-PMPT(X)can be defined as $M \in \text{PMPT}(\bigcup X)$. However, naive definition of TOTAL-CMPT(X)

$$\forall i \forall s > 0 \left(\Pr\left(T_i > f(s, |i|)\right) < \frac{1}{s} + |g(s, |i|)|\, \varepsilon(|i|) \right)$$

is not equivalent to CMPT$(\bigcup X)$. This is because ε is negligible and negligible is an asymptotic notion. Thus, CMPT$(\bigcup X)$ is the right definition of TOTAL-CMPT(X).

Of course, $\bigcup X$ is not semiseparable. But on one hand, indistinguishability makes little sense when applied to $\bigcup X, \bigcup Y$; on the other hand, in our application, the security parameter is usually understood. In this sense, we should study CMPT$(\mathbb{N} \times \bigcup X)$ instead.

5.2. $\exists^\infty i$ **vs.** $\forall^\infty i$**.** In our definition, we assume that an adversary is efficient only if the distribution of it's running time is some what bounded *for all large enough* i. However, this assumption may be too strong. For example, if an encryption scheme can be defeated by an adversary that runs in SPT whenever the security parameter n is odd, then this encryption scheme needs some modification. Of course, in this example, we can simply modified the encryption scheme only works on even n. But in some cases, it may be infeasible to decide for which n the adversary can run in MPT. Let us call T *semi-CMPT* if there are polynomial f, g and negligible ε, such that,

$$\exists^\infty i \forall s > 0 \left(\Pr\left(T_i > f(s, |i|)\right) < \frac{1}{s} + |g(s, |i|)|\, \varepsilon(|i|) \right).$$

Many properties of CMPT can be adapted to semi-CMPT, for example for every set S and $M \in \text{Semi-CMPT}(X)$, there is a $M^* \in \text{SPT}(X)$, such that if $\Pr(M(X) \in S)$ is noticeable, then $\Pr(M^*(X) \in S)$ is noticeable.

We can treat semi-CMPT as a special case of CMPT as following. Enumerate (not necessarily computable) $n_1, n_2, \cdots$ for those n that satisfies

$$n = |i| \Rightarrow \forall s > 0 \left(\Pr\left(T_i > f(s, |i|)\right) < \frac{1}{s} + |g(s, |i|)|\, \varepsilon(|i|) \right).$$

Then we have $M \in \text{CMPT}\left(\{X_i |\, |i| \in \{n_0, n_1, \cdots\}\}\right)$

Since an adversary may run efficient in some n but the attack only success on other n, many definition in this paper would be tedious if we choose semi-CMPT to be our notion. Therefore, it is better to treat semi-CMPT as a special case of CMPT than the other way around.

5.3. **Sub-Super-Polynomial Time.**

Definition 8. Call an ensemble T *computational sub-super-polynomial* if for every super-polynomial $S : \mathbb{N} \to \mathbb{R}^+$, there is a negligible ε, such that for all i

$$\Pr\left(T_i > S\left(|i|\right)\right) < \varepsilon\left(|i|\right).$$

Call T *perfect sub-super-polynomial* if for all super polynomial S,

$$\forall^{\infty} i \left(\Pr\left(T_i > S\left(|i|\right)\right) = 0\right).$$

We define the class CSSPT (computational sub-super-polynomial time) and PSSPT (perfect sub-super-polynomial time) of machines in the manner similar to CMPT and PMPT. Observer that PSSPT = SPT, thus, CSSPT seems to be a straightforward computational generalization of SPT.

CSSPT is a larger class than CMPT.

Theorem 11. *$CMPT(X) \subseteq CSSPT(X)$.*

Proof. It is easy to check this from Theorem 3. □

In fact, observe that Theorem 3 (4) is equivalent of saying that

There is a polynomial f , such that for every essentially positive polynomial p,

$$\forall^{\infty} i \forall s \in \left(0, p\left(|i|\right)\right) \left(\Pr\left(T_i > f\left(s, |i|\right)\right) < \frac{1}{s}\right).$$

CSSPT can be viewed as a non-uniform generalization of CMPT.

Theorem 12. *Let M be a machine and X an input ensemble. The followings are equivalent*

1. *$M \in CSSPT(X)$.*
2. *For every essentially positive polynomial p, there is a polynomial f, such that*
$$\forall^{\infty} i \forall s \in \left(0, p\left(|i|\right)\right) \left(\Pr\left(T_i > f\left(s, |i|\right)\right) < \frac{1}{s}\right)$$

Proof. (not 1⇒not 2)

Assume 1 does not. Then there is a super-polynomial S and $k \in \mathbb{N}$,such that

$$\exists^{\infty} i \left(\Pr\left(T_i > S\left(|i|\right)\right) \geq \frac{1}{|i|^k}\right).$$

However, if 2 holds, then there is an $l \geq 1$, such that

$$\forall^{\infty} i \forall s \in \left(0, |i|^k + 1\right) \left(\Pr\left(T_i > \left(s\,|i|\right)^l\right) < \frac{1}{s}\right).$$

Hence,

$$\forall^{\infty} i \left(\Pr\left(T_i > |i|^{(k+1)l}\right) < \frac{1}{|i|^k}\right).$$

Contradiction.

Therefore, 2 does not hold.

(1⇒2)

Assume 1 holds. Define

$$u(s,n)=\max\left\{c\in\mathbb{N}|\exists i\in I_n\left(\Pr\left(T_i>c\right)\geq\frac{1}{s}\right)\right\}.$$

Let p be an essentially positive polynomial. Let $S\left(n\right)=u\left(p\left(n\right),n\right)$. Towards contradiction, assume that S does not bounded by any polynomial. Therefore, for some subset $W\subseteq\mathbb{N}$,$S|W$ is super-polynomial. Easy to see that by 1, for all $n\in W$,

$$\forall i\in I_n\left(\Pr\left(T_i>S\left(|i|\right)\right)<\varepsilon\left(|i|\right)\right).$$

However, for all $n\in\mathbb{N}$, there exists $i\in I_n$,such that

$$\begin{aligned}\Pr\left(T_i>S\left(|i|\right)\right) &= \Pr\left(T_i>u\left(p\left(n\right),n\right)\right)\\ &\geq \frac{1}{p\left(n\right)},\end{aligned}$$

contradiction.

Therefore, 2 must hold. □

The definition of CSSPT is pretty and clean. It is almost as good as CMPT. In fact, Theorem 1 and it's corollaries, Theorem 2 ant it's corollaries, Theorem, 5, Theorem 6, Theorem 8 holds when replace MPT or CMPT by CSSPT. We give sketch of proofs of CSSPT version of these theorems.

Proof. (CSSPT version of Theorem 1)Let

$$u(n)=\max\left\{c\in\mathbb{N}|\exists i\in I_n\left(\Pr\left(T_i>c\right)\geq\frac{1}{n^k}\right)\right\}.$$

If u is not bounded by a polynomial, then u is super-polynomial on some subset $W\subseteq\mathbb{N}$. Hence M is not CSSPT.

(CSSPT version of Theorem 2)

If M is not CSSPT, then for some super-polynomial u and some $k\in\mathbb{N}$

$$\exists^{\infty}i\left(\Pr\left(T_i>u\left(|i|\right)\right)\geq\frac{1}{|i|^k}\right).$$

However, since both $M^*=M/S$ and S are CSSPT, so there is a negligible ε, such that

$$\forall i\left(\Pr\left(\max\left(T_i^*,T_i^S\right)>\sqrt{u\left(|i|\right)}\right)<\varepsilon\left(|i|\right)\right).$$

Similar to the proof of Theorem 2, we have

$$2\varepsilon\left(|i|\right)>\Pr\left(T_i>u\left(|i|\right)\right)\geq\frac{1}{|i|^k},$$

contradiction.

(CSSPT version of Theorem, 5) Easy.

(CSSPT version of Theorem 6)

Let $m\in\mathbb{N}$ be an arbitrary number. Assume $\frac{1}{d}=e=k$. Then for every $l\in\mathbb{N}$, for large enough i, for all $s\in(0,|i|^m)$,

$$\nu\left(T_i>\left(s\,|i|\right)^l\right)\leq\mu\left(T_i>\left(s\,|i|\right)^l\right)^{\frac{1}{k}}|i|^{k^2}.$$

By Theorem 12, M is ν-CSSPT.

(CSSPT version of Theorem 8)

Since X, Y are semiseparable, we assume that

$$\max\left(\langle X_i\rangle, \langle Y_i\rangle\right) \le |i|^K .$$

Suppose $M * z \in \mathrm{CSSPT}(X) \backslash \mathrm{CSSPT}(Y)$. Then for some $m \in \mathbb{N}$, there is an $l \in \mathbb{N}$, for infinitely many $s_i \in (0, |i|^m)$, such that

$$\Pr\left(T_i^Y > \left(s_i^2 |i|^K\right)^l\right) \ge \frac{1}{s_i}$$

but

$$\Pr\left(T_i^X > \left(s_i^2 |i|^K\right)^l\right) < \frac{1}{s_i^2}.$$

Similar to the proof of Theorem 8, this leads to a contradiction. □

Moreover, if we use S to denote a super-polynomial function, ε a denote a negligible function and $k \ge 1$, we have the following list

SPT	$\forall S \forall^\infty i \left(\Pr\left(T_i > S\left(\lvert i\rvert\right)\right) = 0\right).$
SPT/NEG	$\exists \varepsilon \forall S \forall^\infty i \left(\Pr\left(T_i > S\left(\lvert i\rvert\right)\right) < \varepsilon\left(\lvert i\rvert\right)\right).$
PMPT	$\exists k \forall S \forall^\infty i \left(\Pr\left(T_i > S\left(\lvert i\rvert\right)\right) < S\left(\lvert i\rvert\right)^{-\frac{1}{k}}\right).$
CMPT	$\exists k \exists \varepsilon \forall S \forall^\infty i \left(\Pr\left(T_i > S\left(\lvert i\rvert\right)\right) < \max\left(S\left(\lvert i\rvert\right)^{-\frac{1}{k}}, \varepsilon\left(\lvert i\rvert\right)\right)\right).$
CSSPT	$\forall S \exists \varepsilon \forall^\infty i \left(\Pr\left(T_i > S\left(\lvert i\rvert\right)\right) < \varepsilon\left(\lvert i\rvert\right)\right).$

Where SPT/NEG is the class of machines that except for a negligible probability, the running time is bounded by a polynomial. This result is relatively easy to prove, we omit the proofs.

6. Remarks and Conclusion

6.1. **PMPT is intrinsic to computation model.** In computation theory,we study many different abstract models of computation model that are Turing complete and have same class of SPT. The definition of efficient computing should be intrinsic to these model of computation. SPT is intrinsic to model of computation, but EPT is not. By Theorem 2 and it's corollaries, MPT is intrinsic the model of computation.

Moreover, PMPT is *the smallest class that intrinsic to computation model contains EPT* in the following sense.

Let $M \in \mathrm{PMPT}(X)$ and X is indexed by $\mathbb{N}$. Assume that M has one working tape t_0 and one random tape. We would like to show that M can be accelerated to EPT after some reasonable hardware modification (SPT accelerations).

By Theorem 4, there is an $N \in \mathbb{N}$, such that $\forall^\infty n E\left[T_n^{-\frac{1}{N}}\right] \le n$. The new hardware has extra tapes t_1, t_2, initially blank and 3 new instructions: copy a tape, clear a tape, B. The new instruction B is defined as following.

1. Simulate M up to step l^N, where the data on t_2 is 1^l.
2. If M halt before step l^N, clear t_2.

The hardware runs M in the following way,

1. Copy t_0 to t_1.
2. Add 1 on t_2. Copy t_1 to t_0. Run $B(M, t_0, t_2)$.
3. If t_2 is blank, then clear t_1 and halt.

(4) Go to Step 2.

New instructions including copy a tape, clear a tape, and B can be simulated in SPT. Assume every instruction takes one step, M can run on this new hardware in EPT.

6.2. **CMPT is stable to indistinguishable change of input.** Since in most cases, security parameter is not a secret when study indistinguishability, in this subsection, we assume all inputs are separable.

Consider a M a machine and X, Y be some input ensembles. If $M(X)$ is efficient and $M(Y)$ is not, then we can tell X from Y by their running time. Therefore, if X and Y are indistinguishable, then $M(X)$and $M(Y)$ should have same efficiency. Otherwise, either

(1) indistinguishable ensembles are not really indistinguishable, or
(2) we can not practically tell an efficient from an inefficient machine.

However, neither SPT or EPT is stable to indistinguishable change of input. By Theorem 8, CMPT is.

In addition, CMPT is the smallest such class that contains PMPT. Consider an arbitrary $M \in \mathrm{CMPT}(X)$. By Theorem 3 (5), we can make an negligible modification on input X so that $T_i < S(|i|)$ for the modified input. It is straightforward to verify that M is PMPTon the modified input.

6.3. **Summary.** If we decide to study ZK, then choosing CMPT as the notion of efficiency is only the logical consequence.

In 1.2.3, we argue that it is natural to treat input as probabilistic object when study cryptographic protocols. Then as we point out in 6.2, running time of a machine should be indistinguishable on indistinguishable inputs. Since efficiency ought to be a concrete notion, it should be stable to indistinguishable change of input.

6.1 and 6.2 implies that CMPT is the smallest class that contains EPT, is intrinsic to computation model and is stable to indistinguishable change of input. Since EPT simulator is essential, it is natural to study CMPT. If we accept EPT as efficient, then MPT is just a theoretic friendly EPT.

Moreover, Theorem 9 shows that it is easy to construct simulators using the standard rewind technique. Even the indistinguishability becomes a more concrete notion in MPT (see 4.4).

Theorem 1 and it's corollaries guarantee the backward compatibility of MPT. Hence we can still use standard assumptions like trap-door function, one way function and practical assumptions like DDH, CDH and DLOG.

Therefore, as we claim in 1.3.2, CMPT is a good model of efficient adversary. Even if you don't like the idea of treating MPT as the notion of efficient adversary, you can still extent the class of adversaries from SPT to MPT, prove the security result and then restrict the result to SPT adversary.

To be fair, in order to prove $M(X) \in \mathrm{MPT}$, you probably need to find out what the distribution of X first. It is not always that easy to do so. For example, the "submachine input distribution" in Theorem 2 might be difficult to determined. One may argue that this new notion only move the hard works form one place to another. However, we think it is the way it should be if we want to treat input as distribution. Since SPT is not stable to indistinguishable change of input, one may think that we should study "SPT except negligible" instead. However, this notion lost many nice properties that SPT has. Therefore, the composition theorems of "SPT except negligible" have to consider some sort of "submachine input distribution" like we did.

In fact, the notion MPT lead us to a new protocol and simulator in 4.3. The simulator is not in EPT and we don't know whether it is ZK in usual sense. However, our results guarantees that using this protocol as a building block is practically as secure as using any other ZK argument.

We think this paper present an elegant and intuitive approach for studying the security of practical protocols.

References

[1] B. Barak. How to go beyond the black-box simulation barrier. *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 106–115, 2001.

[2] B. Barak and O. Goldreich. Universal arguments and their applications. *Computational Complexity, 2002. Proceedings. 17th IEEE Annual Conference on*, pages 162–171, 2002.

[3] B. Barak and Y. Lindell. Strict polynomial-time in simulation and extraction. *SICOMP: SIAM Journal on Computing*, 33, 2004.

[4] B. Barak, Y. Lindell, and S. Vadhan. Lower bounds for non-black-box zero knowledge. *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 384–393, 2003.

[5] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the theory of average case complexity. *Journal of Computer and System Sciences*, 44:193–219, 1992.

[6] Dan Boneh. The decision diffie-hellman problem. *Lecture Notes in Computer Science*, 1423:48–63, 1998.

[7] O. Goldreich. Zero-knowledge twenty years after its invention. *Electronic Colloquium on Computational Complexity (http://www. eccc. uni-trier. de/eccc/), Report*, 63, 2002.

[8] O. Goldreich. On expected probabilistic polynomial-time adversaries: A suggestion for restricted definitions and their benefits. *Lecture Notes In Computer Science*, 4392:174, 2007.

[9] R. Impagliazzo. A personal view of average-case complexity. *10th Annual Structure in Complexity Theory Conference*, pages 134–147.

[10] J. Katz and Y. Lindell. Handling expected polynomial-time strategies in simulation-based security proofs. *2nd Theory of Cryptography Conf*, 2005.