Computational Irrelevancy: Bridging the Gap between Pseudo- and Real Randomness in MPC Protocols

Nariyasu Heseri [*]   Koji Nuida [†]

Abstract

Due to the fact that classical computers cannot efficiently obtain random numbers, it is common practice to design cryptosystems in terms of real random numbers and then replace them with (cryptographically secure) pseudorandom ones for concrete implementations. However, as pointed out by [10], this technique may lead to compromise of security in secure multiparty computation (MPC) protocols. Although this work suggests using information-theoretically secure protocols and pseudorandom generators (PRGs) with high min-entropy to alleviate the problem, yet it is preferable to base the security on computational assumptions rather than the stronger information-theoretic ones. By observing that the contrived constructions in the aforementioned work use MPC protocols and PRGs that are closely related to each other, we notice that it may help to alleviate the problem by using protocols and PRGs that are ”unrelated” to each other. In this paper, we propose a notion called ”computational irrelevancy” to formalise the term ”unrelated” and under this condition provide a security guarantee under computational assumptions.

Keywords: secure multiparty computation, MPC, pseudorandom generators, PRG, relativisation

1 Introduction

1.1 Background

It is a widely known fact that classical computers are not able to generate random numbers. When necessary, random numbers are generated from noise of the environment, OS statistics, or user inputs etc. However, in most cryptosystems where very long random bit sequences are required, these random sources are not efficient enough to generate them. To this end, pseudorandom generators (PRGs) are used to expand a short real random bit sequence into a long one that looks random.

Under the observation that if the use of PRGs compromises security of a cryptosystem then the cryptosystem can be modified to a distinguisher against the PRGs, one may naïvely believe that when a cryptographically secure PRG is used in a secure cryptosystem, the resulting system is also secure. However, this naïve reduction only works in settings where the seeds are not explicitly known to the adversaries, and as pointed out by [10], the security definition of secure multiparty computation (MPC) protocols (in the semi-honest model) forms a counterexample of this. Indeed, protocol-PRG pairs are explicitly constructed by [10] such that the protocol is secure itself but becomes insecure when the PRG is used.

Since it has become so common a paradigm in cryptography to design cryptosystems in terms of real random numbers and use the output of PRGs for concrete implementations, it is urgent to find ways to avoid such problems. While it is proved in [10] that using PRGs with very high min-entropy can help avoid the problem provided that the original protocol is information-theoretically secure, it is believed that achieving information-theoretic security for all parties is very hard and with severe limitations. For example, it is shown in [3] that, in terms of boolean functions, information-theoretic security for majority of the participating parties is achievable for only a limited subset of boolean functions. Therefore, instead of using information-theoretically secure MPC protocols, it is more desirable to ensure security in terms of computationally secure ones.

By taking a close look at the constructions in [10], it is easy to observe that these contrived constructions use MPC protocols and PRGs that are closely related to each other. One may develop an intuition that it helps to alleviate the problem to use PRGs “unrelated” to the MPC protocol. We propose a

[*] Graduate School of Information Science and Technology, The University of Tokyo, nariyasu@g.ecc.u-tokyo.ac.jp

[†] Kyushu University / AIST, nuida@imi.kyushu-u.ac.jp

notion called “computational irrelevancy”, which utilises what is called “relativisation” in the literature of complexity theory, to formalise the term “unrelated”, and under computational irrelevancy conditions provide a security guarantee under computational assumptions.

1.2 Our Contributions

In this paper, we define computational irrelevancy in terms of MPC protocols and PRGs and investigate the sufficient conditions under which security is preserved when PRGs are used.

Computational irrelevancy conditions are defined in two aspects:

- between the protocol and PRGs used, and
- between PRGs used by different parties.

Since [10] explicitly constructed examples where the first type of irrelevancy does not hold, we can easily see that the first one is necessary to preserve security. In contrast, the work only considered the case where 1 PRG is used, thus examples where the second condition does not hold and use of PRGs compromises security have not been explicitly constructed. We discovered the necessity of the second condition only during our proof of security on the resulting protocol. That being said, we note that assuming the second type of irrelevancy is natural and intuitive: with similar contrived examples where different parties use closely related PRGs, adversaries may utilise this fact to recover part of the information intended for only honest parties.

We use a paradigm called “relativisation” ([1]) to formalise computational irrelevancy, which is intensely studied in the literature of complexity theory. As an informal description, an MPC protocol or PRG is considered computationally irrelevant from a PRG if security is preserved even if the corresponding distinguisher is given oracle access to the inverter of the latter. Constructing such protocols and PRGs that are (computationally) secure relative to a family of inverters of other PRGs apparently requires some computational problems that are hard even with access to some family of oracles. A class of problems called “the gap-problems”, proposed by [11], can be considered a class of computational problems that are hard relative to an oracle solving the corresponding decision problem. This class of problems proved to be very useful and cryptographic schemes have been constructed and security of existing schemes has been proved under the computational hardness assumptions of these problems (e.g. [11, 8, 7]). In addition, the relativisation paradigm has been used to prove some negative results in the literature of cryptography (e.g. [6]). Hence here we argue that such relativised computational problems are interesting in their own right and security or computational hardness assumptions relative to a family of the inverters of some PRGs, which are essential for the concrete implementations of our proposed sufficient conditions, are hopefully further studied in future works.

Unlike [10] which only considered 2-party protocols where only 1 PRG is used, we consider a broader range of situations:

I 1 adversary (or multiple non-colluding adversaries) exists and 1 PRG is used.

II 1 adversary (or multiple non-colluding adversaries) exists and multiple PRGs are used.

III Multiple colluding adversaries exist and multiple PRGs are used.

Deferring the precise presentation of the sufficient conditions to preserve security for each case, here we briefly and informally summarise the computational irrelevancy conditions used in each case:

I 1 adversary, 1 PRG:

- Protocol is irrelevant from PRG.

II 1 adversary, multiple PRGs:

- Protocol is irrelevant from *each* PRG.
- PRGs are *pairwise* irrelevant.

III multiple adversaries, multiple PRGs:

- Protocol is irrelevant from PRGs.
- PRGs are irrelevant.

By comparing conditions used in each case, we find that these results are very intuitive: in the first case where only 1 PRG is used, the only possible requirement is that the protocol be irrelevant from the PRG; in the second case where only 1 adversary exists or adversaries do not collude, computational irrelevancy is imposed on the PRGs one-by-one; in the third case where adversaries may collude, computational irrelevancy is imposed on the PRGs as a whole.

2 Preliminaries

In this section, we introduce some basic notations as well as security definitions of PRGs and MPC protocols in the traditional sense.

2.1 Basic Notations

Definition 1. In this paper, we use $\mathcal{PTM}$, $\mathcal{PPT}$ and $\mathcal{NUPPT}$ to denote the set of (uniform) probabilistic algorithms, (uniform) probabilistic polynomial-time algorithms and non-uniform probabilistic polynomial-time algorithms respectively. For a finite set S, we write $s \leftarrow_R S$ to denote that s is assigned a uniformly sampled value from the set S.

Definition 2. A function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if for any polynomial p, $\exists \lambda_0 \in \mathbb{N}$, $\forall \lambda > \lambda_0$, $f(\lambda) < \frac{1}{p(\lambda)}$. A function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *noticeable* if there exists a polynomial p and $\lambda_0 \in \mathbb{N}$, $\forall \lambda > \lambda_0$, $f(\lambda) \geq \frac{1}{p(\lambda)}$.

2.2 Pseudorandom Generators

In this section we review the definition of PRGs and their security definition as well as introduce notations about PRGs for later use.

Definition 3. A deterministic polynomial-time algorithm is called a *pseudorandom generator (PRG)* if on input $(1^\lambda, s)$ outputs r such that $\lambda \in \mathbb{N}$, $s, r \in \{0,1\}^*$ and $|r| > |s|$. $\lambda \in \mathbb{N}$ is called the security parameter. s is called the seed. $l_{in}(\lambda) := |s|$ and $l_{out}(\lambda) := |r|$ are called the input length and output length respectively. When multiple PRGs are used, we use $l_{in}(\lambda, i)$ and $l_{out}(\lambda, i)$ to denote the input length and output length of the PRG indexed by i.

Definition 4. A PRG $\mathcal{R}$ is said to be *uniformly (resp. non-uniformly) secure* if $\forall \mathcal{D} \in \mathcal{PPT}$ (resp. $\mathcal{NUPPT}$), the advantage

$$\mathsf{Adv}_{\mathcal{R},\mathcal{D}}(\lambda) := \left| \Pr\left[\mathcal{D}(1^\lambda, \mathcal{R}(1^\lambda, s))\right] - \Pr\left[\mathcal{D}(1^\lambda, r)\right] \right|$$

is negligible, where $s \leftarrow_R \{0,1\}^{l_{in}(\lambda)}$ and $r \leftarrow_R \{0,1\}^{l_{out}(\lambda)}$.

Definition 5. For a PRG $\mathcal{R}$, let $\mathcal{I}_\mathcal{R}$ denote the $\mathcal{PTM}$ speficied in Algorithm 1 that inverts $\mathcal{R}$'s output:

Algorithm 1 $\mathcal{I}_\mathcal{R} \in \mathcal{PTM}$ inverting $\mathcal{R}$

```
procedure $\mathcal{I}_\mathcal{R}(1^\lambda, r)$                    ▷ $r \in \{0,1\}^{l_{out}(1^\lambda)}$
    $S \leftarrow \emptyset$
    for $s \leftarrow \{0,1\}^{l_{in}(\lambda)}$ do
        if $\mathcal{R}(1^\lambda, s) = r$ then
            $S \leftarrow S \cup \{s\}$
        end if
    end for
    if $S \neq \emptyset$ then
        $s \leftarrow_R S$
        return $s$
    else
        return $\perp$
    end if
end procedure
```

2.3 Secure Multiparty Computation

In this section, we review security definitions for MPC protocols. While the notion of secure multiparty computation was first conceived and formalised by Yao ([12, 13]), the modern formalisation of security of MPC protocols that is used in more recent literature was proposed by [5]. In this paper, we shall deal with the semi-honest adversarial model in [5].

Definition 6. Let π be an n-party protocol and $\vec{f} = (f_1, f_2, \ldots, f_n)$ be a probabilistic functionality to be computed by π. We say π is secure against party $\mathcal{P}_i$ if $\exists \mathcal{S} \in \mathcal{PPT}$, $\forall \mathcal{D} \in \mathcal{NUPPT}$,

$$\left|\Pr\left[\mathcal{D}\left(\mathcal{S}(1^\lambda, x_i, f_i(\vec{x})), \vec{f}(\vec{x})\right) = 1\right] - \Pr\left[\mathcal{D}\left(x_i, r_i, \vec{m}_i(1^\lambda, \vec{x}; \vec{r}), \pi(1^\lambda, \vec{x}; \vec{r})\right) = 1\right]\right|$$

is negligible where

- x_i: input of party $\mathcal{P}_i$ and $\vec{x} := (x_1, x_2, \ldots, x_n)$.
- $f_i(\vec{x})$: output of party $\mathcal{P}_i$ on inputs $\vec{x}$ and $\vec{f}(\vec{x}) := (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_n(\vec{x}))$.
- r_i: random bits used by $\mathcal{P}_i$ and $\vec{r} := (r_1, r_2, \ldots, r_n)$.
- $\vec{m}_i(1^\lambda, \vec{x}; \vec{r})$: messages received by $\mathcal{P}_i$ during the executing of the protocol with inputs $\vec{x}$ and randomness $\vec{r}$ and $\vec{m}(1^\lambda, \vec{x}; \vec{r}) := (\vec{m}_1(1^\lambda, \vec{x}; \vec{r}), \vec{m}_2(1^\lambda, \vec{x}; \vec{r}), \ldots, \vec{m}_n(1^\lambda, \vec{x}; \vec{r}))$.

Definition 7. Let π be an n-party protocol and $\vec{f} = (f_1, f_2, \ldots, f_n)$ be a probabilistic functionality to be computed by π. We say π is secure if $\exists \mathcal{S} \in \mathcal{PPT}$, $\forall I = \{i_1, i_2, \ldots, i_m\} \subset \{1, 2, \ldots, n\}$ $(i_1 < i_2 < \cdots < i_m)$, $\forall \mathcal{D} \in \mathcal{NUPPT}$,

$$\left|\Pr\left[\mathcal{D}\left(\mathcal{S}(1^\lambda, I, \vec{x}_I, \vec{f}_I(\vec{x})), \vec{f}(\vec{x})\right) = 1\right] - \Pr\left[\mathcal{D}\left(\mathsf{VIEW}_I(\vec{x}; \vec{r}), \pi(1^\lambda, \vec{x}; \vec{r})\right) = 1\right]\right|$$

is negligible where

$$\begin{aligned}
\vec{x}_I &:= (x_{i_1}, x_{i_2}, \ldots, x_{i_m}) \\
\vec{f}_I(\vec{x}) &:= (\vec{f}_{i_1}(\vec{x}), \vec{f}_{i_2}(\vec{x}), \ldots, \vec{f}_{i_m}(\vec{x})) \\
\mathsf{VIEW}_I(\vec{x}; \vec{r}) &:= (I, \mathsf{VIEW}_{i_1}(\vec{x}; \vec{r}), \mathsf{VIEW}_{i_2}(\vec{x}; \vec{r}), \ldots, \mathsf{VIEW}_{i_m}(\vec{x}; \vec{r})) \\
\mathsf{VIEW}_{i_k}(\vec{x}; \vec{r}) &:= (x_{i_k}, r_{i_k}, \vec{m}_{i_k}(1^\lambda, \vec{x}; \vec{r})).
\end{aligned}$$

3 Main Theorems

3.1 1 Adversary, 1 PRG

We first consider the case where only 1 adversary exists or multiple non-colluding adversaries exist and only 1 PRG is used. As stated before, if the adversary does not use a PRG thus the seed is not explicitly known to the adversary, then we can perform a standard reduction to guarantee security unconditionally. We present this in the following lemma for completeness.

Lemma 8. *Let π be an n-party protocol, and $\mathcal{R}$ be a PRG. Let $i, j \in \{1, 2, \ldots, n\}$ and $i \neq j$. If π is secure against party $\mathcal{P}_j$ and $\mathcal{R}$ is non-uniformly secure, then $\pi \circ_i \mathcal{R}$ is secure against $\mathcal{P}_j$. Here $\pi \circ_i \mathcal{R}$ denotes the protocol derived by replacing party $\mathcal{P}_i$'s randomness with the output of $\mathcal{R}$.*

Proof. Let $\mathcal{S}$ be the simulator for party $\mathcal{P}_j$ by the security of π. We show that this simulator can also be used to prove the security of $\pi \circ_i \mathcal{R}$. For any distinguisher $\mathcal{D}$ against $\mathcal{S}$ and any input $\vec{x}$,

$$\begin{aligned}
&\left|\Pr\left[\mathcal{D}(x_j, r_j, (\overrightarrow{m \circ_i \mathcal{R}})_j(1^\lambda, \vec{x}), \pi \circ_i \mathcal{R}(1^\lambda, \vec{x})) = 1\right] - \Pr\left[\mathcal{D}(\mathcal{S}(1^\lambda, x_j, f_j(\vec{x})), \vec{f}(\vec{x})) = 1\right]\right| \\
= &\left|\Pr\left[\mathcal{D}(x_j, r_j, \vec{m}_j(1^\lambda, \vec{x}; \left\langle\mathcal{R}(1^\lambda, s_i)\right\rangle_i), \pi(1^\lambda, \vec{x}; \left\langle\mathcal{R}(1^\lambda, s_i)\right\rangle_i)) = 1\right] - \Pr\left[\mathcal{D}(\mathcal{S}(1^\lambda, x_j, f_j(\vec{x})), \vec{f}(\vec{x})) = 1\right]\right| \\
\leq &\left|\Pr\left[\mathcal{D}(x_j, r_j, \vec{m}_j(1^\lambda, \vec{x}; \langle r_i\rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i\rangle_i)) = 1\right] - \Pr\left[\mathcal{D}(\mathcal{S}(1^\lambda, x_j, f_j(\vec{x})), \vec{f}(\vec{x})) = 1\right]\right| \\
&\quad + \Big|\Pr\left[\mathcal{D}(x_j, r_j, \vec{m}_j(1^\lambda, \vec{x}; \left\langle\mathcal{R}(1^\lambda, s_i)\right\rangle_i), \pi(1^\lambda, \vec{x}; \left\langle\mathcal{R}(1^\lambda, s_i)\right\rangle_i)) = 1\right] \\
&\qquad\qquad - \Pr\left[\mathcal{D}(x_j, r_j, \vec{m}_j(1^\lambda, \vec{x}; \langle r_i\rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i\rangle_i)) = 1\right]\Big|
\end{aligned}$$

where $r_i \leftarrow_R \{0,1\}^{l_{out}(\lambda)}$ and $s_i \leftarrow_R \{0,1\}^{l_{in}(\lambda)}$. Here $(\overrightarrow{m \circ_i \mathcal{R}})_j$ denotes the messages received by $\mathcal{P}_j$ during the execution of the $\pi \circ_i \mathcal{R}$ (what $\vec{m}_j$ is to π) and $\langle r \rangle_i$ in the randomness part of the inputs means that party $\mathcal{P}_i$ takes randomness r and others take uniformly distributed random bits (as specified in π). The first term on the right hand side is negligible by the security of π against $\mathcal{P}_j$, while the second is negligible by the non-uniform security of $\mathcal{R}$. ■

Next we define security of MPC protocols relative to a family of oracles to formalise computational irrelevancy between the protocol and PRGs.

Definition 9. Let π be an n-party protocol and $i \in \{1, 2, \ldots, n\}$. We say π is secure against party $\mathcal{P}_i$ *relative to* $\mathcal{O} = \{\mathcal{O}_j\}_{j \in I} \subset \mathcal{PTM}$ if $\exists \mathcal{S} \in \mathcal{PPT}$, $\forall \mathcal{D} \in \mathcal{NUPPT}$,

$$\left| \Pr\left[\mathcal{D}^{\mathcal{O}}\left(\mathcal{S}(1^\lambda, x_i, f_i(\vec{x})), \vec{f}(\vec{x})\right) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{O}}\left(x_i, r_i, \vec{m}_i(1^\lambda, \vec{x}; \vec{r}), \pi(1^\lambda, \vec{x}; \vec{r})\right) = 1\right] \right|$$

is negligible where $\mathcal{D}^{\mathcal{O}}$ means $\mathcal{D}$ is given oracle access to all $\mathcal{O}_i$.

The following is necessary as an additional assumption on PRGs.

Definition 10. Let $\mathcal{R}$ be a PRG. Let $\mathrm{range}(\mathcal{R}, \lambda)$ denote the set of all $\mathcal{R}$'s outputs under security parameter λ:

$$\mathrm{range}(\mathcal{R}, \lambda) := \left\{ \mathcal{R}(1^\lambda, s) \mid s \in \{0,1\}^{l_{in}(\lambda)} \right\}.$$

We say $\mathcal{R}$ is *uniformly (resp. non-uniformly) indistinguishable in its range relative to* $\mathcal{O} = \{\mathcal{O}_i\}_{i \in I} \subset \mathcal{PTM}$ if $\forall \mathcal{D} \in \mathcal{PPT}$ (resp. $\mathcal{NUPPT}$),

$$\left| \Pr\left[\mathcal{D}^{\mathcal{O}}\left(1^\lambda, r\right) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{O}}\left(1^\lambda, \mathcal{R}(1^\lambda, s)\right) = 1\right] \right|$$

is negligible where $r \leftarrow_R \mathrm{range}(\mathcal{R}, \lambda)$ and $s \leftarrow_R \{0,1\}^{l_{in}(\lambda)}$.

The following is necessary as an additional assumption on the simulator of the protocol. This basically states that the simulator outputs its own random bits as is to generate the adversary's random tape. Note that this definition is introduced by [10] and proved to be necessary also in the setting of information-theoretic security.

Definition 11 ([10])**.** Let π be an n-party protocol that is secure against $\mathcal{P}_i$ with simulator $\mathcal{S}$. We say $\mathcal{S}$ is with *raw randomness* if $\exists \mathcal{T} \in \mathcal{PPT}$, $\forall \lambda \in \mathbb{N}$,

$$\mathcal{S}(1^\lambda, x_i, f_i(\vec{x}); r_i, \tau_i) = \left\langle r_i, \mathcal{T}(1^\lambda, x_i, f_i(\vec{x}), r_i; \tau_i) \right\rangle$$

where the notation $\langle r_i, y \rangle$ means that components of the tuple (r_i, y) are rearranged such that r_i corresponds to the simulated random tape part.

Next we present the main theorem of this section, which states the sufficient conditions under which use of a PRG preserves security in the 1-adversary-1-PRG case.

Theorem 12. *Let π be an n-party protocol. For $i \in \{1, 2, \ldots, n\}$ and any PRG $\mathcal{R}$ (whose output length matches that of the random tape of $\mathcal{P}_i$), if*

- *π is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}}$ with raw randomness where $\mathcal{I}_{\mathcal{R}}$ is the inverter specified in Definition 5.*
- *$\epsilon_1(\lambda) := \frac{|\mathrm{range}(\mathcal{R}, \lambda)|}{2^{l_{out}(\lambda)}}$ is noticeable.*
- *$\mathcal{R}$ is non-uniformly indistinguishable in its range relative to $\mathcal{I}_{\mathcal{R}}$.*

then $\pi \circ_i \mathcal{R}$ is secure against party $\mathcal{P}_j$ for $j \in \{1, 2, \ldots, n\}$. In particular, $\pi \circ_i \mathcal{R}$ is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}}$ with raw randomness.

Proof. For $j \neq i$, the result follows directly from Lemma 8. We consider security against party $\mathcal{P}_i$.

Let $\mathcal{S}$ be the simulator for π by the first assumption. Since $\mathcal{S}$ is with raw randomness, we write $\mathcal{S}(1^\lambda, x_i, f_i(\vec{x}); r_i, \tau_i) = \left\langle r_i, \mathcal{T}(1^\lambda, x_i, f_i(\vec{x}), r_i; \tau_i) \right\rangle$. Consider the simulator $\tilde{\mathcal{S}}$ defined as

$$\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i, \tau_i) := \left\langle s_i, \mathcal{T}(1^\lambda, x_i, f_i(\vec{x}), \mathcal{R}(1^\lambda, s_i); \tau_i) \right\rangle.$$

For any distinguisher $\tilde{\mathcal{D}}$ (with oracle access to $\mathcal{I}_{\mathcal{R}}$) against $\tilde{\mathcal{S}}$, define distinguisher $\mathcal{D}$ (with oracle access to $\mathcal{I}_{\mathcal{R}}$) against $\mathcal{S}$ as follows.

Algorithm 2 Distinguisher $\mathcal{D}$ against $\mathcal{S}$

```
1: procedure $\mathcal{D}^{\mathcal{I}_{\mathcal{R}}}(x_i^\dagger, r_i^\dagger, \vec{m}_i^\dagger, y_i^\dagger)$
2:     $s_i^\dagger \leftarrow \mathcal{I}_{\mathcal{R}}(r_i^\dagger)$
3:     if $s_i^\dagger \neq \perp$ then
4:         return $\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i^\dagger, s_i^\dagger, \vec{m}_i^\dagger, y_i^\dagger)$
5:     else
6:         return 0
7:     end if
8: end procedure
```

Since $\mathcal{S}$ is with raw randomness, $r_i^\dagger$ is distributed identically in both the real and the simulated views. Now the condition $s_i^\dagger \neq \perp$ at line 3 holds with probability $\epsilon_1(\lambda)$ in either case. Under this condition, since $\mathcal{R}(1^\lambda, s_i^\dagger) = r_i^\dagger$ and by raw randomness of $\mathcal{S}$ and the definition of $\tilde{\mathcal{S}}$, we have

$$\Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}}}(x_i, r_i, \vec{m}_i(1^\lambda, \vec{x}; \langle r_i \rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i \rangle_i)) = 1\right] = \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i^\dagger, m_i(1^\lambda, \vec{x}; \langle r_i \rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i \rangle_i)) = 1\right]$$

$$\Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}}}(\mathcal{S}(1^\lambda, x_i, f_i(\vec{x}); r_i, \tau_i), \vec{f}(\vec{x})) = 1\right] = \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i^\dagger, \tau_i), \vec{f}(\vec{x})) = 1\right]$$

where $r_i \leftarrow_R \{0,1\}^{l_{out}(\lambda)}$ and $s_i^\dagger \leftarrow \mathcal{I}_{\mathcal{R}}(1^\lambda, r_i)$. Here $\langle r \rangle_i$ in the randomness part of the inputs means that party $\mathcal{P}_i$ takes randomness r and others take uniformly distributed random bits (as specified in π).

Claim 12.1. *For any input $\vec{x}$, under the condition $s_i^\dagger \neq \perp$ at line 3 in Algorithm 2, both*

$$\begin{aligned}
\epsilon_2(\lambda, \vec{x}) := \Big| &\Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i^\dagger, \vec{m}_i(1^\lambda, \vec{x}; \langle r_i \rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i \rangle_i)) = 1\right] \\
&- \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i, \vec{m}_i(1^\lambda, \vec{x}; \langle \mathcal{R}(1^\lambda, s_i) \rangle_i), \pi(1^\lambda, \vec{x}; \langle \mathcal{R}(1^\lambda, s_i) \rangle_i)) = 1\right] \Big| \\
\epsilon_3(\lambda, \vec{x}) := \Big| &\Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i^\dagger, \tau_i), \vec{f}(\vec{x})) = 1\right] - \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i, \tau_i), \vec{f}(\vec{x})) = 1\right] \Big|
\end{aligned}$$

are negligible, where $r_i \leftarrow_R \{0,1\}^{l_{out}(\lambda)}$, $s_i^\dagger \leftarrow \mathcal{I}_{\mathcal{R}}(1^\lambda, r_i)$ and $s_i \leftarrow_R \{0,1\}^{l_{in}(\lambda)}$.

Proof. We first show that $\epsilon_2(\lambda, \vec{x})$ is negligible. Assume the negation, i.e. $\exists p \in \text{poly}(\cdot)$, there exists infinitely many $(\lambda, \vec{x})$'s such that $\epsilon_2(\lambda, \vec{x}) \geq \frac{1}{p(\lambda)}$. Now consider a distinguisher $\tilde{D}^*$ with oracle access to $\mathcal{I}_{\mathcal{R}}$ against $\mathcal{R}$. Pick one $\vec{x}$ for each λ and give $\tilde{\mathcal{D}}^*$ as advice for security parameter λ. Define $\tilde{\mathcal{D}}^*$ as follows.

Algorithm 3 Distinguisher $\tilde{\mathcal{D}}^*$ against $\mathcal{R}$

```
procedure $\tilde{\mathcal{D}}^{*\mathcal{I}_{\mathcal{R}}}(1^\lambda, r^\dagger)$        ▷ with $\vec{x}$ as advice
    $s \leftarrow \mathcal{I}_{\mathcal{R}}(r^\dagger)$
    if $s \neq \perp$ then
        Simulate $\pi$ on input $\vec{x}$ and use $r^\dagger$ as random tape for party $\mathcal{P}_i$ to obtain $\mathcal{P}_i$'s view $\vec{m}_i$ and the result $\vec{y}$.
        return $\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s, \vec{m}_i, \vec{y})$
    else
        return 0
    end if
end procedure
```

Under the condition $s_i^\dagger \neq \perp$ at line 3 in Algorithm 2, $r_i \leftarrow_R \{0,1\}^{l_{out}(\lambda)}$ is equivalent to $r_i \leftarrow_R$

range$(\mathcal{R}, \lambda)$. Now

$$\begin{aligned}
&\left|\Pr\left[\tilde{\mathcal{D}}^{*\mathcal{I}_{\mathcal{R}}}(1^\lambda, r_i) = 1\right] - \Pr\left[\tilde{\mathcal{D}}^{*\mathcal{I}_{\mathcal{R}}}(1^\lambda, \mathcal{R}(1^\lambda, s_i)) = 1\right]\right| \\
={}& \left|\Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i^\dagger, \vec{m}_i(1^\lambda, \vec{x}; \langle r_i \rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i \rangle_i)) = 1\right]\right. \\
&\left. - \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i, \vec{m}_i(1^\lambda, \vec{x}; \left\langle \mathcal{R}(1^\lambda, s_i) \right\rangle_i), \pi(1^\lambda, \vec{x}; \left\langle \mathcal{R}(1^\lambda, s_i) \right\rangle_i)) = 1\right]\right| \\
={}& \epsilon_2(\lambda, \vec{x}) > \frac{1}{p(\lambda)}.
\end{aligned}$$

This contradicts the assumption that $\mathcal{R}$ is non-uniformly indistinguishable in its range relative to $\mathcal{I}_{\mathcal{R}}$.

The proof for ϵ_3 is similar to that of ϵ_2, where instead of simulating π, the distinguisher $\tilde{\mathcal{D}}^*$ against $\mathcal{R}$ simulates $\tilde{\mathcal{S}}$ using the input as its random tape. □

Let

$$\begin{aligned}
P_1(\lambda, \vec{x}) &:= \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i^\dagger, \vec{m}_i(1^\lambda, \vec{x}; \langle r_i \rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i \rangle_i)) = 1\right] \\
P_2(\lambda, \vec{x}) &:= \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i, \vec{m}_i(1^\lambda, \vec{x}; \left\langle \mathcal{R}(1^\lambda, s_i) \right\rangle_i), \pi(1^\lambda, \vec{x}; \left\langle \mathcal{R}(1^\lambda, s_i) \right\rangle_i)) = 1\right] \\
Q_1(\lambda, \vec{x}) &:= \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i^\dagger, \tau_i), \vec{f}(\vec{x})) = 1\right] \\
Q_2(\lambda, \vec{x}) &:= \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i, \tau_i), \vec{f}(\vec{x})) = 1\right]
\end{aligned}$$

thus ϵ_2 and ϵ_3 become

$$\begin{aligned}
\epsilon_2(\lambda, \vec{x}) &= |P_1(\lambda, \vec{x}) - P_2(\lambda, \vec{x})| \\
\epsilon_3(\lambda, \vec{x}) &= |Q_1(\lambda, \vec{x}) - Q_2(\lambda, \vec{x})|.
\end{aligned}$$

Summarising the discussion above, under the condition $s_i^\dagger \neq \perp$ at line 3 in Algorithm 2,

$$\begin{aligned}
&\left|\Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}}}(x_i, r_i, \vec{m}_i(1^\lambda, \vec{x}; \langle r_i \rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i \rangle_i)) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}}}(\mathcal{S}(1^\lambda, x_i, f_i(\vec{x}); r_i, \tau_i), \vec{f}(\vec{x})) = 1\right]\right| \\
={}& \left|\Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i^\dagger, \vec{m}_i(1^\lambda, \vec{x}; \langle r_i \rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i \rangle_i)) = 1\right] - \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i^\dagger, \tau_i), \vec{f}(\vec{x})) = 1\right]\right| \\
={}& |P_1(\lambda, \vec{x}) - Q_1(\lambda, \vec{x})| \\
={}& |(P_2(\lambda, \vec{x}) - Q_2(\lambda, \vec{x})) + (P_1(\lambda, \vec{x}) - P_2(\lambda, \vec{x})) - (Q_1(\lambda, \vec{x}) - Q_2(\lambda, \vec{x}))| \\
\geq{}& |P_2(\lambda, \vec{x}) - Q_2(\lambda, \vec{x})| - |P_1(\lambda, \vec{x}) - P_2(\lambda, \vec{x})| - |Q_1(\lambda, \vec{x}) - Q_2(\lambda, \vec{x})| \\
={}& \left|\Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i, (\overrightarrow{m \circ_i \mathcal{R}})_i(1^\lambda, \vec{x}; \langle s_i \rangle_i), \pi \circ_i \mathcal{R}(1^\lambda, \vec{x}; \langle s_i \rangle_i)) = 1\right]\right. \\
&\left. - \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i, \tau_i), \vec{f}(\vec{x})) = 1\right]\right| - \epsilon_2(\lambda, \vec{x}) - \epsilon_3(\lambda, \vec{x}).
\end{aligned}$$

Note that $\mathcal{D}^{\mathcal{I}_{\mathcal{R}}}$ always outputs 0 when the condition $s_i^\dagger \neq \perp$ at line 3 in Algorithm 2 is not satisfied. Thus as for *overall* probability, we have

$$\begin{aligned}
&\left|\Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}}}(x_i, r_i, \vec{m}_i(1^\lambda, \vec{x}; \langle r_i \rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i \rangle_i)) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}}}(\mathcal{S}(1^\lambda, x_i, f_i(\vec{x}); r_i, \tau_i), \vec{f}(\vec{x})) = 1\right]\right| \\
\geq{}& \left(\left|\Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(x_i, s_i, (\overrightarrow{m \circ_i \mathcal{R}})_i(1^\lambda, \vec{x}; \langle s_i \rangle_i), \pi \circ_i \mathcal{R}(1^\lambda, \vec{x}; \langle s_i \rangle_i)) = 1\right]\right.\right. \\
&\left.\left. - \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_{\mathcal{R}}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i, \tau_i), \vec{f}(\vec{x})) = 1\right]\right| - \epsilon_2(\lambda, \vec{x}) - \epsilon_3(\lambda, \vec{x}) \right) \cdot \epsilon_1(\lambda)
\end{aligned}$$

which implies that

$$\begin{aligned}
&\left|\Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_\mathcal{R}}(x_i, s_i, (\overrightarrow{m \circ_i \mathcal{R}})_i(1^\lambda, \vec{x}; \langle s_i\rangle_i), \pi \circ_i \mathcal{R}(1^\lambda, \vec{x}; \langle s_i\rangle_i)) = 1\right]\right. \\
&\qquad \left. - \Pr\left[\tilde{\mathcal{D}}^{\mathcal{I}_\mathcal{R}}(\tilde{\mathcal{S}}(1^\lambda, x_i, f_i(\vec{x}); s_i, \tau_i), \vec{f}(\vec{x})) = 1\right]\right| \\
\leq\ & \frac{\left|\Pr\left[\mathcal{D}^{\mathcal{I}_\mathcal{R}}(x_i, r_i, \vec{m}_i(1^\lambda, \vec{x}; \langle r_i\rangle_i), \pi(1^\lambda, \vec{x}; \langle r_i\rangle_i)) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{I}_\mathcal{R}}(\mathcal{S}(1^\lambda, x_i, f_i(\vec{x}); r_i, \tau_i), \vec{f}(\vec{x})) = 1\right]\right|}{\epsilon_1(\lambda)} \\
&+ \epsilon_2(\lambda, \vec{x}) + \epsilon_3(\lambda, \vec{x}).
\end{aligned}$$

The first term is negligible by the assumptions that π is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_\mathcal{R}$ and that ϵ_1 is bounded below by the inverse of a polynomial. $\epsilon_2(\lambda, \vec{x})$ and $\epsilon_3(\lambda, \vec{x})$ are negligible by Claim 12.1. Therefore the expression is negligible, which completes the proof of Theorem 12. ■

3.2 1 Adversary, Multiple PRGs

We extend the result to the case where multiple parties use PRGs.

Since now we are dealing with multiple PRGs, we have to first formalise computational irrelevancy between PRGs. It turns out that in this case, the protocol is only required to be irrelevant from each PRG and PRGs are pairwise irrelevant.

Definition 13. Let $\mathcal{R}$ be a PRG. We say $\mathcal{R}$ is *uniformly (resp. non-uniformly) secure relative to* $\mathcal{O} = \{\mathcal{O}_i\}_{i\in I} \subset \mathcal{PTM}$ if $\forall \mathcal{D} \in \mathcal{PPT}$ (resp. $\mathcal{NUPPT}$),

$$\left|\Pr\left[\mathcal{D}^{\mathcal{O}}\left(1^\lambda, r\right) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{O}}\left(1^\lambda, \mathcal{R}(1^\lambda, s)\right) = 1\right]\right|$$

is negligible where $r \leftarrow_R \{0,1\}^{l_{out}(\lambda)}$ and $s \leftarrow_R \{0,1\}^{l_{in}(\lambda)}$.

Definition 14. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be uniformly (resp. non-uniformly) secure PRGs. We say $\mathcal{R}_1$ and $\mathcal{R}_2$ are *computationally irrelevant* if for $i \in \{1,2\}$, $\mathcal{R}_i$ is uniformly (resp. non-uniformly) secure relative to $\mathcal{I}_{\mathcal{R}_{3-i}}$. For a family of uniformly (resp. non-uniformly) secure PRGs $\{\mathcal{R}_i\}_{i\in I}$, we say $\{\mathcal{R}_i\}_{i\in I}$ are *pairwise computationally irrelevant* if $\forall i, j \in I$ with $i \neq j$, $\mathcal{R}_i$ and $\mathcal{R}_j$ are computationally irrelevant.

Using pairwise computationally irrelevant PRGs and with the help of Theorem 12, it is relatively easy to derive the main theorem of this section, which states the sufficient conditoins to preserve security for the 1-adversary-multiple-PRG case.

Theorem 15. *Let π be an n-party protocol and $I \subset \{1, 2, \ldots, n\}$. Let $\{\mathcal{R}_i\}_{i\in I}$ be a family of non-uniformly secure PRGs that are pairwise computationally irrelevant. If $\forall i \in I$,*

- *π is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}_i}$ with raw randomness.*
- *$\frac{|\mathrm{range}(\mathcal{R}_i,\lambda)|}{2^{l_{out}(\lambda,i)}}$ is noticeable.*
- *$\mathcal{R}_i$ is non-uniformly indistinguishable in its range relative to $\mathcal{I}_{\mathcal{R}_i}$.*

then $\pi \circ_I \{\mathcal{R}_i\}_{i\in I}$ is secure against all *parties. In particular, $\forall i \in I$, $\pi \circ_I \{\mathcal{R}_j\}_{j\in I}$ is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}_i}$ with raw randomness. Here $\pi \circ_I \{\mathcal{R}_i\}_{i\in I}$ denotes the protocol derived by replacing party $\mathcal{P}_i$'s randomness with the output of $\mathcal{R}_i$ for all $i \in I$ in π.*

Proof. For $i \notin I$, the security of $\pi \circ_I \{\mathcal{R}_j\}_{j\in I}$ against $\mathcal{P}_i$ can be easily derived by applying Lemma 8 $|I|$ times.

Consider the case $i \in I$. By Theorem 12, it suffices to show that $\pi \circ_{I\setminus\{i\}} \{\mathcal{R}_j\}_{j\in I\setminus\{i\}}$ is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}_i}$ with raw randomness. We show this in the next claim.

Claim 15.1. *$\forall i \in I$, $\pi \circ_{I\setminus\{i\}} \{\mathcal{R}_j\}_{j\in I\setminus\{i\}}$ is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}_i}$ with raw randomness.*

Proof. Let $I = \{i, i_1, i_2, \ldots, i_{m-1}\}$ $(2 \leq m \leq n, i_1 < i_2 < \cdots < i_{m-1})$. By assumption, π is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}_i}$ with raw randomness. Let $\mathcal{S}$ be such a simulator and we use it to prove the security of $\pi \circ_{I\setminus\{i\}} \{\mathcal{R}_j\}_{j\in I\setminus\{i\}}$. For any distinguisher $\mathcal{D}$ with oracle access to $\mathcal{I}_{\mathcal{R}_i}$ against $\mathcal{S}$,

$$
\begin{aligned}
&\left|\Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}_i}}(x_i, r_i, (\overrightarrow{m \circ_{I\setminus\{i\}} \{\mathcal{R}_j\}_{j\in I\setminus\{i\}}})_i(1^\lambda, \vec{x}), (\pi \circ_{I\setminus\{i\}} \{\mathcal{R}_j\}_{j\in I\setminus\{i\}})(1^\lambda, \vec{x})) = 1\right]\right.\\
&\qquad \left. - \Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}_i}}(\mathcal{S}(1^\lambda, x_i, f_i(\vec{x})), \vec{f}(\vec{x})) = 1\right]\right|\\
\leq &\left|\Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}_i}}(x_i, r_i, \vec{m}_i(1^\lambda, \vec{x}), \pi(1^\lambda, \vec{x})) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}_i}}(\mathcal{S}(1^\lambda, x_i, f_i(\vec{x})), \vec{f}(\vec{x})) = 1\right]\right|\\
&+ \sum_J \left|\Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}_i}}(x_i, r_i, (\overrightarrow{m \circ_{J\cup\{i_{|J|+1}\}} \{\mathcal{R}_j\}_{j\in J\cup\{i_{|J|+1}\}}})_i(1^\lambda, \vec{x}),\right.\right.\\
&\qquad\qquad \left.(\pi \circ_{J\cup\{i_{|J|+1}\}} \{\mathcal{R}_j\}_{j\in J\cup\{i_{|J|+1}\}})(1^\lambda, \vec{x})) = 1\right]\\
&\qquad \left. - \Pr\left[\mathcal{D}^{\mathcal{I}_{\mathcal{R}_i}}(x_i, r_i, (\overrightarrow{m \circ_J \{\mathcal{R}_j\}_{j\in J}})_i(1^\lambda, \vec{x}), (\pi \circ_J \{\mathcal{R}_j\}_{j\in J})(1^\lambda, \vec{x})) = 1\right]\right|
\end{aligned}
$$

where J spans over $\emptyset, \{i_1\}, \{i_1, i_2\}, \ldots, \{i_1, i_2, \ldots, i_{m-2}\}$. Note that $\pi \circ_J \{\mathcal{R}_j\}_{j\in J} = \pi$ when $J = \emptyset$, and $\pi \circ_{J\cup\{i_{|J|+1}\}} \{\mathcal{R}_j\}_{j\in J\cup\{i_{|J|+1}\}} = \pi \circ_{I\setminus\{i\}} \{\mathcal{R}_j\}_{j\in I\setminus\{i\}}$ when $J = \{i_1, i_2, \ldots, i_{m-2}\}$. By pairwise irrelevancy, each $\mathcal{R}_{i_{|J|+1}}$ is non-uniformly secure relative to $\mathcal{I}_{\mathcal{R}_i}$, thus each summand is negligible. The first term is negligible by the assumption that π is secure against $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}_i}$. Hence the whole expression is negligible, which completes the proof. □

Now $\pi \circ_{I\setminus\{i\}} \{\mathcal{R}_j\}_{j\in I\setminus\{i\}}$ satisfies the first condition of Theorem 12 by Claim 15.1, and $\mathcal{R}_i$ satisfies the second and third conditions of Theorem 12 by assumption. We can conclude that $\pi \circ_I \{\mathcal{R}_j\}_{j\in I}$ is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}_i}$ with raw randomness. ■

3.3 Multiple Adversaries, Multiple PRGs

We extend the result to the case where multiple colluding adversaries exist and multiple parties use PRGs.

Due to the difference of security definitions between the 1-adversary and multiple-adversary cases, we first extend the security of an MPC protocol relative to a family of oracles to the case where multiple parties are corrupted and may collude.

Definition 16. Let π be an n-party protocol. We say π is secure *relative to* $\mathcal{O} = \{\mathcal{O}_I\}_{I\subset\{1,2,\ldots,n\}}$ *where* $\forall I \subset \{1, 2, \ldots, n\}$, $\mathcal{O}_I = \{\mathcal{O}_i\}_{i\in J_I} \subset \mathcal{PTM}$ if $\exists \mathcal{S} \in \mathcal{PPT}$, $\forall I = \{i_1, i_2, \ldots, i_m\} \subset \{1, 2, \ldots, n\}$ $(i_1 < i_2 < \cdots < i_m)$, $\forall \mathcal{D} \in \mathcal{NUPPT}$,

$$\left|\Pr\left[\mathcal{D}^{\mathcal{O}_I}\left(\mathcal{S}(1^\lambda, I, \vec{x}_I, \vec{f}_I(\vec{x})), \vec{f}(\vec{x})\right) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{O}_I}\left(\mathsf{VIEW}_I(\vec{x}; \vec{r}), \pi(1^\lambda, \vec{x}; \vec{r})\right) = 1\right]\right|$$

is negligible.

Similarly, the definition of raw randomness (Definition 11) can also be extended.

Definition 17. Let π be a secure n-party protocol with simulator $\mathcal{S}$. We say $\mathcal{S}$ is with *raw randomness* if $\exists \mathcal{T} \in \mathcal{PPT}$, $\forall \lambda \in \mathbb{N}$, $\forall I = \{i_1, i_2, \ldots, i_m\} \subset \{1, 2, \ldots, n\}$ $(i_1 < i_2 < \cdots < i_m)$,

$$\mathcal{S}(1^\lambda, I, \vec{x}_I, \vec{f}_I(\vec{x}); r_{i_1}, r_{i_2}, \ldots, r_{i_m}, \tau) = \left\langle r_{i_1}, r_{i_2}, \ldots, r_{i_m}, \mathcal{T}(1^\lambda, I, \vec{x}_I, \vec{f}_I(\vec{x}), r_{i_1}, r_{i_2}, \ldots, r_{i_m}; \tau)\right\rangle$$

where the notation $\langle r_{i_1}, r_{i_2}, \ldots, r_{i_m}, y\rangle$ means that components of the tuple $(r_{i_1}, r_{i_2}, \ldots, r_{i_m}, y)$ are rearranged such that each r_{i_k} corresponds to the simulated random tape part.

Unlike the 1-adversary case where it is sufficient to consider the PRGs one-by-one, we define overall computational irrelevancy for families of PRGs.

Definition 18. For a family of uniformly (resp. non-uniformly) secure PRGs $\{\mathcal{R}_i\}_{i\in I}$, we say $\{\mathcal{R}_i\}_{i\in I}$ are *computationally irrelevant* if $\forall i \in I$, $\mathcal{R}_i$ is uniformly (resp. non-uniformly) secure relative to $\left\{\mathcal{I}_{\mathcal{R}_j}\right\}_{j\in I\setminus\{i\}}$.

Now that we have extended the necessary definitions in the 1-adversary cases, the main theorem of this section presenting a similar result can be shown for the case where multiple colluding adversaries exist.

Theorem 19. *Let π be an n-party protocol and $I \subset \{1,2,\dots,n\}$. Let $\{\mathcal{R}_i\}_{i\in I}$ be a family of non-uniformly secure PRGs that are computationally irrelevant. If*

- *π is secure relative to $\left\{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}\right\}_{J\subset\{1,2,\dots,n\}}$ with raw randomness.*
- *$\forall i \in I$, $\epsilon_{1,i}(\lambda) := \frac{|\mathrm{range}(\mathcal{R}_i,\lambda)|}{2^{l_{out}(\lambda,i)}}$ is noticeable.*
- *$\forall i \in I$, $\mathcal{R}_i$ is non-uniformly indistinguishable in its range relative to $\left\{\mathcal{I}_{\mathcal{R}_j}\right\}_{j\in I}$.*

then $\pi \circ_I \{\mathcal{R}_i\}_{i\in I}$ is secure relative to $\left\{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}\right\}_{J\subset\{1,2,\dots,n\}}$ with raw randomness.

Proof. The argument is essentially the same as the one used in the proof of Theorem 12, combined with a hybrid argument.

Let $\mathcal{S}$ be the simulator for π by the first assumption. Take $\forall J = \{j_1, j_2, \dots, j_m\} \subset \{1,2,\dots,n\}$ ($j_1 < j_2 < \dots < j_m$). Since $\mathcal{S}$ is with raw randomness, we write

$$\mathcal{S}(1^\lambda, J, \vec{x}_J, f_J(\vec{x}); r_{j_1}, r_{j_2}, \dots, r_{j_m}, \tau) = \left\langle r_{j_1}, r_{j_2}, \dots, r_{j_m}, \mathcal{T}(1^\lambda, J, \vec{x}_J, f_J(\vec{x}), r_{j_1}, r_{j_2}, \dots, r_{j_m}; \tau)\right\rangle .$$

Consider the simulator $\tilde{\mathcal{S}}$ defined as

$$\begin{aligned}&\tilde{\mathcal{S}}(1^\lambda, J, \vec{x}_J, \vec{f}_J(\vec{x}); s_{j_1}, s_{j_2}, \dots, s_{j_m}, \tau)\\ :=& \left\langle s_{j_1}, s_{j_2}, \dots s_{j_m}, \mathcal{T}(1^\lambda, J, x_i, f_i(\vec{x}), \mathcal{R}^*_{j_1}(1^\lambda, s_{j_1}), \mathcal{R}^*_{j_2}(1^\lambda, s_{j_2}), \dots, \mathcal{R}^*_{j_m}(1^\lambda, s_{j_m}); \tau)\right\rangle\end{aligned}$$

where

$$\mathcal{R}^*_{j_k}(1^\lambda, s_{j_k}) := \begin{cases}\mathcal{R}_{j_k}(1^\lambda, s_{j_k}) & \text{if } j_k \in I \cap J\\ s_{j_k} & \text{if } j_k \notin I \cap J.\end{cases}$$

For any distinguisher $\tilde{\mathcal{D}}$ (with oracle access to $\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}$) against $\tilde{\mathcal{S}}$, define distinguisher $\mathcal{D}$ (with oracle access to $\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}$) against $\mathcal{S}$ as follows.

Algorithm 4 Distinguisher $\mathcal{D}$ against $\mathcal{S}$

```
 1: procedure $\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(J^\dagger, \mathsf{VIEW}^\dagger_{j_1}, \mathsf{VIEW}^\dagger_{j_2}, \dots, \mathsf{VIEW}^\dagger_{j_m}, y^\dagger_i)$
 2:     for $j_k \in J^\dagger$ do
 3:         $(x^\dagger_{j_k}, r^\dagger_{j_k}, \vec{m}^\dagger_{j_k}) \leftarrow \mathsf{VIEW}^\dagger_{j_k}$
 4:         if $j_k \in I$ then
 5:             $s^\dagger_{j_k} \leftarrow \mathcal{I}_{\mathcal{R}_{j_k}}(r^\dagger_{j_k})$
 6:             if $s^\dagger_{j_k} = \perp$ then
 7:                 return 0
 8:             end if
 9:         else
10:             $s^\dagger_{j_k} \leftarrow r^\dagger_{j_k}$
11:         end if
12:         $\mathsf{VIEW}^{\dagger\dagger}_{j_k} \leftarrow (x^\dagger_{j_k}, s^\dagger_{j_k}, \vec{m}^\dagger_{j_k})$
13:     end for
14:     return $\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(J^\dagger, \mathsf{VIEW}^{\dagger\dagger}_{j_1}, \mathsf{VIEW}^{\dagger\dagger}_{j_2}, \dots, \mathsf{VIEW}^{\dagger\dagger}_{j_m}, y^\dagger_i)$
15: end procedure
```

Since $\mathcal{S}$ is with raw randomness, $r^\dagger_{j_k}$'s are distributed identically in both the real and the simulated views. Now in each iteration of the for-loop, under the condition that $s^\dagger_{j_k} = \perp$ at line 6 gets evaluated, it evaluates to false with probability $\epsilon_{1,j_k}(\lambda)$ in either case. Since if $j_k \notin I$, line 7 is never executed, line 14 gets executed with probability $\prod_{j_k\in I\cap J} \epsilon_{1,j_k}(\lambda)$ in the real view as well as in the simulated view conditioned on $J = J^\dagger$. Under this condition, since $\mathcal{R}^*_{j_k}(1^\lambda, s^\dagger_{j_k}) = r^\dagger_{j_k}$ and by raw randomness of $\mathcal{S}$ and the definition of $\tilde{\mathcal{S}}$, we have

$$\begin{aligned}
&\Pr\left[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\big(\mathsf{VIEW}_J(\vec{x};\vec{r}),\pi(1^\lambda,\vec{x};\vec{r})\big)=1\right]\\
&=\Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(J,(x_{j_1},s^\dagger_{j_1},m_{j_1}(1^\lambda,\vec{x};\langle r_{j_k}\rangle_{j_k\in J})),\\
&\qquad\ldots,(x_{j_m},s^\dagger_{j_m},m_{j_m}(1^\lambda,\vec{x};\langle r_{j_k}\rangle_{j_k\in J})),\pi(1^\lambda,\vec{x};\langle r_{j_k}\rangle_{j_k\in J}))=1\Big]\\
&\Pr\left[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(\mathcal{S}(1^\lambda,J,\vec{x}_J,\vec{f}_J(\vec{x});r_{j_1},\ldots,r_{j_m},\tau),\vec{f}(\vec{x}))=1\right]\\
&=\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(\tilde{\mathcal{S}}(1^\lambda,J,\vec{x}_J,\vec{f}_J(\vec{x});s^\dagger_{j_1},\ldots,s^\dagger_{j_m},\tau),\vec{f}(\vec{x}))=1\right]
\end{aligned}$$

where $r_{j_k}\leftarrow_R\{0,1\}^{l_{out,j_k}(\lambda)}$ and

$$s^\dagger_{j_k}\leftarrow\begin{cases}\mathcal{I}_{\mathcal{R}_{j_k}}(1^\lambda,r_{j_k}) & \text{if } j_k\in I\cap J\\ r_{j_k} & \text{if } j_k\notin I\cap J.\end{cases}$$

Here $\langle r_{j_k}\rangle_{j_k\in J}$ in the randomness part of the inputs means that each party $\mathcal{P}_{j_k},\forall j_k\in J$ takes randomness r_{j_k} and others take uniformly distributed random bits (as specified in π).

Claim 19.1. *For any input $\vec{x}$ and $J=\{j_1,j_2,\ldots,j_m\}\subset\{1,2,\ldots,n\}$ $(j_1<j_2<\cdots<j_m)$, under the condition that line 7 is never executed in the real view and the simulated view conditioned on $J=J^\dagger$, both*

$$\begin{aligned}
\epsilon_2(\lambda,\vec{x},J):=\Big|&\Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(J,(x_{j_1},s^\dagger_{j_1},m_{j_1}(1^\lambda,\vec{x};\langle r_{j_k}\rangle_{j_k\in J})),\\
&\qquad\ldots,(x_{j_m},s^\dagger_{j_m},m_{j_m}(1^\lambda,\vec{x};\langle r_{j_k}\rangle_{j_k\in J})),\pi(1^\lambda,\vec{x};\langle r_{j_k}\rangle_{j_k\in J}))=1\Big]\\
&-\Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(J,(x_{j_2},s_{j_1},m_{j_1}(1^\lambda,\vec{x};\left\langle\mathcal{R}^*_{j_k}(1^\lambda,s_{j_k})\right\rangle_{j_k\in J})),\\
&\qquad\ldots,(x_{j_m},s_{j_m},m_{j_m}(1^\lambda,\vec{x};\left\langle\mathcal{R}^*_{j_k}(1^\lambda,s_{j_k})\right\rangle_{j_k\in J})),\pi(1^\lambda,\vec{x};\left\langle\mathcal{R}^*_{j_k}(1^\lambda,s_{j_k})\right\rangle_{j_k\in J}))=1\Big]\Big|\\
\epsilon_3(\lambda,\vec{x},J):=\Big|&\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(\tilde{\mathcal{S}}(1^\lambda,J,\vec{x}_J,\vec{f}_J(\vec{x});s^\dagger_{j_1},\ldots,s^\dagger_{j_m},\tau),\vec{f}(\vec{x}))=1\right]\\
&-\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(\tilde{\mathcal{S}}(1^\lambda,J,\vec{x}_J,\vec{f}_J(\vec{x});s_{j_1},\ldots,s_{j_m},\tau),\vec{f}(\vec{x}))=1\right]\Big|
\end{aligned}$$

are negligible, where $r_{j_k}\leftarrow_R\{0,1\}^{l_{out,j_k}(\lambda)}$,

$$s^\dagger_{j_k}\leftarrow\begin{cases}\mathcal{I}_{\mathcal{R}_{j_k}}(1^\lambda,r_{j_k}) & \textit{if } j_k\in I\cap J\\ r_{j_k} & \textit{if } j_k\notin I\cap J\end{cases}$$

and

$$s_{j_k}\leftarrow_R\begin{cases}\{0,1\}^{l_{in,j_k}(\lambda)} & \textit{if } j_k\in I\cap J\\ \{0,1\}^{l_{out,j_k}(\lambda)} & \textit{if } j_k\notin I\cap J.\end{cases}$$

Proof. We first show that $\epsilon_2(\lambda,\vec{x},J)$ is negligible. It suffices to show that $0\le\forall l\le m-1$,

$$\begin{aligned}
\epsilon_{2,l}(\lambda,\vec{x},J):=\Big|&\Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(J,(x_{j_1},s_{j_1},m_{j_1}(1^\lambda,\vec{x};\langle\mathcal{R}^*_l\rangle)),\\
&\qquad\ldots,(x_{j_l},s_{j_l},m_{j_l}(1^\lambda,\vec{x};\langle\mathcal{R}^*_l\rangle)),(x_{j_{l+1}},s^\dagger_{j_{l+1}},m_{j_{l+1}}(1^\lambda,\vec{x};\langle\mathcal{R}^*_l\rangle)),\\
&\qquad\ldots,(x_{j_m},s^\dagger_{j_m},m_{j_m}(1^\lambda,\vec{x};\langle\mathcal{R}^*_l\rangle)),\pi(1^\lambda,\vec{x};\langle\mathcal{R}^*_l\rangle))=1\Big]\\
&-\Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(J,(x_{j_1},s_{j_1},m_{j_1}(1^\lambda,\vec{x};\left\langle\mathcal{R}^*_{l+1}\right\rangle)),\\
&\qquad\ldots,(x_{j_l},s_{j_l},m_{j_l}(1^\lambda,\vec{x};\left\langle\mathcal{R}^*_{l+1}\right\rangle)),(x_{j_{l+1}},s_{j_{l+1}},m_{j_{l+1}}(1^\lambda,\vec{x};\left\langle\mathcal{R}^*_{l+1}\right\rangle)),\\
&\qquad\ldots,(x_{j_m},s^\dagger_{j_m},m_{j_m}(1^\lambda,\vec{x};\left\langle\mathcal{R}^*_{l+1}\right\rangle)),\pi(1^\lambda,\vec{x};\left\langle\mathcal{R}^*_{l+1}\right\rangle))=1\Big]\Big|
\end{aligned}$$

is negligible and the result follows from a standard hybrid argument. Here the notation $\langle \mathcal{R}_l^* \rangle$ means that parties $\mathcal{P}_{j_1}, \dots, \mathcal{P}_{j_l}$ use random tapes $\mathcal{R}^*(1^\lambda, s_{j_1}), \dots, \mathcal{R}^*(1^\lambda, s_{j_l})$, parties $\mathcal{P}_{j_{l+1}}, \dots, \mathcal{P}_{j_m}$ use random tapes $r_{j_{l+1}}, \dots, r_{j_m}$, and others use uniformly distributed random bits. If $j_{l+1} \notin I \cap J$, then the claim holds trivially. Consider the case $j_{l+1} \in I \cap J$. Assume the negation, i.e. $\exists p \in \text{poly}(\cdot)$, there exists infinitely many $(\lambda, \vec{x})$'s such that $\epsilon_{2,l}(\lambda, \vec{x}, J) \geq \frac{1}{p(\lambda)}$. Now consider a distinguisher $\tilde{\mathcal{D}}_l^*$ with oracle access to $\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}$ against $\mathcal{R}_{j_{l+1}}$. Pick one $\vec{x}$ for each λ and give $\tilde{\mathcal{D}}_l^*$ as advice for security parameter λ. Define $\tilde{\mathcal{D}}_l^*$ as follows.

Algorithm 5 Distinguisher $\tilde{\mathcal{D}}_l^*$ against $\mathcal{R}_{j_{l+1}}^*$

```
1: procedure $\tilde{\mathcal{D}}_l^{*\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(1^\lambda, r^\dagger)$        ▷ with $\vec{x}$ as advice
2:     $s \leftarrow \mathcal{I}_{\mathcal{R}_{j_{l+1}}}(r^\dagger)$
3:     if $s \neq \perp$ then
4:         Simulate $\pi$ on input $\vec{x}$ using pseudorandom tapes for parties $\{\mathcal{P}_j\}_{j \in I \cap J \cap \{j_1, \dots, j_l\}}$ and $r^\dagger$ as random tape for party $\mathcal{P}_{l+1}$ to obtain $\{\mathcal{P}_j\}_{j \in J}$'s views $(x_{j_1}, s_{j_1}^{\dagger\dagger}, m_{j_1}^{\dagger\dagger}), \dots, (x_{j_l}, s_{j_l}^{\dagger\dagger}, m_{j_l}^{\dagger\dagger}), (x_{j_{l+1}}, s, m_{j_{l+1}}^{\dagger\dagger}), \dots, (x_{j_m}, s_{j_m}^{\dagger\dagger}, m_{j_m}^{\dagger\dagger})$ and the result $\vec{y}$.
5:         return $\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(J, (x_{j_1}, s_{j_1}^{\dagger\dagger}, m_{j_1}^{\dagger\dagger}), \dots, (x_{j_l}, s_{j_l}^{\dagger\dagger}, m_{j_l}^{\dagger\dagger}), (x_{j_{l+1}}, s, m_{j_{l+1}}^{\dagger\dagger}), \dots, (x_{j_m}, s_{j_m}^{\dagger\dagger}, m_{j_m}^{\dagger\dagger}), \vec{y})$
6:     else
7:         return 0
8:     end if
9: end procedure
```

Under the condition that line 7 is never executed in the real view and the simulated view in Algorithm 4, $r_{j_{l+1}} \leftarrow_R \{0,1\}^{l_{out}(\lambda, j_{l+1})}$ is equivalent to $r_{j_{l+1}} \leftarrow_R \text{range}(\mathcal{R}_{j_{l+1}}, \lambda)$. Now

$$\begin{aligned}
&\left|\Pr\left[\tilde{\mathcal{D}}^{*\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}\left(1^\lambda, r_{j_{l+1}}\right) = 1\right] - \Pr\left[\tilde{\mathcal{D}}^{*\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}\left(1^\lambda, \mathcal{R}_{j_{l+1}}(1^\lambda, s_{j_{l+1}})\right) = 1\right]\right| \\
= &\left|\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(J, (x_{j_1}, s_{j_1}, m_{j_1}(1^\lambda, \vec{x}; \langle \mathcal{R}_l^* \rangle)),\right.\right. \\
&\quad \dots, (x_{j_l}, s_{j_l}, m_{j_l}(1^\lambda, \vec{x}; \langle \mathcal{R}_l^* \rangle)), (x_{j_{l+1}}, s_{j_{l+1}}^\dagger, m_{j_{l+1}}(1^\lambda, \vec{x}; \langle \mathcal{R}_l^* \rangle)), \\
&\quad \left. \dots, (x_{j_m}, s_{j_m}^\dagger, m_{j_m}(1^\lambda, \vec{x}; \langle \mathcal{R}_l^* \rangle)), \pi(1^\lambda, \vec{x}; \langle \mathcal{R}_l^* \rangle)) = 1\right] \\
&\; - \Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(J, (x_{j_1}, s_{j_1}, m_{j_1}(1^\lambda, \vec{x}; \left\langle \mathcal{R}_{l+1}^* \right\rangle)),\right. \\
&\qquad \dots, (x_{j_l}, s_{j_l}, m_{j_l}(1^\lambda, \vec{x}; \left\langle \mathcal{R}_{l+1}^* \right\rangle)), (x_{j_{l+1}}, s_{j_{l+1}}, m_{j_{l+1}}(1^\lambda, \vec{x}; \left\langle \mathcal{R}_{l+1}^* \right\rangle)), \\
&\qquad \left.\left. \dots, (x_{j_m}, s_{j_m}^\dagger, m_{j_m}(1^\lambda, \vec{x}; \left\langle \mathcal{R}_{l+1}^* \right\rangle)), \pi(1^\lambda, \vec{x}; \left\langle \mathcal{R}_{l+1}^* \right\rangle)) = 1\right]\right| \\
= &\; \epsilon_{2,l}(\lambda, \vec{x}, J) > \frac{1}{p(\lambda)}.
\end{aligned}$$

Since $\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J} \subset \{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I}$, by the second assumption, $\mathcal{R}_{j_{l+1}}$ is also non-uniformly indistinguishable in its range relative to $\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}$, which leads to a contradiction.

The proof for ϵ_3 is similar to that of ϵ_2, where instead of simulating π, the distinguisher $\tilde{\mathcal{D}}_l^*$ against $\mathcal{R}_{j_{l+1}}$ simulates $\tilde{\mathcal{S}}$ using random tapes as the above algorithm. □

Let

$$\begin{aligned}
P_1(\lambda, \vec{x}, J) &:= \Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(J, (x_{j_1}, s^{\dagger}_{j_1}, m_{j_1}(1^\lambda, \vec{x}; \langle r_{j_k} \rangle_{j_k \in J})), \\
&\qquad \ldots, (x_{j_m}, s^{\dagger}_{j_m}, m_{j_m}(1^\lambda, \vec{x}; \langle r_{j_k} \rangle_{j_k \in J})), \pi(1^\lambda, \vec{x}; \langle r_{j_k} \rangle_{j_k \in J})) = 1\Big] \\
P_2(\lambda, \vec{x}, J) &:= \Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(J, (x_{j_1}, s_{j_1}, m_{j_1}(1^\lambda, \vec{x}; \big\langle \mathcal{R}^*_{j_k}(1^\lambda, s_{j_k}) \big\rangle_{j_k \in J})), \\
&\qquad \ldots, (x_{j_m}, s_{j_m}, m_{j_m}(1^\lambda, \vec{x}; \big\langle \mathcal{R}^*_{j_k}(1^\lambda, s_{j_k}) \big\rangle_{j_k \in J})), \pi(1^\lambda, \vec{x}; \big\langle \mathcal{R}^*_{j_k}(1^\lambda, s_{j_k}) \big\rangle_{j_k \in J})) = 1\Big] \\
Q_1(\lambda, \vec{x}, J) &:= \Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(\tilde{\mathcal{S}}(1^\lambda, J, \vec{x}_J, \vec{f}_J(\vec{x}); s^{\dagger}_{j_1}, \ldots, s^{\dagger}_{j_m}, \tau), \vec{f}(\vec{x})) = 1\Big] \\
Q_2(\lambda, \vec{x}, J) &:= \Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(\tilde{\mathcal{S}}(1^\lambda, J, \vec{x}_J, \vec{f}_J(\vec{x}); s_{j_1}, \ldots, s_{j_m}, \tau), \vec{f}(\vec{x})) = 1\Big]
\end{aligned}$$

thus ϵ_2 and ϵ_3 become

$$\begin{aligned}
\epsilon_2(\lambda, \vec{x}, J) &= |P_1(\lambda, \vec{x}, J) - P_2(\lambda, \vec{x}, J)| \\
\epsilon_3(\lambda, \vec{x}, J) &= |Q_1(\lambda, \vec{x}, J) - Q_2(\lambda, \vec{x}, J)| \,.
\end{aligned}$$

Summarising the discussion above, under the condition that line 7 is never executed in the real view and the simulated view conditioned on $J = J^{\dagger}$,

$$\begin{aligned}
&\Big|\Pr\Big[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}\left(\mathsf{VIEW}_J(\vec{x}; \vec{r}), \pi(1^\lambda, \vec{x}; \vec{r})\right) = 1\Big] \\
&\quad - \Pr\Big[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(\mathcal{S}(1^\lambda, J, \vec{x}_J, \vec{f}_J(\vec{x}); r_{j_1}, \ldots, r_{j_m}, \tau), \vec{f}(\vec{x})) = 1\Big]\Big| \\
&= \Big|\Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(J, (x_{j_1}, s^{\dagger}_{j_1}, m_{j_1}(1^\lambda, \vec{x}; \langle r_{j_k} \rangle_{j_k \in J})), \\
&\qquad \ldots, (x_{j_m}, s^{\dagger}_{j_m}, m_{j_m}(1^\lambda, \vec{x}; \langle r_{j_k} \rangle_{j_k \in J})), \pi(1^\lambda, \vec{x}; \langle r_{j_k} \rangle_{j_k \in J})) = 1\Big] \\
&\quad - \Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(\tilde{\mathcal{S}}(1^\lambda, J, \vec{x}_J, \vec{f}_J(\vec{x}); s^{\dagger}_{j_1}, \ldots, s^{\dagger}_{j_m}, \tau), \vec{f}(\vec{x})) = 1\Big]\Big| \\
&= |P_1(\lambda, \vec{x}, J) - Q_1(\lambda, \vec{x}, J)| \\
&= |(P_2(\lambda, \vec{x}, J) - Q_2(\lambda, \vec{x}, J)) + (P_1(\lambda, \vec{x}, J) - P_2(\lambda, \vec{x}, J)) - (Q_1(\lambda, \vec{x}, J) - Q_2(\lambda, \vec{x}, J))| \\
&\geq |P_2(\lambda, \vec{x}, J) - Q_2(\lambda, \vec{x}, J)| - |P_1(\lambda, \vec{x}, J) - P_2(\lambda, \vec{x}, J)| - |Q_1(\lambda, \vec{x}, J) - Q_2(\lambda, \vec{x}, J)| \\
&= \Big|\Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}\Big(\left(\mathsf{VIEW} \circ_{I \cap J} \{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}\right)_J(\vec{x}; \langle s_j \rangle_{j \in I \cap J}), \pi(1^\lambda, \vec{x}; \langle s_j \rangle_{j \in I \cap J})\Big) = 1\Big] \\
&\quad - \Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(\tilde{\mathcal{S}}(1^\lambda, J, \vec{x}_J, \vec{f}_J(\vec{x}); s_{j_1}, \ldots, s_{j_m}, \tau), \vec{f}(\vec{x})) = 1\Big]\Big| \\
&\quad - \epsilon_2(\lambda, \vec{x}, J) - \epsilon_3(\lambda, \vec{x}, J)
\end{aligned}$$

where $\left(\mathsf{VIEW} \circ_{I \cap J} \{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}\right)_J$ denotes the view of $\{\mathcal{P}\}_{i \in J}$ during the execution of $\pi \circ_{I \cap J} \{\mathcal{R}_i\}_{i \in I \cap J}$ (what VIEW_J is to π). Note that $\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}$ always outputs 0 when line 7 is executed in Algorithm 4. Thus as for *overall* probability in *the real view and the simulated view conditioned on* $J = J^{\dagger}$, we have

$$\begin{aligned}
&\Big|\Pr\Big[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}\left(\mathsf{VIEW}_J(\vec{x}; \vec{r}), \pi(1^\lambda, \vec{x}; \vec{r})\right) = 1\Big] \\
&\quad - \Pr\Big[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(\mathcal{S}(1^\lambda, J, \vec{x}_J, \vec{f}_J(\vec{x}); r_{j_1}, \ldots, r_{j_m}, \tau), \vec{f}(\vec{x})) = 1 \mid J^{\dagger} = J\Big]\Big| \\
&\geq \Big(\Big|\Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}\Big(\left(\mathsf{VIEW} \circ_{I \cap J} \{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}\right)_J(\vec{x}; \langle s_j \rangle_{j \in I \cap J}), \pi(1^\lambda, \vec{x}; \langle s_j \rangle_{j \in I \cap J})\Big) = 1\Big] \\
&\quad - \Pr\Big[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i \in I \cap J}}(\tilde{\mathcal{S}}(1^\lambda, J, \vec{x}_J, \vec{f}_J(\vec{x}); s_{j_1}, \ldots, s_{j_m}, \tau), \vec{f}(\vec{x})) = 1 \mid J^{\dagger} = J\Big]\Big| \\
&\quad - \epsilon_2(\lambda, \vec{x}, J) - \epsilon_3(\lambda, \vec{x}, J)\Big) \cdot \prod_{j_k \in I \cap J} \epsilon_{1, j_k}(\lambda)
\end{aligned}$$

which implies that

$$\begin{aligned}
&\Bigg|\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\left(\mathsf{VIEW}\circ_{I\cap J}\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}\right)_J(\vec{x};\langle s_j\rangle_{j\in I\cap J}),\pi(1^\lambda,\vec{x};\langle s_j\rangle_{j\in I\cap J})\right)=1\right]\\
&-\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(\tilde{\mathcal{S}}(1^\lambda,J,\vec{x}_J,\vec{f}_J(\vec{x});s_{j_1},\dots,s_{j_m},\tau),\vec{f}(\vec{x}))=1\mid J^\dagger=J\right]\Bigg|\\
\le\;&\frac{\begin{array}{l}\Big|\Pr\left[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\mathsf{VIEW}_J(\vec{x};\vec{r}),\pi(1^\lambda,\vec{x};\vec{r})\right)=1\right]\\ \quad-\Pr\left[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(\mathcal{S}(1^\lambda,J,\vec{x}_J,\vec{f}_J(\vec{x});r_{j_1},\dots,r_{j_m},\tau),\vec{f}(\vec{x}))=1\mid J^\dagger=J\right]\Big|\end{array}}{\prod_{j_k\in I\cap J}\epsilon_{1,j_k}(\lambda)}\\
&+\epsilon_2(\lambda,\vec{x},J)+\epsilon_3(\lambda,\vec{x},J).
\end{aligned}$$

By the security of π, the component of the output of $\mathcal{S}$ corresponding to J in the real view, denoted $J^\dagger$, satisfies $J^\dagger\overset{\text{nu. c}}{\equiv}J$, i.e. $J^\dagger$ and J are non-uniformly indistinguishable. By the definition of $\tilde{\mathcal{S}}$, the same holds for $\tilde{\mathcal{S}}$. Since J is not a random variable, this can only happen when $J^\dagger\neq J$ occurs with negligible probability, say $\epsilon_4(1^\lambda,\vec{x},J)$. Thus the inequality above can be rewritten as

$$\begin{aligned}
&\Bigg|\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\left(\mathsf{VIEW}\circ_{I\cap J}\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}\right)_J(\vec{x};\langle s_j\rangle_{j\in I\cap J}),\pi(1^\lambda,\vec{x};\langle s_j\rangle_{j\in I\cap J})\right)=1\right]\\
&-\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(\tilde{\mathcal{S}}(1^\lambda,J,\vec{x}_J,\vec{f}_J(\vec{x});s_{j_1},\dots,s_{j_m},\tau),\vec{f}(\vec{x}))=1\right]\Bigg|\\
\le\;&\frac{\begin{array}{l}\Big|\Pr\left[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\mathsf{VIEW}_J(\vec{x};\vec{r}),\pi(1^\lambda,\vec{x};\vec{r})\right)=1\right]\\ \quad-\Pr\left[\mathcal{D}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}(\mathcal{S}(1^\lambda,J,\vec{x}_J,\vec{f}_J(\vec{x});r_{j_1},\dots,r_{j_m},\tau),\vec{f}(\vec{x}))=1\right]\Big|+\epsilon_4(\lambda,\vec{x},J)\end{array}}{\prod_{j_k\in I\cap J}\epsilon_{1,j_k}(\lambda)}\\
&+\epsilon_2(\lambda,\vec{x},J)+\epsilon_3(\lambda,\vec{x},J)+\epsilon_4(\lambda,\vec{x},J).
\end{aligned}$$

The first term is negligible by the assumptions that π is secure against party $\mathcal{P}_i$ relative to $\mathcal{I}_{\mathcal{R}}$, that $\epsilon_4(\lambda,\vec{x},J)$ is negligible, and that each ϵ_{1,j_k} is bounded below by the inverse of a polynomial. $\epsilon_2(\lambda,\vec{x},J)$ and $\epsilon_3(\lambda,\vec{x},J)$ are negligible by Claim 19.1. Therefore the expression is negligible.

To prove Theorem 19, it suffices to show that

$$\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\left(\mathsf{VIEW}\circ_{I\cap J}\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}\right)_J(\vec{x};\langle s_j\rangle_{j\in I\cap J}),\pi(1^\lambda,\vec{x};\langle s_j\rangle_{j\in I\cap J})\right)=1\right]$$

and

$$\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\left(\mathsf{VIEW}\circ_{I}\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I}\right)_J(\vec{x};\langle s_j\rangle_{j\in I}),\pi(1^\lambda,\vec{x};\langle s_j\rangle_{j\in I})\right)=1\right]$$

only differ negligibly. This follows from a hybrid argument. Let $I\backslash J=\{i_1,i_2,\dots,i_l\}\,(0\le l\le|I|,i_1<i_2<\dots<i_l)$. Now

$$\begin{aligned}
&\Bigg|\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\left(\mathsf{VIEW}\circ_{I}\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I}\right)_J(\vec{x};\langle s_j\rangle_{j\in I}),\pi(1^\lambda,\vec{x};\langle s_j\rangle_{j\in I})\right)=1\right]\\
&-\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\left(\mathsf{VIEW}\circ_{I\cap J}\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}\right)_J(\vec{x};\langle s_j\rangle_{j\in I\cap J}),\pi(2^\lambda,\vec{x};\langle s_j\rangle_{j\in I\cap J})\right)=1\right]\Bigg|\\
\le&\sum_K\Bigg|\Pr\Bigg[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\Bigg(\left(\mathsf{VIEW}\circ_{K\cup\{i_{|K|+1}\}}\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in K\cup\{i_{|K|+1}\}}\right)_J(\vec{x};\langle s_j\rangle_{j\in K\cup\{i_{|K|+1}\}}),\\
&\qquad\pi(1^\lambda,\vec{x};\langle s_j\rangle_{j\in K\cup\{i_{|K|+1}\}})\Bigg)=1\Bigg]\\
&\quad-\Pr\left[\tilde{\mathcal{D}}^{\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}}\left(\left(\mathsf{VIEW}\circ_{K}\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in K}\right)_J(\vec{x};\langle s_j\rangle_{j\in K}),\pi(1^\lambda,\vec{x};\langle s_j\rangle_{j\in K})\right)=1\right]\Bigg|
\end{aligned}$$

where K spans over $\emptyset,\{i_1\},\{i_1,i_2\},\dots,\{i_1,i_2,\dots,i_{l-1}\}$. For each K, $I\cap J=I\backslash(I\backslash J)\subset I\backslash\{i_{|K|+1}\}$, thus $\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}\subset\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\backslash\{i_{|K|+1}\}}$. By computational irrelevancy, each $\mathcal{R}_{i_{|K|+1}}$ is non-uniformly secure relative to $\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\backslash\{i_{|K|+1}\}}$, thus also non-uniformly secure relative to $\{\mathcal{I}_{\mathcal{R}_i}\}_{i\in I\cap J}$. Hence each summand is negligible, so is the whole expression, which completes the proof of Theorem 19. ■

4 Related Works

4.1 Relation to Information-Theoretic Assumptions

We discuss the relation between our proposed sufficient conditions and the information-theoretic ones used in [10]. Since [10] only considered the 1-adversary-1-PRG case, we compare its result with Theorem 12.

[10] assumes information-theoretically secure protocols (i.e. the variation distance between the real and simulated views is negligible) with raw randomness. This is a stronger one than the first assumption of Theorem 12 since it is a well-known fact that statistical closeness implies computational indistinguishability and its proof relativises to any family of oracles.

Next we show that the min-entropy condition on PRGs is a stronger one than our second assumption of Theorem 12.

Proposition 20. *Let $\mathcal{R}$ be a PRG. If $l_{out}(\lambda) - H_\infty(\mathcal{R}(1^\lambda, \cdot)) \in O(\log \lambda)$, then $\frac{|\mathrm{range}(\mathcal{R},\lambda)|}{2^{l_{out}(\lambda)}}$ is noticeable. Here $H_\infty(\mathcal{R}(1^\lambda, \cdot))$ is called the min-entropy of $\mathcal{R}$ and is defined as*

$$H_\infty(\mathcal{R}(1^\lambda, \cdot)) := -\max_{r \in \{0,1\}^{l_{out}(\lambda)}} \log_2 \Pr\left[\mathcal{R}(1^\lambda, s) = r\right]$$

where $s \leftarrow_R \{0,1\}^{l_{in}(\lambda)}$.

Proof. By definition we can focus on the range of $\mathcal{R}$ when considering min-entropy:

$$H_\infty(\mathcal{R}(1^\lambda, \cdot)) = -\max_{r \in \mathrm{range}(\mathcal{R},\lambda)} \log_2 \Pr\left[\mathcal{R}(1^\lambda, s) = r\right].$$

For a finite set, the uniform distribution yields the highest min-entropy among all distributions over this set, thus

$$\begin{aligned} H_\infty(\mathcal{R}(1^\lambda, \cdot)) &\leq -\max_{r \in \mathrm{range}(\mathcal{R},\lambda)} \log_2 \Pr\left[U_{\mathrm{range}(\mathcal{R},\lambda)} = r\right] \\ &= \log_2 |\mathrm{range}(\mathcal{R},\lambda)|. \end{aligned}$$

The assumption can be rewritten as $\frac{2^{l_{out}(\lambda)}}{2^{H_\infty(\mathcal{R}(1^\lambda,\cdot))}} \leq p(\lambda)$ for sufficiently large λ's for some polynomial p. Thus

$$\frac{2^{l_{out}(\lambda)}}{|\mathrm{range}(\mathcal{R},\lambda)|} = \frac{2^{l_{out}(\lambda)}}{2^{\log_2 |\mathrm{range}(\mathcal{R},\lambda)|}} \leq \frac{2^{l_{out}(\lambda)}}{2^{H_\infty(\mathcal{R}(1^\lambda,\cdot))}} \leq p(\lambda)$$

Taking the inverse on both sides yields the desired result. ■

4.2 On Random Oracle vs Hash Function Ensembles

Here we note that the technique used above seems unlikely to resolve the problem that occurs when a random oracle is replaced with a hash function ensemble ([2]). For a cryptosystem that is secure under the random oracle model, if we want to prove (based on this fact) the security when the random oracle is replaced with a hash function ensemble with similar techniques, we have to rely on some computational indistinguishability between them. However, no well-known security requirements (one-wayness, collision resistance, etc.) on hash functions seem to provide such indistinguishability in any sense. A seemingly promising indistinguishability requirement might be that

$$\left| \Pr_{\mathscr{O}_k, s \leftarrow_R \{0,1\}^k} \left[\mathcal{D}^{\mathscr{O}_k}(1^k, s)\right] - \Pr_{s \leftarrow_R \{0,1\}^k} \left[\mathcal{D}^{f_s}(1^k, s)\right] \right|$$

be negligible, where $\mathscr{O}_k$ denotes (the distribution of) random oracles outputting strings of length $l_{out}(k)$ and f_s denotes the element with index s of a hash function ensemble. Note that we have to pass the seed s to the distinguisher since all parties (including adversaries) are supposed to know the seed in an implementation of random oracles by hash functions. However, a distinguisher can easily distinguish the two by computing $f_s(x)$ itself with arbitrary x and compare with the result of the oracle query.

Since both appropriate users (or honest parties) have access to the same random oracle or hash functions, one may think that the notion of indistinguishability, which assumes that the random bits are private to each party, is anyway not suitable to be used in the random oracle vs hash function setting.

A less naïve notion called "indifferentiability", proposed by [9], is a generalisation of indistinguishability to deal with public and private interfaces. However, even this notion cannot be applied to the random oracle vs hash function setting – no hash function ensemble is indifferentiable from a random oracle.

Thus we can see there seems to exist a huge gap between the random oracle model and reality (in the sense that even trivial algorithms can distinguish them). Indeed, [2] presents stronger negative results on RO vs hash functions than does [10] on real randomness vs pseudorandomness.

4.3 Relation to Computational Independency of One-Way Functions

It has been noticed by previous works that use of closely related cryptographic primitives may cause problems. [4] discussed a notion called "computationally independent one-way functions" to avoid the problems in interactive proof systems. Here we briefly discuss the relationship between our proposed computational irrelevancy of pairs of PRGs (Definition 14) and computational dependency of pairs of one-way functions proposed by [4].

A straightforward adaptation of computational irrelevancy for onw-way functions can be formalised as follows.

Definition 21. Let f be a one-way function. We say $\mathcal{R}$ is *one-way relative to* $\mathcal{O} = \{\mathcal{O}_i\}_{i \in I} \subset \mathcal{PTM}$ if $\forall \mathcal{I} \in \mathcal{PPT}$,

$$\Pr\left[\mathcal{I}^{\mathcal{O}}\left(1^\lambda, f(x)\right) \in f^{-1}(f(x))\right]$$

is negligible where $x \leftarrow_R \{0,1\}^\lambda$.

Definition 22. Let f_1 and f_2 be one-way functions. We say f_1 and f_2 are *computationally irrelevant* if for $i \in \{1,2\}$, f_i is one-way relative to $\mathcal{I}_{f_{3-i}}$, where $\mathcal{I}_f$ is the inverter specified the same way as in Definition 5.

For comparison, we restate the definition of pairs of computationally independent one-way functions.

Definition 23 ([4])**.** Let f_1 and f_2 be one-way functions. We say f_1 and f_2 are *computationally irrelevant* if

- (CI-a) $g(x) := (f_1(x), f_2(x))$ is also one-way.
- (CI-b) For $i \in \{1,2\}$, $\forall \mathcal{A} \in \mathcal{PPT}$,

  $$\Pr\left[\mathcal{A}\left(1^\lambda, f_i(x)\right) = f_{3-i}(x)\right]$$

  is negligible where $x \leftarrow_R \{0,1\}^\lambda$.

Computational irrelevancy does not capture (CI-a) since we only considered PRGs in this paper and different PRGs are supposed to use different seeds anyway. However, it is a stronger notion than (CI-b).

Proposition 24. *For pairs of one-way functions, computational irrelevancy implies (CI-b), i.e. for two one-way functions f_1 and f_2, if they are computationally irrelevant, then for $i \in \{1,2\}$, $\forall \mathcal{A} \in \mathcal{PPT}$,*

$$\Pr\left[\mathcal{A}\left(1^\lambda, f_i(x)\right) = f_{3-i}(x)\right]$$

is negligible.

Proof. Assume for some $i \in \{1,2\}$, there exists $\mathcal{A} \in \mathcal{PPT}$ that given $f_i(x)$ computes $f_{3-i}(x)$. Then an inverter of f_i can be obtained by calling $\mathcal{A}$ on $f_i(x)$ and calling $\mathcal{I}_{f_{3-i}}$ on the output of $\mathcal{A}$. The success probability is the same as that of $\mathcal{A}$. ■

5 Conclusion

In this paper, we formalised computational irrelevancy in terms of MPC protocols and PRGs using the relativisation paradigm. Also, for various adversarial settings in the semi-honest model, we provided sufficient conditions under which security of MPC protocols are preserved even if PRGs are used under computational assumptions.

It remains open to construct protocols and PRGs that satisfy these computational irrelevancy conditions. We note here that constructing such examples theoretically is very easy. For example, in terms of protocols that are irrelevant from PRGs, information-theoretically secure ones always satisfy these conditions; for the ones that are not necessarily information-theoretically secure, replacing the underlying computational hardness assumptions with the ones relativised to the inverters of PRGs directly results in the protocols with the desired properties. However, whether these relativised assumptions can be considered "reasonable" requires further study in the literature. Since, as noted before, the relativisation paradigm has been of great interest in both complexity theory and cryptography and proved to be useful in previous works, we optimistically hope that subsequent works will stress this open problem.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR2113, Japan.

References

[1] Theodore Baker, John Gill, and Robert Solovay. "Relativizations of the $\mathcal{P} =?\mathcal{NP}$ Question". In: *SIAM Journal on Computing* 4.4 (1975), pp. 431–442. DOI: 10.1137/0204037.

[2] Ran Canetti, Oded Goldreich, and Shai Halevi. "The Random Oracle Methodology, Revisited". In: *J. ACM* 51.4 (July 2004), pp. 557–594. ISSN: 0004-5411. DOI: 10.1145/1008731.1008734. URL: https://doi.org/10.1145/1008731.1008734.

[3] B. Chor and E. Kushilevitz. "A Zero-One Law for Boolean Privacy". In: *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*. STOC '89. Seattle, Washington, USA: Association for Computing Machinery, 1989, pp. 62–72. ISBN: 0897913078. DOI: 10.1145/73007.73013. URL: https://doi.org/10.1145/73007.73013.

[4] Sabyasachi Dutta and Kouichi Sakurai. "Theory and Application of Computationally-Independent One-Way Functions: Interactive Proof of Ability—Revisited". In: *Proceedings of the Fifth International Conference on Mathematics and Computing*. Ed. by Debasis Giri et al. Singapore: Springer Singapore, 2021, pp. 97–109. ISBN: 978-981-15-5411-7.

[5] Oded Goldreich. *Foundations of Cryptography*. Vol. 2. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511721656.

[6] R. Impagliazzo and S. Rudich. "Limits on the Provable Consequences of One-Way Permutations". In: *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*. STOC '89. Seattle, Washington, USA: Association for Computing Machinery, 1989, pp. 44–61. ISBN: 0897913078. DOI: 10.1145/73007.73012. URL: https://doi.org/10.1145/73007.73012.

[7] Eike Kiltz. "Chosen-Ciphertext Security from Tag-Based Encryption". In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 581–600. ISBN: 978-3-540-32732-5.

[8] Ben Lynn. *On the Implementation of Pairing-based Cryptosystems*. Ph.D thesis, Stanford University, 2007. URL: https://crypto.stanford.edu/pbc/thesis.pdf.

[9] Ueli Maurer, Renato Renner, and Clemens Holenstein. "Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology". In: *Theory of Cryptography*. Ed. by Moni Naor. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 21–39. ISBN: 978-3-540-24638-1.

[10] Koji Nuida. "Cryptographic Pseudorandom Generators Can Make Cryptosystems Problematic". In: *Public-Key Cryptography – PKC 2021*. Ed. by Juan A. Garay. Cham: Springer International Publishing, 2021, pp. 441–468. ISBN: 978-3-030-75248-4.

[11] Tatsuaki Okamoto and David Pointcheval. "The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes". In: *Public Key Cryptography*. Ed. by Kwangjo Kim. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 104–118. ISBN: 978-3-540-44586-9.

[12] Andrew C. Yao. "Protocols for secure computations". In: *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*. 1982, pp. 160–164. DOI: 10.1109/SFCS.1982.38.

[13] Andrew Chi-Chih Yao. "How to generate and exchange secrets". In: *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25.