On pseudorandomization of information-theoretically secure schemes without hardness assumptions

Koji Nuida*

September 1, 2012

Abstract

A recent work by Nuida and Hanaoka (in ICITS 2009) provided a proof technique for security of information-theoretically secure cryptographic schemes in which the random input tape is implemented by a pseudorandom generator (PRG). In this paper, we revisit their proof technique and generalize it by introducing some trade-off factor, which involves the original proof technique as a special case and provides a room of improvement of the preceding result. Secondly, we consider two issues of the preceding result; one is the requirement of some hardness assumption in their proof; another is the gap between non-uniform and uniform computational models appearing when transferring from the exact security formulation adopted in the preceding result to the usual asymptotic security. We point out that these two issues can be resolved by using a PRG proposed by Impagliazzo, Nisan and Wigderson (in STOC 1994) against memory-bounded distinguishers, instead of usual PRGs against time-bounded distinguishers. We also give a precise formulation of a computational model explained by Impagliazzo et al., and by using this, perform a numerical comparison showing that, despite the significant advantage of removing hardness assumptions, our result is still better than, or at least competitive to, the preceding result from quantitative viewpoints. The results of this paper would suggest a new motivation to use PRGs against distinguishers with computational constraints other than time complexity in practical situations rather than just theoretical works.

Keywords: Information-theoretic security, pseudorandomization, unconditional security, Impagliazzo–Nisan–Wigderson pseudorandom generator

1 Introduction

1.1 Background and preceding works

In practical uses of cryptographic schemes, pseudorandom generators (PRGs) are usually applied to efficiently "stretch" short (truly) random bits to much longer pseudorandom sequences used in the schemes. From the viewpoint of provable security, it is reasonable to combine computationally indistinguishable PRGs to computationally secure cryptographic schemes. In contrast, when the cryptographic schemes under discussion have *information-theoretic* security, i.e., security against adversaries with *unbounded* computational powers is considered, a straightforward application of PRGs to such schemes seems problematic, as there exist no (non-trivial) PRGs whose proof of indistinguishability requires no constraints on computational powers of distinguishers. This problem might be a hurdle when one wants to rigorously implement information-theoretically secure cryptographic schemes in practical applications.

Recently, Nuida and Hanaoka [11, 12] invented a proof technique for combinations of PRGs with a certain kind of information-theoretically secure schemes, which enabled us to prove that the differences of attack success probabilities for the scheme in random case (i.e., when truly random bits are used in the scheme) and

*Research Institute for Secure Systems (RISEC), National Institute of Advanced Industrial Science and Technology (AIST) (k.nuida@aist.go.jp)

in pseudorandom case (i.e., when the output of a PRG is used in the scheme instead) are bounded *without computational constraints on the adversary's attack algorithm.* Their provable bound depends not only on the strength of indistinguishability of the PRG, but also on the size of the input set for the adversary's attack algorithm (which is determined by the formulation of a security notion under consideration); the larger the adversary's input set is, the worse the resulting bound will be. Therefore, the adversary's input set should be significantly small to make the bound effective, which is a somewhat considerable restriction for types of the cryptographic schemes. (Another proof technique was presented in an earlier work of Dubrov and Ishai [3], but the restriction for the cryptographic schemes in their argument seems much more severe than that in [11, 12]; see Section 2.3 for a detailed discussion.)

On the other hand, it is mentioned in [12] that their proof technique still requires some *hardness assumption*, despite that the computational constraints on the adversary's attack algorithms for the cryptographic schemes have been removed. For example, the numerical example in [11] used a certain PRG given in [4], whose proof of indistinguishability is based on the DDH assumption (in addition to certain constraints on computational times of distinguishers). As such a hardness assumption may be compromised due to time passage (in contrast to the information-theoretic security of the original cryptographic scheme), this also becomes a disadvantage from the viewpoint of long-term security. To the author's best knowledge, there are no preceding works that discuss the problem.

1.2 Our contributions, and organization of the paper

In order to investigate the restriction for the proof technique of Nuida and Hanaoka [11, 12] mentioned above, in Section 2 we give another proof technique for the security of pseudorandomization of information-theoretically secure cryptographic schemes. Recall that the size of the adversary's input set was a crucial factor in the result of [11, 12]. Intuitively, our argument interprets the size of the adversary's input set as "the number of components in the partition of the adversary's input set into subsets consisting of a single element", and our proof technique generalizes the shapes of the partition in order to reduce the number of the components. In fact, the bound derived in [11, 12] also depends on the computational complexity of certain "imaginary distinguishers" (involving the original cryptographic scheme as a part) introduced in the proof, and our generalization actually introduces a trade-off between the number of components in the partition and the complexity of the imaginary distinguishers. We also introduce another generalizing factor, i.e., approximability of the adversary's attack algorithms by probabilistic ensembles of relatively simpler algorithms. A detailed study on the matter of the approximability is left as a future research topic.

For the other issue of [11, 12] on the requirement of hardness assumptions, we notice that there exist PRGs whose indistinguishability against distinguishers with constraints on computational complexity of some types other than time complexity is provable *without any hardness assumption* and that, by using such a PRG instead of those used in [11, 12], the requirement of hardness assumptions in [11, 12] can be removed. An example of such PRGs is the one given by Impagliazzo, Nisan and Wigderson [6] (henceforth called *INW PRG*), with constraints on the distinguishers specified in terms of *memory* complexity. Moreover, we point out another issue of the argument in [11, 12] using PRGs against time-bounded distinguishers. Namely, the argument in [11, 12] is based on the exact security formulation (e.g., (T, ε)-security), while the standard security formulation for such PRGs is asymptotic security, as a convincing verification of the underlying hardness assumption in terms of time complexity in exact security formulation is very difficult. Now if we want to convert the argument in terms of exact security into that in terms of asymptotic security, then we will face a gap between non-uniform and uniform models of computation (such as ones recently discussed by Koblitz and Menezes [7]). This issue is also resolved by using e.g., INW PRGs, as the exact security of the PRG can be rigorously evaluated (without any hardness assumption). A more detailed discussion (including a relation to some PRGs against constant-size circuits proposed in [3]) will be given in Section 3.

For the sake of quantitative evaluation of the indistinguishability, in Section 4 we give a precise formulation of a computational model which suits the structure of INW PRGs. In fact the idea of the model was already explained in [6] without a precise formulation; we establish a mathematically rigorous model that enables us to perform a detailed quantitative evaluation. Intuitively, this model interprets a computation process using a random input tape as a "bucket-brigade" played by the cells of the random tape, where an

intermediate computation result stored in the memory is delivered from a currently accessed cell to the next cell. The model is thus described in terms of multi-party protocols; for the purpose, we also summarize a mathematical formulation of multi-party protocols. We emphasize that, although in the computational model each algorithm is expressed as a multi-party protocol, it does *not* mean that the cryptographic scheme itself should be a multi-party protocol. (When the cryptographic scheme under consideration is a multi-party protocol, it has also to be concerned who should execute the PRG and how to securely and efficiently distribute the PRG's outputs to players of the protocol in the presence of computationally unbounded adversary. However, this issue need not be concerned when the whole of the PRG's output is used by a single player; for instance, fingerprint code [2] studied in our numerical example below satisfies the condition.)

In Section 5, we summarize a construction of INW PRGs given in [6]. We also show some properties of INW PRGs, a part of which improves slightly the counterpart in [6]. Section 6 gives an evaluation of the seed lengths of INW PRGs combined with information-theoretically secure cryptographic schemes according to our proof technique. By Theorem 4 below, the seed length $\nu_0^\dagger$ is asymptotically estimated as

$$\nu_0^\dagger \sim 3(\log_2 \mu)^2 + 2N_{\mathrm{read}} \log_2 |M| \log_2 \mu + 2\log_2(n/\varepsilon) \log_2 \mu \quad (\mu, |M| \to \infty), \tag{1}$$

where μ denotes the original bit length of the random input, M denotes the set of the possible memory state used by the imaginary distinguishers mentioned above (hence $\log_2 |M|$ means the bit length of the memory), N_{read} means that each cell of the random tape is accessed at most N_{read} times during the computation, n denotes the number of components in the partition of the adversary's input set mentioned above, and ε denotes the desired bound for the differences of attack success probabilities for the cryptographic scheme in random and pseudorandom cases. Finally, in Section 7 we provide a numerical example in the same setting for the cryptographic scheme as the example in [11]. This example shows that the performance of our technique in reducing the required random bits is better than, or at least competitive to, the one in [11], despite the removal of hardness assumptions which are required by the argument in [11].

2 Our proposed proof technique

In this section we describe, by using a toy example, our proposed proof technique to prove information-theoretic security of a scheme in which some random objects are constructed by using a pseudorandom generator (PRG). We emphasize that the indistinguishability of any PRG (except trivial ones) requires computational constraints on the distinguishers, while we are concerning the cases of adversaries without computational constraints (which is the targeted situation of information-theoretic security), therefore our proof technique is never trivial. We also notice that our technique is an improvement of the one proposed by Nuida and Hanaoka [11, 12], as mentioned in Section 2.3.

2.1 Notations and terminology

Here we summarize some notations and terminology used below. For each probabilistic algorithm A, let $\mathrm{dom}(\mathsf{A})$ denote the set of inputs for A for which A halts within finite time, and let $\mathrm{ran}(\mathsf{A})$ be the output set of A. Let F_{A} denote the probabilistic function $\mathrm{dom}(\mathsf{A}) \to \mathrm{ran}(\mathsf{A})$ realized by the algorithm A.

Each probabilistic function $f\colon X \to Y$, with X and Y being both finite sets, is in one-to-one correspondence to a matrix $p(f) = (p(f)_{y,x})_{x \in X, y \in Y}$ defined by

$$p(f)_{y,x} := Pr[f(x) = y] \text{ for every } x \in X\,,\ y \in Y\ . \tag{2}$$

Now for a collection $(f_i)_{i=1}^\ell$ of probabilistic functions $X \to Y$ and a probability distribution $(p_i)_{i=1}^\ell$ on the index set $\{1, 2, \ldots, \ell\}$ (i.e., $0 \le p_i \le 1$ for every i and $\sum_{i=1}^\ell p_i = 1$), we define $\sum_{i=1}^\ell p_i f_i$ to be the probabilistic function $X \to Y$ corresponding to the matrix $(\sum_{i=1}^\ell p_i p(f_i)_{y,x})_{x \in X, y \in Y}$, namely, the value $(\sum_{i=1}^\ell p_i f_i)(x)$ for $x \in X$ is determined by first choosing f_{i_0} according to the probability distribution $(p_i)_{i=1}^\ell$ and secondly calculating $f_{i_0}(x)$. Such a function $\sum_{i=1}^\ell p_i f_i$ is called a *probabilistic ensemble* of the functions f_i.

For two probabilistic functions $f, g\colon X \to Y$, we define the distance $d(f,g)$ between f and g by

$$d(f,g) := \max_{x\in X, y\in Y} |p(f)_{y,x} - p(g)_{y,x}| \ . \tag{3}$$

Then for two sets $\mathcal{C}, \mathcal{C}'$ of probabilistic functions $X \to Y$, we define

$$r(\mathcal{C},\mathcal{C}') := \sup_{f\in\mathcal{C}} \inf_{g\in\mathcal{C}'} d(f,g) \ . \tag{4}$$

Intuitively, the quantity $r(\mathcal{C},\mathcal{C}')$ measures the approximation error of each member of $\mathcal{C}$ approximated by a suitably chosen member of $\mathcal{C}'$. Note that we have $r(\mathcal{C},\mathcal{C}') = 0$ if $\mathcal{C} \subset \mathcal{C}'$.

2.2 Illustrating example

Here we explain our proposed proof technique to prove information-theoretic security for a pseudorandomized scheme, by applying it to an example of a typical situation for security evaluation. We consider the following kind of security game associated to a certain security notion for a cryptographic scheme. Although the following argument looks like a toy example, the proof technique itself can be applied to more general and various situations.

The security game is described as follows. First, a protocol Π runs to generate an object s which should be concealed from the adversary, and during the execution of Π, the adversary receives some information denoted by x. Secondly, the adversary tries to guess a certain property of the secret s from the object x, by using an algorithm A to output the guess g. Whether the adversary's guess g hits the property of s (denoted by '1') or not (denoted by '0') is evaluated by an algorithm Eval, namely $\mathsf{Eval}(g,s) \in \{0,1\}$. In this setting, when (a part of) the random object that controls the probabilistic behavior of Π is provided by a random source $\mathcal{R}$, the success probability $\mathsf{Succ}(\mathsf{A};\mathcal{R})$ of the adversary is given by

$$\mathsf{Succ}(\mathsf{A};\mathcal{R}) := Pr[r \leftarrow \mathcal{R}; (s,x) \leftarrow \Pi(r); g \leftarrow \mathsf{A}(x); b \leftarrow \mathsf{Eval}(g,s) : b = 1] \tag{5}$$

(see Figure 1 for a picture of the security game). Here, in order to focus on the random source $\mathcal{R}$, a random output r of $\mathcal{R}$ used in the protocol Π is regarded as the input for Π, and any original input for Π given independently of r (if exists) is regarded as an implicit parameter for Π.

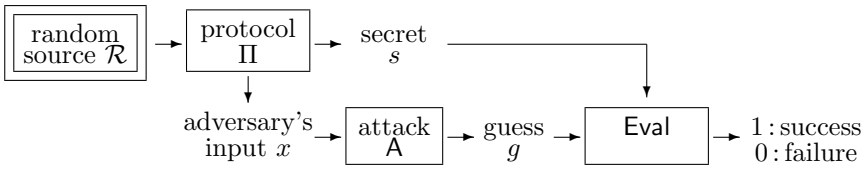

Figure 1: Security game in the example

Let R, S, X and G denote the set of possible choices of the objects r, s, x and g, respectively. Let $\mathcal{A}$ denote the set of the possible attack algorithms $\mathsf{A}\colon X \to G$ considered by the security notion. If we are discussing information-theoretic security, the definition of $\mathcal{A}$ is not relevant to any constraint on computational costs (despite that an attack algorithm may be ruled out by some other reason, e.g., Marking Assumption for fingerprint codes [2]). Now the protocol Π is regarded as secure in the current sense, if $\mathsf{Succ}(\mathsf{A};\mathcal{R})$ is sufficiently small for every attack algorithm $\mathsf{A} \in \mathcal{A}$ and an ideal random source $\mathcal{R}$ on the set R. (Note that our present argument is based on exact security such as (t,ϵ)-security, rather than the usual asymptotic security. See also a related discussion in Section 3.)

The aim of this paper is to present a proof technique to prove that the attack success probability $\mathsf{Succ}(\mathsf{A};\mathcal{R})$ for any $\mathsf{A} \in \mathcal{A}$ is almost unchanged when the ideal random source $\mathcal{R}$ is replaced with a pseudorandom source $\mathcal{R}'$, i.e., the protocol with the pseudorandom source $\mathcal{R}'$ also has the desired security. For the computational indistinguishability of $\mathcal{R}'$, we put the following assumption:

Definition 1. In the above setting, let Class be a class of algorithms (distinguishers) $\mathsf{D}\colon R \to \{0,1\}$ for the pseudorandom source $\mathcal{R}'$, and let $\varepsilon_{\mathsf{Class}} > 0$. We say that $\mathcal{R}'$ is $(\mathsf{Class}, \varepsilon_{\mathsf{Class}})$-*secure*, if for any distinguisher $\mathsf{D} \in \mathsf{Class}$, the advantage $\mathsf{Adv}_{\mathcal{R}'}(\mathsf{D})$ of D defined by

$$\mathsf{Adv}_{\mathcal{R}'}(\mathsf{D}) := |Pr[1 \leftarrow \mathsf{D}(\mathcal{R})] - Pr[1 \leftarrow \mathsf{D}(\mathcal{R}')]| \tag{6}$$

is not larger than $\varepsilon_{\mathsf{Class}}$.

Usually, such a class Class of distinguishers is defined in terms of some kind of computational cost, i.e., a distinguisher D belongs to Class if and only if the computational cost of D is lower than a certain specified level. For example, if Class is the class of distinguishers with computational time shorter than T, then the above security notion is the same as the (T, ε)-security of PRGs appeared sometimes in the literature (e.g., [4]). We assume that $\mathcal{R}'$ is $(\mathsf{Class}, \varepsilon_{\mathsf{Class}})$-secure for a certain specified class Class of distinguishers. We introduce some more definitions and notations. Let $X = \bigcup_{i=1}^{n} X_i$ be a partition of X into disjoint subsets $X_1, \ldots, X_n$. For each index $1 \leq i \leq n$, let $\mathcal{C}_i$ be the set of all probabilistic functions $F_{\mathsf{A}}|_{X_i}\colon X_i \to G$, where "$|_{X_i}$" denotes the restriction of a function on X_i. For any finite set $\mathcal{B}_i$ of probabilistic functions $X_i \to G$, let $\mathcal{C}'(\mathcal{B}_i)$ denote the set of all probabilistic ensembles $\sum_{f \in \mathcal{B}_i} p_f f$ of the members of $\mathcal{B}_i$. Moreover, for each probabilistic function $f\colon X_i \to G$, let $\mathsf{Game}_{i,f}$ denote the probabilistic function $R \to \{0,1\}$ whose value is determined in the following manner:

1. Given $r \in R$, first calculate $(x, s) := \Pi(r)$, and if $x \notin X_i$ then let the final output be 0.
2. If $x \in X_i$, then calculate $g := f(x)$, and let the final output be $\mathsf{Eval}(g, s) \in \{0,1\}$.

Now we have the following result:

Theorem 1. *In the above setting, suppose that for each index $1 \leq i \leq n$, there exists a finite set $\mathcal{B}_i$ of probabilistic functions $X_i \to G$ with the following properties:*

- *For each i and $f \in \mathcal{B}_i$, there exists an algorithm $\mathsf{A}_{i,f}$ satisfying that $F_{\mathsf{A}_{i,f}} = \mathsf{Game}_{i,f}$ and $\mathsf{A}_{i,f} \in \mathsf{Class}$.*
- *For each i, we have $r(\mathcal{C}_i, \mathcal{C}'(\mathcal{B}_i)) \leq \delta$ for a common constant $\delta \geq 0$*

(note that $\mathcal{C}'(\mathcal{B}_i)$ may have a member not belonging to $\mathcal{C}_i$). Then for any $\mathsf{A} \in \mathcal{A}$, we have

$$|\mathsf{Succ}(\mathsf{A}; \mathcal{R}) - \mathsf{Succ}(\mathsf{A}; \mathcal{R}')| \leq 2|G| \cdot \delta + n \cdot \varepsilon_{\mathsf{Class}} \ . \tag{7}$$

Proof. First, for any algorithm $\mathsf{A} \in \mathcal{A}$, we have

$$\begin{aligned}
\mathsf{Succ}(\mathsf{A}; \mathcal{R}) &= \sum_{(r,s,x,g) \in R \times S \times X \times G} Pr[r \leftarrow \mathcal{R}] Pr[(s,x) \leftarrow \Pi(r)] Pr[g \leftarrow \mathsf{A}(x)] Pr[1 \leftarrow \mathsf{Eval}(g,s)] \\
&= \sum_{i=1}^{n} \sum_{(r,s,x,g) \in R \times S \times X_i \times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} p(F_{\mathsf{A}})_{g,x} p(F_{\mathsf{Eval}})_{1,(g,s)} \\
&= \sum_{i=1}^{n} \sum_{(r,s,x,g) \in R \times S \times X_i \times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} p(F_{\mathsf{A}}|_{X_i})_{g,x} p(F_{\mathsf{Eval}})_{1,(g,s)} \ .
\end{aligned} \tag{8}$$

For each $1 \leq i \leq n$, we have $\inf_{g \in \mathcal{C}'(\mathcal{B}_i)} d(F_{\mathsf{A}}|_{X_i}, g) \leq \delta$ by the assumption $r(\mathcal{C}_i, \mathcal{C}'(\mathcal{B}_i)) \leq \delta$. Therefore, for an arbitrary $\delta' > \delta$, there exists a probability distribution $(q_f)_{f \in \mathcal{B}_i}$ on $\mathcal{B}_i$ satisfying that $d(F_{\mathsf{A}}|_{X_i}, \sum_{f \in \mathcal{B}_i} q_f f) < \delta'$, i.e., $|p(F_{\mathsf{A}}|_{X_i})_{g,x} - p(\sum_{f \in \mathcal{B}_i} q_f f)_{g,x}| < \delta'$ for every $x \in X_i$ and $g \in G$. Now we put

$$\mathsf{Succ}'(\mathcal{R}) := \sum_{i=1}^{n} \sum_{(r,s,x,g) \in R \times S \times X_i \times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} p(\sum_{f \in \mathcal{B}_i} q_f f)_{g,x} p(F_{\mathsf{Eval}})_{1,(g,s)} \ . \tag{9}$$

Then by the triangle inequality, we have

$$\begin{aligned}
&|\mathsf{Succ}(\mathsf{A};\mathcal{R}) - \mathsf{Succ}'(\mathcal{R})| \\
&\leq \sum_{i=1}^{n} \sum_{(r,s,x,g)\in R\times S\times X_i\times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} \cdot \left| p(F_\mathsf{A}|_{X_i})_{g,x} - p(\sum_{f\in\mathcal{B}_i} q_f f)_{g,x} \right| \cdot p(F_{\mathsf{Eval}})_{1,(g,s)} \\
&\leq \sum_{i=1}^{n} \sum_{(r,s,x,g)\in R\times S\times X_i\times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} \cdot \delta' \cdot p(F_{\mathsf{Eval}})_{1,(g,s)} \qquad (10) \\
&= \delta' \sum_{(r,s,x,g)\in R\times S\times X\times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} p(F_{\mathsf{Eval}})_{1,(g,s)} \\
&\leq \delta' \sum_{(r,s,x,g)\in R\times S\times X\times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} = \delta' \sum_{(r,g)\in R\times G} Pr[r \leftarrow \mathcal{R}] = \delta' \sum_{g\in G} 1 = |G| \cdot \delta' \ .
\end{aligned}$$

When we consider $\mathcal{R}'$ instead of $\mathcal{R}$, the same argument implies that $|\mathsf{Succ}(\mathsf{A};\mathcal{R}') - \mathsf{Succ}'(\mathcal{R}')| \leq |G| \cdot \delta'$. On the other hand, by the definition of probabilistic ensembles of probabilistic functions, we have

$$\begin{aligned}
\mathsf{Succ}'(\mathcal{R}) &= \sum_{i=1}^{n} \sum_{(r,s,x,g)\in R\times S\times X_i\times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} \left(\sum_{f\in\mathcal{B}_i} q_f p(f)_{g,x} \right) p(F_{\mathsf{Eval}})_{1,(g,s)} \\
&= \sum_{i=1}^{n} \sum_{f\in\mathcal{B}_i} q_f \sum_{(r,s,x,g)\in R\times S\times X_i\times G} Pr[r \leftarrow \mathcal{R}] p(F_\Pi)_{(s,x),r} p(f)_{g,x} p(F_{\mathsf{Eval}})_{1,(g,s)} \qquad (11) \\
&= \sum_{i=1}^{n} \sum_{f\in\mathcal{B}_i} q_f Pr[1 \leftarrow \mathsf{Game}_{i,f}(\mathcal{R})]
\end{aligned}$$

and similarly $\mathsf{Succ}'(\mathcal{R}') = \sum_{i=1}^{n} \sum_{f\in\mathcal{B}_i} q_f Pr[1 \leftarrow \mathsf{Game}_{i,f}(\mathcal{R}')]$. Therefore we have

$$\begin{aligned}
|\mathsf{Succ}'(\mathcal{R}) - \mathsf{Succ}'(\mathcal{R}')| &\leq \sum_{i=1}^{n} \sum_{f\in\mathcal{B}_i} q_f |Pr[1 \leftarrow \mathsf{Game}_{i,f}(\mathcal{R})] - Pr[1 \leftarrow \mathsf{Game}_{i,f}(\mathcal{R}')]| \\
&= \sum_{i=1}^{n} \sum_{f\in\mathcal{B}_i} q_f \mathsf{Adv}_{\mathcal{R}'}(\mathsf{A}_{i,f}) \ ,
\end{aligned} \qquad (12)$$

where $\mathsf{A}_{i,f}$ is the algorithm specified in the statement associated to $\mathsf{Game}_{i,f}$. As $\mathcal{R}'$ is assumed to be $(\mathsf{Class}, \varepsilon_{\mathsf{Class}})$-secure, the assumption $\mathsf{A}_{i,f} \in \mathsf{Class}$ implies that

$$|\mathsf{Succ}'(\mathcal{R}) - \mathsf{Succ}'(\mathcal{R}')| \leq \sum_{i=1}^{n} \sum_{f\in\mathcal{B}_i} q_f \varepsilon_{\mathsf{Class}} = \varepsilon_{\mathsf{Class}} \sum_{i=1}^{n} \sum_{f\in\mathcal{B}_i} q_f = \varepsilon_{\mathsf{Class}} \sum_{i=1}^{n} 1 = n \cdot \varepsilon_{\mathsf{Class}} \ . \qquad (13)$$

Hence, by the triangle inequality and the above results, we have

$$\begin{aligned}
&|\mathsf{Succ}(\mathsf{A};\mathcal{R}) - \mathsf{Succ}(\mathsf{A};\mathcal{R}')| \\
&\leq |\mathsf{Succ}(\mathsf{A};\mathcal{R}) - \mathsf{Succ}'(\mathcal{R})| + |\mathsf{Succ}'(\mathcal{R}) - \mathsf{Succ}'(\mathcal{R}')| + |\mathsf{Succ}'(\mathcal{R}') - \mathsf{Succ}(\mathsf{A};\mathcal{R}')| \qquad (14) \\
&\leq |G| \cdot \delta' + n \cdot \varepsilon_{\mathsf{Class}} + |G| \cdot \delta' = 2|G| \cdot \delta' + n \cdot \varepsilon_{\mathsf{Class}} \ .
\end{aligned}$$

Finally, by the fact that the value $\delta' > \delta$ is arbitrarily chosen, it follows that $|\mathsf{Succ}(\mathsf{A};\mathcal{R}) - \mathsf{Succ}(\mathsf{A};\mathcal{R}')| \leq 2|G| \cdot \delta + n \cdot \varepsilon_{\mathsf{Class}}$, as desired. This concludes the proof of Theorem 1. □

We explain an intuition behind the above proof technique. A main idea is to approximate each attack algorithm (which may be computationally unbounded) by a probabilistic ensemble of a common collection of

algorithms, which we call basic attack algorithms in this discussion. The parameter δ in the statement of the theorem bounds the approximation errors. First we consider the simplest case that each attack algorithm is precisely expressed, not just approximated, by such a probabilistic ensemble (i.e., $\delta = 0$). In this case, it is naively expected that the computational cost of the original attack algorithm is composed of the following two parts; the cost of sampling the probability distribution associated to the probabilistic ensemble to choose a basic attack algorithm, and the cost of computing each basic attack algorithm. The former cost varies according to the arbitrary choice of the original attack algorithm, hence may be unbounded, while the latter cost is independent of the original attack algorithm, hence is bounded. Now the above proof shows that, when evaluating the difference of the attack success probabilities between the cases of the ideal random source $\mathcal{R}$ and of the pseudorandom source $\mathcal{R}'$, the former computational cost (which is the unbounded part of the total cost) is in fact not relevant to the evaluation result. Therefore, as the remaining part of the computational cost is bounded, the computational indistinguishability of $\mathcal{R}'$ becomes sufficient to bound the difference of the attack success probabilities even if the original attack algorithm is computationally unbounded.

For the latter computational cost mentioned above, the cost for the basic attack algorithms is expected to have positive correlation to the size of the domain X of the algorithms (for example, in the smallest case $|X| = 1$, the basic attack algorithms can be defined to be constant functions; see Section 2.3). Therefore, in order to decrease the cost for the basic attack algorithms, we divided the domain X into smaller parts $X_1, \ldots, X_n$ and considered the basic attack algorithms with each smaller domain X_i. This idea introduced a trade-off between the cost for the basic attack algorithms and the size n of the partition of X.

For a general case that the probabilistic ensembles of the basic attack algorithms are just approximations of arbitrary attack algorithms (i.e., $\delta > 0$), the derived bound for the difference of the attack success probabilities involves additional term $2|G| \cdot \delta$. For example, when the security notion under consideration is some kind of indistinguishability, we usually have $|G| = 2$ and hence the additional term will be significantly small if δ is sufficiently small. On the other hand, in general, the size of G may be very large. In such a case, it is desirable to let the parameter δ being 0, which makes the size of G irrelevant to the bound in our result. For the purpose, we present the following corollary of the above theorem:

Corollary 1. *In the above setting, suppose that for every $1 \leq i \leq n$ and every deterministic function $f\colon X_i \to G$, there exists an algorithm $\mathsf{A}_{i,f}$ satisfying that $F_{\mathsf{A}_{i,f}} = \mathsf{Game}_{i,f}$ and $\mathsf{A}_{i,f} \in \mathsf{Class}$. Then for any $\mathsf{A} \in \mathcal{A}$, we have*

$$|\mathsf{Succ}(\mathsf{A}; \mathcal{R}) - \mathsf{Succ}(\mathsf{A}; \mathcal{R}')| \leq n \cdot \varepsilon_{\mathsf{Class}} \ . \tag{15}$$

Proof. For each $1 \leq i \leq n$, let $\mathcal{B}_i$ be the set of all deterministic functions $X_i \to G$. Then for any algorithm $\mathsf{A} \in \mathcal{A}$, the probabilistic function $F_{\mathsf{A}}|_{X_i}\colon X_i \to G$ can be expressed as a probabilistic ensemble of the members of $\mathcal{B}_i$; the coefficient of $f \in \mathcal{B}_i$ in the expression of $F_{\mathsf{A}}|_{X_i}$ is given by $q_f := \prod_{x \in X_i} p(F_{\mathsf{A}}|_{X_i})_{f(x),x}$. Indeed, for each $x \in X_i$ and $g \in G$, we have

$$\begin{aligned} p(\sum_{f \in \mathcal{B}_i} q_f f)_{g,x} = \sum_{f \in \mathcal{B}_i} q_f p(f)_{g,x} &= \sum_{f \in \mathcal{B}_i \,;\, f(x)=g} q_f \\ &= p(F_{\mathsf{A}}|_{X_i})_{g,x} \sum_{f \in \mathcal{B}_i \,;\, f(x)=g} \prod_{y \in X_i \setminus \{x\}} p(F_{\mathsf{A}}|_{X_i})_{f(y),y} \\ &= p(F_{\mathsf{A}}|_{X_i})_{g,x} \prod_{y \in X_i \setminus \{x\}} \sum_{f(y) \in G} p(F_{\mathsf{A}}|_{X_i})_{f(y),y} \\ &= p(F_{\mathsf{A}}|_{X_i})_{g,x} \prod_{y \in X_i \setminus \{x\}} 1 = p(F_{\mathsf{A}}|_{X_i})_{g,x} \ . \end{aligned} \tag{16}$$

This implies that $r(\mathcal{C}_i, \mathcal{C}'(\mathcal{B}_i)) = 0$, therefore the second condition in Theorem 1 is satisfied by putting $\delta := 0$. Hence the claim follows from Theorem 1. □

Remark 1. The former of the two conditions in the statement of Theorem 1 implies that the protocol Π itself can be executed in sufficiently low computational cost to let the function $\mathsf{Game}_{i,f}$ (involving Π as a part)

be computable by an algorithm in Class. We emphasize that this condition does *not* mean that the protocol should always be implemented in such a low-cost manner in a practical application; the above condition requires only *existence* of such a low-cost implementation. Similarly, the condition in Theorem 1 also implies that the (in)correctness of the adversary's guess g about the secret s can be efficiently checked.

2.3 Comparison to preceding results

Here we discuss differences of our result above from the preceding results on pseudorandomization of information-theoretically secure schemes. First, our result above is a generalization and an improvement of the preceding result of Nuida and Hanaoka [11, 12]. Indeed, when we use the partition $X = \bigcup_{x \in X} \{x\}$ of the set X into subsets with single elements (hence $n = |X|$), the result of Corollary 1 in this case coincides with the one by their result. In the preceding result, the size n of the domain X of attack algorithms is required to be significantly small due to the term $n \cdot \varepsilon_{\mathsf{Class}}$ of the inequality in the theorem. Intuitively, our improved technique introduces a trade-off between the size n of the partition of X and the costs for the basic attack algorithms, which can reduce the constraint on the (small) size of X. Owing to the improvement, our proof technique would be applicable to more various situations than the technique in [11, 12].

On the other hand, a method for pseudorandomization of some kinds of information-theoretically secure schemes was also presented by Dubrov and Ishai [3], by introducing an extended notion for PRGs (that is, PRGs against distinguishers whose output sets consist of more than two but bounded numbers of elements). When applying their method to the above example, the protocol Π is regarded as the extended distinguisher for the pseudorandom source $\mathcal{R}'$, in order to show that the distributions of the outputs (s, x) of Π themselves in the random and pseudorandom cases are statistically almost equal, resulting in almost equal attack success probabilities even by computationally unbounded attack algorithms. However, the argument requires the size of the output set $S \times X$ of the "distinguisher" Π, in particular the size of S, to be significantly small, which seems frequently not practical (as a secret s is frequently chosen from a large number of candidates to achieve a desired security against brute-force attacks). In contrast, our proof technique does not require any constraint on the size of S, therefore the potential applications of our technique are wider than those of the method in [3].

3 On types of distinguishers for the PRGs

In the preceding result by Nuida and Hanaoka [11, 12], it was mainly supposed that the PRGs used in the pseudorandomization are of the following type: The class of distinguishers in the definition of indistinguishability are determined in terms of constraints on time complexity, and the formulation of indistinguishability is based on exact security rather than asymptotic security. There are in fact two drawbacks in the choice of PRGs. First, the existing PRGs against those time-bounded distinguishers require some hardness assumptions (not only the computational constraints on distinguishers) to prove their indistinguishability. As a result, if we use such kinds of PRGs in our pseudorandomization of information-theoretically secure schemes, then the proof requires some hardness assumption even though there are no computational constraints on the attack algorithms, which somewhat spoils the advantage of information-theoretic security. Secondly, for PRGs against time-bounded distinguishers, the formulation of indistinguishability based on asymptotic security rather than exact security has mainly been adopted in the literature. When we want to convert the result based on exact security into asymptotic security, we will face a gap between non-uniform and uniform models of computation (cf., an article [7] by Koblitz and Menezes). Namely, in the case that the security game under discussion is scalable with respect to security parameter 1^k, it would be possible that there is a basic attack algorithm (i.e., a member of $\mathcal{B}_i$) for each parameter 1^k for which the difference of attack success probabilities between random and pseudorandom cases is not significantly small, but there does *not* exist an efficient *uniform* algorithm that agrees with the undesired basic attack algorithm at each parameter 1^k (in other words, it may happen that an undesired algorithm exists at each parameter 1^k, but these cannot be unified as an efficient uniform algorithm). If it happens, then even provable indistinguishability of a PRG cannot rule out such undesired basic attack algorithms, which prevents us to transfer from

exact security to asymptotic security.

To resolve the above two drawbacks, in this paper we propose to use PRGs of different types rather than those against time-bounded distinguishers. More precisely, we use PRGs against distinguishers whose constraints are determined by the amount of memories (instead of computational times). A concrete example of such PRGs are the one proposed by Impagliazzo, Nisan and Wigderson [6], which is henceforth called *INW PRG*. A central advantage of INW PRG is that the indistinguishability against memory-bounded distinguishers can be proven in exact security formulation without any hardness assumption. This property fits our result well, and by combining the PRG to our result, pseudorandomization of information-theoretically secure schemes without any hardness assumption is achieved. In the following sections, we investigate a concrete computational model suitable for describing memory-bounded distinguishers for INW PRGs, for the purpose of quantitative evaluation of the performance of our proof technique, i.e., appropriate seed lengths for the PRGs used in the pseudorandomization.

Remark 2. One may think that the PRGs (more precisely, PRGs in the extended sense mentioned in Section 2.3) given in Section 3.2.1 of [3], whose indistinguishability against distinguishers with constant-depth circuits was proven without any hardness assumption, can be used in our result instead of INW PRGs. However, as the indistinguishability of those PRGs was considered in the form of asymptotic security, we will face the same problem as the above case of PRGs against time-bounded distinguishers about the gap between uniform and non-uniform computational models. On the other hand, if the indistinguishability of some other PRG (against a class of distinguishers) can be proven without any hardness assumption in the form of exact security, then such a PRG can be combined with our proof technique as well as INW PRGs.

4 The computational model

For quantitative evaluation of the indistinguishability of INW PRGs against memory-bounded distinguishers, a computational model was introduced by Impagliazzo, Nisan and Wigderson [6] (see also papers by Babai, Nisan and Szegedy [1] and by Nisan [9]). As the description of the model in the paper [6] is not enough rigorous to perform detailed evaluations, in this section we give a formal description of this model.

In the computational model, which we call the *bucket-brigade model*, an execution of an algorithm using a random input tape is interpreted as a multi-party protocol played by the cells of the random tape; the computation process since the content of j-th cell is read until the content of $(j+1)$-th cell is read is interpreted as a local computation by j-th cell and a communication from j-th cell to $(j+1)$-th cell, where the communicated message represents the intermediate computation result stored in the memory at the time of reading the content of $(j+1)$-th cell. Figure 2 shows an illustrated example of the bucket-brigade model. To rigorously deal with the bucket-brigade model in terms of multi-party protocols, in the following subsections we summarize a formalization of the concept of multi-party protocols and present a precise definition of the bucket-brigade model.

4.1 Notations and terminology

Here we summarize some notations and terminology used below. Unless otherwise specified, each (undirected) graph and directed graph are finite and simple (i.e., without self-loops and multiple edges). We write $[n] := \{1, 2, \ldots, n\}$ for an integer n. For sets X and Y, let $f : X \twoheadrightarrow Y$ mean that f is a partial map from X to Y, that is, f is a map with range $\mathrm{ran}(f) = Y$ and the domain $\mathrm{dom}(f)$ of f is a subset of X (including the simplest case $\mathrm{dom}(f) = X$).

4.2 Multi-party protocols

This subsection summarizes a formulation of the concept of multi-party protocols adopted in this paper. We also give a lemma for multi-party protocols used later.

Let Π be a μ-party protocol, consisting of *input phase*, *communication phase* and *output phase*. We use the following notations. Let $P_1, \ldots, P_\mu$ denote the μ players of Π, and put $\mathcal{P}(\Pi) := \{P_1, \ldots, P_\mu\}$. Let $\mathcal{I}_i$

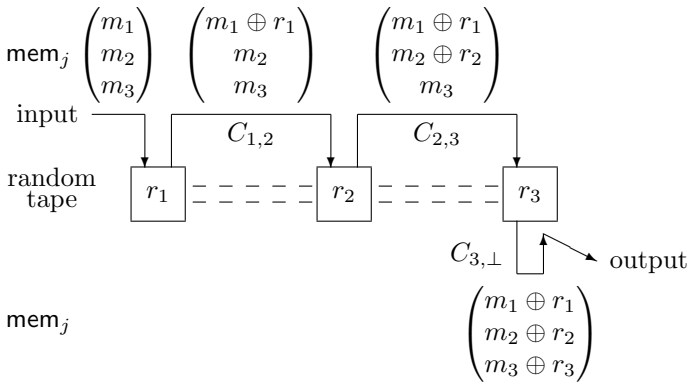

Figure 2: Algorithm $\mathsf{A}(m) = m \oplus r$, $m \in \{0,1\}^3$, $r \leftarrow_R \{0,1\}^3$ described in the bucket-brigade model (see Section 4.3 for some notations)

and $\mathcal{O}_i$ be the *input space* and the *output space*, respectively, for i-th player P_i. A *joint input space* $\mathcal{I}$ is determined as a subset of $\mathcal{I}_1 \times \cdots \times \mathcal{I}_\mu$. We say that a subset $\mathcal{I}' \subset \mathcal{I}$ is *rectangular* if $\mathcal{I}' = \mathcal{I}'_1 \times \cdots \times \mathcal{I}'_\mu$ for some subsets $\mathcal{I}'_i \subset \mathcal{I}_i$, $i \in [\mu]$. Let $\mathcal{M}$ denote the common *message space* used during the communication phase. Let $\mathcal{C}$ be the set of *communication channels* (or simply *channels*); each $C \in \mathcal{C}$ is a channel from a single *source* $\mathsf{source}(C) \in \mathcal{P}(\Pi)$ to the non-empty set of *targets* $\mathsf{target}(C) \subset \mathcal{P}(\Pi)$. For each index $i \in [\mu]$, we define

$$\mathcal{C}_{i\to} := \{C \in \mathcal{C} \mid \mathsf{source}(C) = P_i\}\,,\ \mathcal{C}_{\to i} := \{C \in \mathcal{C} \mid P_i \in \mathsf{target}(C)\}\ , \tag{17}$$

i.e., the outgoing and incoming channels for player P_i. Moreover, let $L_i = (L_{i,0}; L_{i,1}; L_{i,2}; \cdots)$ denote the *communication log* for player P_i, updated stepwise during the protocol.

In the **input phase**, a μ-tuple $x = (x_1, \ldots, x_\mu) \in \mathcal{I}$ of local inputs x_i for players P_i is chosen in a certain specified manner. Then each communication log L_i is initialized as $L_i := (L_{i,0})$ with $L_{i,0} := x_i$.

The j-th step ($j \geq 1$) of the **communication phase** proceeds as follows:

1. Each player P_i chooses a collection $m^{\mathrm{out}}_{i,j} \colon \mathcal{C}_{i\to} \twoheadrightarrow \mathcal{M}$ of outgoing messages from P_i during the step. That is, P_i uses the channels in the (possibly empty) subset $\mathrm{dom}(m^{\mathrm{out}}_{i,j}) \subset \mathcal{C}_{i\to}$, and for each channel $C \in \mathrm{dom}(m^{\mathrm{out}}_{i,j})$ the message $m^{\mathrm{out}}_{i,j}(C) \in \mathcal{M}$ is sent to every player in $\mathsf{target}(C)$. We suppose that the (possibly probabilistic) choice of $m^{\mathrm{out}}_{i,j}$ depends solely on the 0-th to $(j-1)$-th parts of L_i.

2. Each player P_i consequently receives a collection $m^{\mathrm{in}}_{i,j} \colon \mathcal{C}_{\to i} \twoheadrightarrow \mathcal{M}$ of incoming messages to P_i. That is, $\mathrm{dom}(m^{\mathrm{in}}_{i,j}) := \mathcal{C}_{\to i} \cap \bigcup_{i'} \mathrm{dom}(m^{\mathrm{out}}_{i',j})$, and $m^{\mathrm{in}}_{i,j}(C) := m^{\mathrm{out}}_{\mathsf{source}(C),j}(C)$ for each $C \in \mathrm{dom}(m^{\mathrm{in}}_{i,j})$.

3. The communication log L_i for each P_i is updated by appending a new component $L_{i,j} := (m^{\mathrm{out}}_{i,j}, m^{\mathrm{in}}_{i,j})$ to the last; $L_i := (L_{i,0}; \cdots; L_{i,j-1}; L_{i,j})$.

4. If $\mathrm{dom}(m^{\mathrm{out}}_{i,j}) = \emptyset$ for every $i \in [\mu]$, i.e., no messages are communicated during the present step, then the communication phase halts (note that each $\mathrm{dom}(m^{\mathrm{in}}_{i,j})$ is also empty in this case).

We assume, unless otherwise specified, that Π always halts within a finite number of steps.

Finally, in the **output phase**, each player P_i chooses a local output $y_i \in \mathcal{O}_i$ according to a (possibly probabilistic) manner depending solely on P_i's communication log L_i. Then the output of the protocol Π is the μ-tuple $y = (y_1, \ldots, y_\mu)$, denoted by $y = \Pi(x) = \Pi(x_1, \ldots, x_\mu)$.

In the above setting, we introduce the following definitions:

Definition 2. First, we divide each $m^{\mathrm{out}}_{i,j}$ defined in the communication phase into $m^{\mathrm{out.self}}_{i,j}$ and $m^{\mathrm{out.other}}_{i,j}$ in such a way that

$$\mathrm{dom}(m^{\mathrm{out.self}}_{i,j}) = \{C \in \mathrm{dom}(m^{\mathrm{out}}_{i,j}) \mid \mathsf{target}(C) = \{i\}\}\,,\ \mathrm{dom}(m^{\mathrm{out.other}}_{i,j}) = \mathrm{dom}(m^{\mathrm{out}}_{i,j}) \setminus \mathrm{dom}(m^{\mathrm{out.self}}_{i,j})\ . \tag{18}$$

Then we define the *conditional communication capacity* $c(\Pi|\mathcal{I}')$ of Π conditioned on a subset $\mathcal{I}' \subset \mathcal{I}$ (or the *communication capacity* $c(\Pi)$ of Π, when $\mathcal{I}' = \mathcal{I}$) to be the number of possible matrices $m^{\mathrm{out.other}} := (m_{i,j}^{\mathrm{out.other}})_{i,j}$ of outgoing messages during the communication phase, with input for Π chosen from $\mathcal{I}'$. (Intuitively, it counts the number of possible communication patterns during the protocol.)

Here we present the following lemma, which is an explicitly stated and slightly generalized version of a fact mentioned (without proof) in the proof of Theorem 1 in [6]:

Lemma 1. *Let Π be a μ-party deterministic protocol (i.e., the communication and output phases are not probabilistic). Let $\mathcal{I}' = \mathcal{I}'_1 \times \cdots \times \mathcal{I}'_\mu$ be a rectangular subset of $\mathcal{I}$. Suppose that the number of possible outputs y of Π with input chosen from $\mathcal{I}'$ is not larger than $k \in \mathbb{Z}$. Then there exist subsets $S_{i,j} \subset \mathcal{I}'_i$ for $i \in [\mu]$ and $j \in [kc(\Pi|\mathcal{I}')]$, with the property that $\mathcal{I}'$ is the disjoint union of the subsets $S_{1,j} \times \cdots \times S_{\mu,j}$ over $j \in [kc(\Pi|\mathcal{I}')]$ and the output of Π is constant on each subset $S_{1,j} \times \cdots \times S_{\mu,j}$ of possible inputs.*

Proof. Put $N := c(\Pi|\mathcal{I}')$, and let $m^{(1)}, \ldots, m^{(N)}$ denote the possible matrices $m^{\mathrm{out.other}}$ appeared in the definition of $c(\Pi|\mathcal{I}')$. For each $x = (x_1, \ldots, x_\mu) \in \mathcal{I}'$, let $m^{\mathrm{out}}(x)$ denote the matrix $m^{\mathrm{out}} := (m_{i,j}^{\mathrm{out}})_{i,j}$ arising from the input x for Π (which is uniquely determined, as Π is now deterministic). We define $m^{\mathrm{out.self}}(x)$, $m^{\mathrm{out.other}}(x)$ and $m^{\mathrm{in}}(x)$ similarly. Now for each $i \in [\mu]$, $h \in [N]$ and possible output y of Π, define $S_{i,h,y}$ to be the set of all $x_i \in \mathcal{I}'_i$ satisfying that $m^{\mathrm{out.other}}(x_1, \ldots, x_\mu) = m^{(h)}$ and $\Pi(x_1, \ldots, x_\mu) = y$ for some $x_{i'} \in \mathcal{I}'_{i'}$, $i' \in [\mu] \setminus \{i\}$.

To prove the lemma, it suffices to show that, for each $h \in [N]$ and possible output y,

$$\{x \in \mathcal{I}' \mid m^{\mathrm{out.other}}(x) = m^{(h)} \text{ and } \Pi(x) = y\} = S_{1,h,y} \times \cdots \times S_{\mu,h,y} \tag{19}$$

(note that there are at most k possibilities for y). The inclusion $\subset$ is trivial. For the other direction, let $x_i \in S_{i,h,y}$ for each $i \in [\mu]$, and put $x = (x_1, \ldots, x_\mu)$. Note that $x \in \mathcal{I}'$, as $\mathcal{I}'$ is now rectangular. By the definition of the sets $S_{i,h,y}$, for each i, there exists a $z_i = (z_{i,1}, \ldots, z_{i,\mu}) \in \mathcal{I}'$ satisfying that $z_{i,i} = x_i$, $m^{\mathrm{out.other}}(z_i) = m^{(h)}$ and $\Pi(z_i) = y$. Let $L_i(x)$ and $L_i(z_{i'})$ denote the communication logs for P_i arising from inputs x and $z_{i'}$, respectively. Now we have $L_{i,0}(x) = (x_i) = (z_{i,i}) = L_{i,0}(z_i)$ for every $i \in [\mu]$. Suppose that $L_{i,j'}(x) = L_{i,j'}(z_i)$ for every $i \in [\mu]$ and $0 \leq j' \leq j-1$. Then, as the outgoing messages from each player P_i at j-th step are determined solely by $(L_{i,0}; \cdots; L_{i,j-1})$, we have $m^{\mathrm{out.self}}(x)_{i,j} = m^{\mathrm{out.self}}(z_i)_{i,j}$ for any $i \in [\mu]$, and $m^{\mathrm{out.other}}(x)_{i',j} = m^{\mathrm{out.other}}(z_{i'})_{i',j} = m^{(h)}_{i',j} = m^{\mathrm{out.other}}(z_i)_{i',j}$ for any $i, i' \in [\mu]$. These relations imply that $m^{\mathrm{out}}(x)_{i,j} = m^{\mathrm{out}}(z_i)_{i,j}$ and $m^{\mathrm{in}}(x)_{i,j} = m^{\mathrm{in}}(z_i)_{i,j}$, therefore we have $L_{i,j}(x) = L_{i,j}(z_i)$ for every i. Hence, by induction, we have $L_i(x) = L_i(z_i)$ for every i, and $m^{\mathrm{out}}(x) = m^{(h)}$. This also implies that the local output of each P_i for the case of input x is equal to that for the case of input z_i (i.e., y_i), therefore $\Pi(x) = y$. Hence the proof of Lemma 1 is concluded. □

4.3 Bucket-brigade computational model

From now, we give a precise formulation of the bucket-brigade model for probabilistic algorithms. As mentioned above, this model interprets an algorithm A as a multi-party protocol $\Pi(\mathsf{A})$ played by the cells of the random input tape, in which the intermediate computation results stored in the memory are updated and communicated by the "players" (cells) according to the original algorithm. We define the multi-party protocol $\Pi(\mathsf{A})$ associated to an algorithm A in the following manner (see Figure 2 above for a toy example).

We adopt the following settings. We suppose that the algorithm A uses a random tape with $\mu \geq 2$ cells. The protocol $\Pi(\mathsf{A})$ has μ players P_i ($i \in [\mu]$), each endowed with a random variable r_i on a finite set R_i, which represents the content of i-th cell of the random tape. Moreover, each player P_i is also endowed with his/her own communication log, by which he/she can record what messages he/she received and sent at previous steps. Let M denote the (finite) set of possible states of the memory in the algorithm A, which is regarded as the message space for the protocol $\Pi(\mathsf{A})$. We assume that $|M| \geq 2$, as otherwise the memory has a constant state and hence is useless.

To express the configuration of the random tape used by the algorithm A, we introduce a directed graph $\mathcal{G}(\mathsf{A})$ with vertex set $[\mu]$, called the *configuration graph*. An edge in $\mathcal{G}(\mathsf{A})$ from $i \in [\mu]$ to $i' \in [\mu]$ means that

the tape head of the random tape can move from i-th cell directly to i'-th cell (for example, $\mathcal{G}(\mathsf{A})$ is the path $1 \to 2 \to \cdots \to \mu$ in the case of a read-once one-way random tape). Then we introduce the following two kinds of channels for $\Pi(\mathsf{A})$. A channel $C_{i,i'}$ of the first type with source P_i and target set $\{P_{i'}\}$ is introduced for each edge in $\mathcal{G}(\mathsf{A})$ from $i \in [\mu]$ to $i' \in [\mu]$. On the other hand, a channel $C_{i,\perp}$ of the second type with source P_i and target set $\{P_\mu\}$ is introduced for each $i \in [\mu]$, which will be used in the last of the communication phase of $\Pi(\mathsf{A})$.

In the **input phase** of $\Pi(\mathsf{A})$, each player P_i receives the local input $\overline{r_i} \in R_i$ chosen according to the random variable r_i. Moreover, the first player P_1 also receives an initial state $\mathsf{mem}_0 \in M$ of the memory as input (which depends on the input for the original algorithm A), therefore the local input for P_1 is indeed the pair $(\mathsf{mem}_0, \overline{r_1})$. Hence the joint input space is $\mathcal{I} := M \times R_1 \times \cdots \times R_\mu$ and an input for $\Pi(\mathsf{A})$ is given by $x := (\mathsf{mem}_0, \overline{r_1}, \ldots, \overline{r_\mu})$.

In j-th step ($j \geq 1$) of the **communication phase**, only one player P_{i_j} is active at the step, that is, any other player $P_{i'}$ ($i' \neq i_j$) sends no messages at the step (i.e., $\mathrm{dom}(m^{\mathrm{out}}_{i',j}) = \emptyset$). We suppose that, when $j \geq 2$, the active player P_{i_j} received a message $\mathsf{mem}_{j-1} \in M$ at the previous (i.e., $(j-1)$-th) step. For the first step, we set $i_1 := 1$. Then the j-th step proceeds as follows.

- If $j \geq 2$, $i_j = \mu$ and P_μ received the message mem_{j-1} at $(j-1)$-th step via a channel $C_{i_{j-1},\perp}$ of the second type, then the communication phase is going to halt, hence all players including P_μ send no messages at the step.
- Otherwise, the active player P_{i_j} calculates a new memory state $\mathsf{mem}_j \in M$ from mem_{j-1}, $\overline{r_{i_j}}$ and the communication log of P_{i_j} in a manner specified by the original algorithm A (we emphasize that the local calculation has no computational constraints). Now
  - if the original algorithm A should halt at this stage, then P_{i_j} sets $i_{j+1} := \mu$ and sends the message mem_j via the channel $C_{i_j,\perp}$ of the second type;
  - otherwise, P_{i_j} also determines i_{j+1} for which a channel $C_{i_j,i_{j+1}}$ of the first type exists, and sends the message mem_j via $C_{i_j,i_{j+1}}$.

Naively speaking, boundedness of the memory size $|M|$ implies boundedness of the communication capacity of $\Pi(\mathsf{A})$. See Section 6.1 for more details.

Suppose that the communication phase has halted at j-th step, therefore the last message received by P_μ is $\mathsf{mem}_{j-1} \in M$ at $(j-1)$-th step. Then in the **output phase**, the local outputs of the players other than P_μ are set as empty, while the local output y_μ of P_μ is calculated from mem_{j-1} and the communication log of P_μ in a manner specified by the algorithm A (we emphasize again that the local calculation has no computational constraints). For simplicity, we identify the output y of the protocol $\Pi(\mathsf{A})$ with y_μ rather than the μ-tuple $(\emptyset, \emptyset, \ldots, \emptyset, y_\mu)$.

5 Impagliazzo–Nisan–Wigderson PRGs

In this section, we summarize the construction of INW PRGs given in [6]. The PRGs were first described in the context of pseudorandomization of random inputs for multi-party protocols, and then applied to usual algorithms via the above-mentioned computational model. We also study some properties of INW PRGs.

5.1 Notations and terminology

Here we summarize some notations and terminology used below. For integers $n \geq 0$ and $m \geq 1$, let "$n \bmod m$" denote the remainder of n modulo m, which is taken in such a way that $n \bmod m \in \{0, 1, \ldots, m-1\}$. Then we define an operation mod^{+1}_m on non-negative integers by $\mathrm{mod}^{+1}_m(x) := (x \bmod m) + 1 \in [m]$, $0 \leq x \in \mathbb{Z}$. On the other hand, for integers $n \geq 0$ and $m \geq 1$, we put

$$\rho(n, m) := \frac{(n \bmod m) \cdot (m - (n \bmod m))}{nm} \ . \tag{20}$$

Let U_X denote the uniform random variable over a finite set X. Moreover, let $\Delta(r_1, r_2)$ denote the statistical distance between two random variables r_1, r_2 over a common finite set X:

$$\Delta(r_1, r_2) := \frac{1}{2} \sum_{x \in X} |Pr[x \leftarrow r_1] - Pr[x \leftarrow r_2]| \ . \tag{21}$$

For any undirected graph $\mathcal{G} = (V, E)$, we suppose that some linear orderings on the vertex set V and on the edge set E are implicitly fixed to let expressions such as "k-th vertex" and "k-th edge" make sense. We say that $\mathcal{G}$ is *δ-regular*, if each vertex of $\mathcal{G}$ has precisely δ edges. Let $A(\mathcal{G})$ denote the adjacency matrix of undirected graph $\mathcal{G}$, that is, a $|V| \times |V|$ symmetric $\{0,1\}$-matrix for which the (i,j)-entry is 1 if and only if i-th vertex of $\mathcal{G}$ is adjacent to j-th vertex of $\mathcal{G}$. If $\mathcal{G}$ is a δ-regular graph and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{|V|}$ are the eigenvalues of $A(\mathcal{G})$ (note that $\lambda_1 = \delta$), then we put $\lambda = \lambda(\mathcal{G}) := \max\{|\lambda_2|, |\lambda_{|V|}|\}$ which is called the *second largest eigenvalue* for $\mathcal{G}$.

5.2 A building block

Each INW PRG is constructed by composing smaller building-block PRGs according to some binary tree structure. In this subsection, we describe the construction of the building-block PRGs $g^{\mathcal{G}}_{N_0;N_1,N_2}$. The parameters are chosen as follows: Let $\mathcal{G} = (V, E)$ be a δ-regular (undirected) graph with ν vertices ($|V| = \nu$), and let N_0, N_1 and N_2 be integers satisfying $N_0 \geq \nu\delta$ and $1 \leq N_i \leq \nu$ ($i \in \{1, 2\}$).

Definition 3. In the above setting, the PRG $g^{\mathcal{G}}_{N_0;N_1,N_2}$ is defined as a map $[N_0] \to [N_1] \times [N_2]$, where the value $g^{\mathcal{G}}_{N_0;N_1,N_2}(x)$ for $x \in [N_0]$ is determined in the following manner:

1. Decompose x uniquely as $x = a_1\nu\delta + a_2\delta + a_3$, where a_1, a_2, a_3 are non-negative integers with $a_2 < \nu$ and $a_3 < \delta$.
2. Take a_2-th vertex $v \in V$ of $\mathcal{G}$.
3. Take a_3-th edge $e \in E$ of $\mathcal{G}$ among the δ edges adjacent to v. Let v' denote the other vertex of e.
4. Take the index $a_4 \in \{0, 1, \ldots, \nu - 1\}$ satisfying that v' is a_4-th vertex of $\mathcal{G}$.
5. Finally, set $g^{\mathcal{G}}_{N_0;N_1,N_2}(x) := (\mathrm{mod}^{+1}_{N_1}(a_2), \mathrm{mod}^{+1}_{N_2}(a_4)) \in [N_1] \times [N_2]$ (see Section 5.1 for the notations).

We notice that $g^{\mathcal{G}}_{N_0;N_1,N_2}$ coincides with composition of the following three maps; $\mathrm{mod}^{+1}_{\nu\delta} \colon [N_0] \to [\nu\delta]$, $g^{\mathcal{G}}_{\nu\delta;\nu,\nu} \colon [\nu\delta] \to [\nu] \times [\nu]$, and $\mathrm{mod}^{+1}_{N_1} \times \mathrm{mod}^{+1}_{N_2} \colon [\nu] \times [\nu] \to [N_1] \times [N_2]$.

Roughly speaking, it can be shown *without any hardness assumption* that the PRG $g^{\mathcal{G}}_{N_0;N_1,N_2}$ is indistinguishable against distinguishers given by 2-party protocols with bounded communication capacity. More precisely, we have the following lemma, which is an improvement of Theorem 1 in [6] and Lemma 1 in [6] (namely, the degree 1/2 of the positive integer term $kc(\Pi|\mathcal{I}')$ in the following lemma is lower than the degree 1 derived by the argument of [6]):

Lemma 2. *Let Π be a 2-party protocol, $\mathcal{I}' = \mathcal{I}'_1 \times \mathcal{I}'_2$ be a rectangular subset of the joint input space $\mathcal{I}$ of Π, and let $N \in \mathbb{Z}$. We identify each $\mathcal{I}'_i$ with $[n_i]$, where $n_i := |\mathcal{I}'_i|$. Suppose that $N \geq \nu\delta$, $\nu \geq n_1$ and $\nu \geq n_2$. Suppose further that the number of possible outputs of Π arising from inputs chosen from $\mathcal{I}'$ is not larger than $k \in \mathbb{Z}$. Then we have*

$$\Delta(\Pi(g^{\mathcal{G}}_{N;n_1,n_2}(U_{[N]})), \Pi(U_{[n_1]}, U_{[n_2]})) \leq \frac{\lambda(\mathcal{G})}{2\delta}\sqrt{kc(\Pi|\mathcal{I}')} + \rho(N, \nu\delta) + \rho(\nu, n_1) + \rho(\nu, n_2) \tag{22}$$

(see Section 5.1 for notations).

Proof. First we present the following three lemmas (we notice that Lemma 3 and Lemma 5 are better than the counterparts in the proofs in [6], which result in the above-mentioned improvement in Lemma 2):

Lemma 3 (Expander Mixing Lemma; see e.g., Section 2.4 of [5]). *Let $\mathcal{G} = (V, E)$ be a δ-regular graph with ν vertices. Put $E(S,T) := \{(s,t) \in S \times T \mid s \text{ and } t \text{ are adjacent in } \mathcal{G}\}$ for any subsets $S, T \subset V$. Then*

$$\left| |E(S,T)| - \frac{\delta |S| \cdot |T|}{|V|} \right| \leq \lambda(\mathcal{G}) \sqrt{|S| \cdot |T|} \ . \tag{23}$$

Lemma 4 (see Lemma 1 of [11]). *For integers $n \geq 0$ and $m \geq 1$, we have $\Delta(\mathrm{mod}_m^{+1}(U_{[n]}), U_{[m]}) = \rho(n,m)$.*

Lemma 5. *For any $x_1, \ldots, x_n \geq 0$, we have $\sum_{i=1}^n x_i \leq \sqrt{n \sum_{i=1}^n {x_i}^2}$.*

Proof. We have

$$\begin{aligned} n \sum_{i=1}^n {x_i}^2 - \left(\sum_{i=1}^n x_i \right)^2 &= (n-1) \sum_{i=1}^n {x_i}^2 - 2 \sum_{1 \leq i < j \leq n} x_i x_j \\ &= \sum_{i=1}^{n-1} \left((n-i) {x_i}^2 + \sum_{\ell=i+1}^n {x_\ell}^2 \right) - 2 \sum_{1 \leq i < j \leq n} x_i x_j \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n ({x_i}^2 + {x_j}^2 - 2 x_i x_j) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (x_i - x_j)^2 \geq 0 \ , \end{aligned} \tag{24}$$

therefore $\left(\sum_{i=1}^n x_i\right)^2 \leq n \sum_{i=1}^n {x_i}^2$, as desired. □

To prove Lemma 2, we may assume without loss of generality that the 2-party protocol Π is deterministic, as a general Π can be expressed as a probabilistic ensemble of deterministic ones. For simplicity, we put

$$\sigma_1 := \mathrm{mod}_{n_1}^{+1} \ , \ \sigma_2 := \mathrm{mod}_{n_2}^{+1} \ , \ \sigma := \sigma_1 \times \sigma_2 \ , \ \tau := \mathrm{mod}_{\nu\delta}^{+1} \ . \tag{25}$$

Moreover, let Π^σ denote the 2-party deterministic protocol that, given an input $(a_1, a_2) \in [\nu] \times [\nu]$, simulates the protocol Π with input $\sigma(a_1, a_2) \in \mathcal{I}'$. Then by the relation $g^{\mathcal{G}}_{N;n_1,n_2} = \sigma \circ g^{\mathcal{G}}_{\nu\delta;\nu,\nu} \circ \tau$, we have

$$\begin{aligned} &\Delta(\Pi(g^{\mathcal{G}}_{N;n_1,n_2}(U_{[N]})), \Pi(U_{[n_1]}, U_{[n_2]})) \\ &\leq \Delta(\Pi(g^{\mathcal{G}}_{N;n_1,n_2}(U_{[N]})), \Pi((\sigma \circ g^{\mathcal{G}}_{\nu\delta;\nu,\nu})(U_{[\nu\delta]}))) + \Delta(\Pi((\sigma \circ g^{\mathcal{G}}_{\nu\delta;\nu,\nu})(U_{[\nu\delta]})), \Pi(\sigma(U_{[\nu]}, U_{[\nu]}))) \\ &\quad + \Delta(\Pi(\sigma(U_{[\nu]}, U_{[\nu]})), \Pi(U_{[n_1]}, U_{[n_2]})) \\ &\leq \Delta(\tau(U_{[N]}), U_{[\nu\delta]}) + \Delta(\Pi^\sigma(g^{\mathcal{G}}_{\nu\delta;\nu,\nu}(U_{[\nu\delta]})), \Pi^\sigma(U_{[\nu]}, U_{[\nu]})) + \Delta(\sigma(U_{[\nu]}, U_{[\nu]}), U_{[n_1]} \times U_{[n_2]}) \end{aligned} \tag{26}$$

where we used the triangle inequality (to deduce the first inequality) and the fact (to deduce the second inequality) that $\Delta(f(r_1), f(r_2)) \leq \Delta(r_1, r_2)$ for any random variables r_1, r_2 over the same set and any map f. Lemma 4 implies that $\Delta(\tau(U_{[N]}), U_{[\nu\delta]}) = \rho(N, \nu\delta)$ and

$$\begin{aligned} \Delta(\sigma(U_{[\nu]}, U_{[\nu]}), U_{[n_1]} \times U_{[n_2]}) &= \Delta(\sigma_1(U_{[\nu]}) \times \sigma_2(U_{[\nu]}), U_{[n_1]} \times U_{[n_2]}) \\ &\leq \Delta(\sigma_1(U_{[\nu]}), U_{[n_1]}) + \Delta(\sigma_2(U_{[\nu]}), U_{[n_2]}) = \rho(\nu, n_1) + \rho(\nu, n_2) \ . \end{aligned} \tag{27}$$

To evaluate the remaining value $\Delta(\Pi^\sigma(g^{\mathcal{G}}_{\nu\delta;\nu,\nu}(U_{[\nu\delta]})), \Pi^\sigma(U_{[\nu]}, U_{[\nu]}))$, put $c := c(\Pi|\mathcal{I}') = c(\Pi^\sigma)$. Take the subsets $S_{i,h} \subset [\nu]$ ($i \in \{1,2\}$, $h \in [kc]$) given by Lemma 1 applied to Π^σ. For each possible output y of Π^σ, define $I_y \subset [kc]$ in such a way that we have $\Pi^\sigma(x_1, x_2) = y$ if and only if $(x_1, x_2) \in S_{1,h} \times S_{2,h}$ for some $h \in I_y$. Note that $[kc]$ is the disjoint union of those I_y. Moreover, for each $(x_1, x_2) \in [\nu] \times [\nu]$, let $\iota(x_1, x_2)$

denote the index $h \in [kc]$ with $(x_1, x_2) \in S_{1,h} \times S_{2,h}$. Then

$$\begin{aligned}
&\Delta(\Pi^\sigma(g^{\mathcal{G}}_{\nu\delta;\nu,\nu}(U_{[\nu\delta]})), \Pi^\sigma(U_{[\nu]}, U_{[\nu]})) \\
&= \frac{1}{2}\sum_y |Pr[y \leftarrow \Pi^\sigma(g^{\mathcal{G}}_{\nu\delta;\nu,\nu}(U_{[\nu\delta]}))] - Pr[y \leftarrow \Pi^\sigma(U_{[\nu]}, U_{[\nu]})]| \\
&= \frac{1}{2}\sum_y |Pr[(x_1,x_2) \leftarrow g^{\mathcal{G}}_{\nu\delta;\nu,\nu}(U_{[\nu\delta]}) : \iota(x_1,x_2) \in I_y] - Pr[(x_1,x_2) \leftarrow U_{[\nu]} \times U_{[\nu]} : \iota(x_1,x_2) \in I_y]| \\
&\leq \frac{1}{2}\sum_y \sum_{h \in I_y} \Big(|Pr[(x_1,x_2) \leftarrow g^{\mathcal{G}}_{\nu\delta;\nu,\nu}(U_{[\nu\delta]}) : (x_1,x_2) \in S_{1,h} \times S_{2,h}] \\
&\qquad\qquad - Pr[(x_1,x_2) \leftarrow U_{[\nu]} \times U_{[\nu]} : (x_1,x_2) \in S_{1,h} \times S_{2,h}]| \Big) \ .
\end{aligned} \tag{28}$$

Now by the construction of $g^{\mathcal{G}}_{\nu\delta;\nu,\nu}$, the probability that $g^{\mathcal{G}}_{\nu\delta;\nu,\nu}(U_{[\nu\delta]})$ outputs an element in $S_{1,h} \times S_{2,h}$ is equal to $|E(S_{1,h}, S_{2,h})|/(\nu\delta)$ (see Lemma 3 for the notation), while the probability that $U_{[\nu]} \times U_{[\nu]}$ outputs an element in $S_{1,h} \times S_{2,h}$ is $|S_{1,h}| \cdot |S_{2,h}|/\nu^2$. Therefore the right-hand side of (28) is equal to

$$\begin{aligned}
\frac{1}{2}\sum_{h\in[kc]} \left| \frac{|E(S_{1,h},S_{2,h})|}{\nu\delta} - \frac{|S_{1,h}| \cdot |S_{2,h}|}{\nu^2} \right| &= \frac{1}{2\nu\delta} \sum_{h\in[kc]} \left| |E(S_{1,h},S_{2,h})| - \frac{\delta |S_{1,h}| \cdot |S_{2,h}|}{\nu} \right| \\
&\leq \frac{1}{2\nu\delta} \sum_{h\in[kc]} \lambda(\mathcal{G}) \sqrt{|S_{1,h}| \cdot |S_{2,h}|} \\
&\leq \frac{\lambda(\mathcal{G})}{2\nu\delta} \sqrt{kc \sum_{h\in[kc]} |S_{1,h}| \cdot |S_{2,h}|} = \frac{\lambda(\mathcal{G})}{2\nu\delta} \sqrt{kc \cdot \nu^2} = \frac{\lambda(\mathcal{G})}{2\delta}\sqrt{kc} \ ,
\end{aligned} \tag{29}$$

where the first inequality is implied by Lemma 3, the second inequality is implied by Lemma 5, and the second last equality follows from the fact that $[\nu] \times [\nu]$ is the disjoint union of $S_{1,h} \times S_{2,h}$ for $h \in [kc]$. Hence the proof of Lemma 2 is concluded. □

5.3 The overall construction

From now, we describe the way of composing the building-block PRGs introduced in Section 5.2 to construct an INW PRG. Let Π be a μ-party protocol ($\mu \geq 2$) with bounded communication capacity. For simplicity, we assume that Π is deterministic, which is in fact sufficient for our purpose (as an arbitrary Π is a probabilistic ensemble of deterministic ones). We suppose that the inputs for Π are chosen from a rectangular subset $\mathcal{I}' = \mathcal{I}'_1 \times \cdots \times \mathcal{I}'_\mu$ of $\mathcal{I}$. We identify each $\mathcal{I}'_i$ with $[n_i]$, where $n_i := |\mathcal{I}'_i|$.

To construct INW PRGs, first we introduce the following objects. Let T be a rooted binary tree with μ leaves, where i-th leaf is identified with i-th player P_i of Π. Let $\mathrm{dp}(v)$ denote the depth of a vertex v in T, i.e., the number of edges in the path from the root to v. Let $\mathrm{dp}(T) := \max_v \mathrm{dp}(v)$ be the depth of the tree T. We choose positive integers ν_0, δ_1, $\nu_1, \ldots, \delta_{\mathrm{dp}(T)}$, $\nu_{\mathrm{dp}(T)}$ in such a way that $\nu_{i-1} \geq \nu_i\delta_i$ for every $i \in [\mathrm{dp}(T)]$ and $\nu_{\mathrm{dp}(P_i)} \geq n_i$ for every $i \in [\mu]$. Moreover, for each $i \in [\mathrm{dp}(T)]$, let $\mathcal{G}_i$ be a δ_i-regular graph with ν_i vertices. In this setting, the INW PRG g^{INW} is defined as a map $[\nu_0] \to [n_1] \times \cdots \times [n_\mu]$ in the following manner:

Definition 4. To define the value $g^{\mathrm{INW}}(x)$ of the INW PRG g^{INW} for input $x \in [\nu_0]$, we determine an intermediate value x_v associated to each vertex v of T inductively, and then we set $g^{\mathrm{INW}}(x) := (x_{P_1}, x_{P_2}, \ldots, x_{P_\mu})$. Each intermediate value x_v is determined as follows. First, for the root v_0 of T, we simply set $x_{v_0} := x$. For the other values, suppose that an element $x_v \in [\nu_i]$ has been determined for a non-leaf vertex v of depth i. If the left (respectively, right) child v_L (respectively, v_R) of v in T is a leaf (say, P_h), then we put $m_L := n_h$ (respectively, $m_R := n_h$); otherwise we put $m_L := \nu_{i+1}$ (respectively, $m_R := \nu_{i+1}$). Then we set $(x_{v_L}, x_{v_R}) := g^{\mathcal{G}_{i+1}}_{\nu_i;m_L,m_R}(x_v) \in [m_L] \times [m_R]$.

Roughly speaking, it can be shown *without any hardness assumption* that the INW PRG g^{INW} is indistinguishable against distinguishers given by μ-party protocols with bounded communication capacity. The proof of indistinguishability is reduced to the indistinguishability of the building-blocks $g^{\mathcal{G}_{i+1}}_{\nu_i;m_L,m_R}$ against 2-party distinguishers. For the sake of the reduction, we explain a conversion method as follows, which was used in [6], of a μ-party protocol Π into a 2-party protocol $\overline{\Pi}$:

Definition 5. For a μ-party protocol Π, we choose a partition $\mathcal{P}(\Pi) = \mathcal{P}_1 \cup \mathcal{P}_2$ of $\mathcal{P}(\Pi)$ into two non-empty disjoint subsets. Then we convert Π into a 2-party protocol $\overline{\Pi} = \overline{\Pi}_{\mathcal{P}_1}$ by gathering all players in $\mathcal{P}_1$ and in $\mathcal{P}_2$ as the first player $P^{(1)}$ and the second player $P^{(2)}$ of $\overline{\Pi}$, respectively. Namely, the local inputs for $P^{(1)}$ and for $P^{(2)}$ in $\overline{\Pi}$ are collections of local inputs in Π for the players in $\mathcal{P}_1$ and in $\mathcal{P}_2$, respectively. In the communication phase, $P^{(1)}$ and $P^{(2)}$ simulate the roles of all players in $\mathcal{P}_1$ and in $\mathcal{P}_2$, respectively. More precisely, for each channel C of Π, we introduce a channel $\overline{C}$ of $\overline{\Pi}$ in such a way that the source of $\overline{C}$ is $P^{(1)}$ (respectively, $P^{(2)}$) when $\mathsf{source}(C) \in \mathcal{P}_1$ (respectively, $\mathsf{source}(C) \in \mathcal{P}_2$), and $P^{(1)}$ (respectively, $P^{(2)}$) is in the target of $\overline{C}$ when some player in $\mathcal{P}_1$ (respectively, $\mathcal{P}_2$) belongs to $\mathsf{target}(C)$. Now if a message is sent at a step in the communication phase of Π via a channel C, then the same message is sent at the corresponding step in the communication phase of $\overline{\Pi}$ via the corresponding channel $\overline{C}$. Finally, the local outputs for $P^{(1)}$ and for $P^{(2)}$ in $\overline{\Pi}$ are collections of local outputs in Π for the players in $\mathcal{P}_1$ and in $\mathcal{P}_2$, respectively.

We introduce some more notations. For each non-leaf vertex v of T, let $T_{v,\mathrm{left}}$ and $T_{v,\mathrm{right}}$ denote the subtrees of T with the roots being the left and the right children of v in T, respectively. Let $L_{v,\mathrm{left}}$ and $L_{v,\mathrm{right}}$ denote the sets of leaves of $T_{v,\mathrm{left}}$ and $T_{v,\mathrm{right}}$, respectively, and let $L_{v,\mathrm{out}}$ denote the sets of leaves of T not belonging to $L_{v,\mathrm{left}} \cup L_{v,\mathrm{right}}$. Now for each collection $x_{v,\mathrm{out}}$ of possible local inputs for the players of Π in $L_{v,\mathrm{out}}$, let $\mathcal{I}'_{x_{v,\mathrm{out}}}$ denote the subset of $\mathcal{I}'$ satisfying that each component of any element of $\mathcal{I}'_{x_{v,\mathrm{out}}}$ corresponding to a player in $L_{v,\mathrm{out}}$ coincides with the component of $x_{v_{\mathrm{out}}}$ corresponding to the same player. Note that $\mathcal{I}'_{x_{v,\mathrm{out}}}$ is rectangular as well as $\mathcal{I}'$. We define

$$c_{\mathrm{two}} := \max_{v,x_{v,\mathrm{out}}} \min\{c(\overline{\Pi}_{L_{v,\mathrm{left}}}|\mathcal{I}'_{x_{v,\mathrm{out}}}), c(\overline{\Pi}_{L_{v,\mathrm{right}}}|\mathcal{I}'_{x_{v,\mathrm{out}}})\} \ , \tag{30}$$

where the maximum is taken over all non-leaf vertices v of T and over all possible $x_{v,\mathrm{out}}$.

Now we have the following result, which is an improvement of Theorem 2 in [6] (by virtue of the above-mentioned improvement of Lemma 2 from the counterpart in [6]):

Theorem 2. *In the above setting, suppose that the number of possible outputs of* Π *with inputs chosen from* $\mathcal{I}'$ *is not larger than* $k \in \mathbb{Z}$. *Let* $V_h(T)$, $0 \leq h \leq \mathrm{dp}(T) - 1$, *denote the set of non-leaf vertices of* T *with depth* h. *Then we have*

$$\begin{aligned} &\Delta(\Pi(g^{\mathrm{INW}}(U_{[\nu_0]})), \Pi(U_{[n_1]}, \dots, U_{[n_\mu]})) \\ &\leq \sum_{h=0}^{\mathrm{dp}(T)-1} |V_h(T)| \left(\frac{\lambda(\mathcal{G}_{h+1})}{2\delta_{h+1}} \sqrt{k c_{\mathrm{two}}} + \rho(\nu_h, \nu_{h+1}\delta_{h+1}) \right) + \sum_{i=1}^{\mu} \rho(\nu_{\mathrm{dp}(P_i)}, n_i) \end{aligned} \tag{31}$$

(see Section 5.1 for notations).

Proof. We notice that the basic strategy of the proof is the same as the proof of Theorem 2 in [6]. In the proof, to each vertex v of T we associate a set I_v in such a way that we have $I_v = [\nu_h]$ if v is not a leaf of T and $\mathrm{dp}(v) = h$, while we have $I_v = [n_i]$ if $v = P_i$ (which is a leaf of T). Let $T_{\wedge v}$ denote the subtree of T with root v. We define $\mathcal{B}$ to be the family of sets consisting of vertices of T, in such a way that a set X of vertices of T belongs to $\mathcal{B}$ if and only if every leaf of T is contained in a unique subtree $T_{\wedge v}$ with $v \in X$. For example, the set $L(T)$ of all the leaves of T and the one-element set $\{v_0\}$ consisting of the root v_0 of T are members of $\mathcal{B}$. Now for each $X \in \mathcal{B}$, we define an auxiliary map $g^X \colon \prod_{v \in X} I_v \to [n_1] \times \cdots \times [n_\mu]$ in the following manner analogous to g^{INW}. Given $(x_v)_{v \in X} \in \prod_{v \in X} I_v$, we determine elements $x_u \in I_u$ for vertices u of subtrees $T_{\wedge v}$ ($v \in X$) inductively as follows:

Suppose that an element $x_u \in I_u$ has been determined for a non-leaf vertex u with depth i. Then we put $(x_{u_L}, x_{u_R}) := g^{\mathcal{G}_{i+1}}_{|I_u|;|I_{u_L}|,|I_{u_R}|}(x_u) \in I_{u_L} \times I_{u_R}$, where u_L and u_R are the left and the right children of u in T, respectively.

Then we set $g^X((x_v)_{v\in X}) := (x_{P_1}, \dots, x_{P_\mu})$.

By the above construction, we have $g^{\{v_0\}} = g^{\mathrm{INW}}$, while $g^{L(T)}$ is the identity map on $[n_1] \times \cdots \times [n_\mu]$. From now, to evaluate the left-hand side of (31), we choose a sequence $X_0 := L(T), X_1, \dots, X_s := \{v_0\}$ of elements of $\mathcal{B}$ in a certain manner and evaluate the statistical distance between $\Pi(g^{X_{h-1}}((U_{I_v})_{v\in X_{h-1}}))$ and $\Pi(g^{X_h}((U_{I_v})_{v\in X_h}))$ for every $h \in [s]$. Then these evaluation results will be gathered (by using the triangle inequality) to deduce the inequality (31).

We define the elements X_t of $\mathcal{B}$ inductively as follows: If X_t has been chosen and $X_t \neq \{v_0\}$, then take a vertex u_{t+1} of T whose left child $u_{t+1,L}$ and right child $u_{t+1,R}$ are both members of X_t, and define $X_{t+1} := (X_t \setminus \{u_{t+1,L}, u_{t+1,R}\}) \cup \{u_{t+1}\}$. Note that such a vertex u_{t+1} always exists (for example, the parent of an element of X_t with largest depth satisfies the condition for u_{t+1}) and the new set X_{t+1} is also a member of $\mathcal{B}$. Note also that the process halts within finitely many steps, as the sizes of X_t are strictly decreasing.

For each $t \in [s]$, put $u := u_t$ for notational simplicity, and let $g_u \colon I_{u_L} \to \prod_{P_i \in L_{u,\mathrm{left}}}[n_i]$, $g'_u \colon I_{u_R} \to \prod_{P_i \in L_{u,\mathrm{right}}}[n_i]$ and $g''_u \colon \prod_{v\in X_t\setminus\{u\}} I_v \to \prod_{P_i \in L_{u,\mathrm{out}}}[n_i]$ be maps defined in the same way as g^{X_t}. Then by the triangle inequality, the statistical distance between $\Pi(g^{X_{t-1}}((U_{I_v})_{v\in X_{t-1}}))$ and $\Pi(g^{X_t}((U_{I_v})_{v\in X_t}))$ is bounded by the (weighted) average of statistical distances between $\Pi(g_u(U_{I_{u_L}}), g'_u(U_{I_{u_R}}), \xi)$ and $\Pi((g_u \times g'_u)(\overline{g}(U_{I_u})), \xi)$, where we put $\xi := g''_u((x_v)_{v\in X_t\setminus\{u\}})$, $\mathcal{G} := \mathcal{G}_{\mathrm{dp}(u)+1}$ and $\overline{g} := g^{\mathcal{G}}_{|I_u|;|I_{u_L}|,|I_{u_R}|}$ for notational simplicity, over all collections $(x_v)_{v\in X_t\setminus\{u\}}$ of $x_v \in I_v$ for $v \in X_t \setminus \{u\}$. Now we define an auxiliary 2-party protocol Π' (respectively, Π'') in such a way that, given inputs $x_{u_L} \in I_{u_L}$ and $x_{u_R} \in I_{u_R}$ for the two players, they perform the protocol $\overline{\Pi}_{L_{u,\mathrm{left}}}$ with inputs $g_u(x_{u_L})$ and $(g'_u(x_{u_R}), \xi)$ (respectively, the protocol $\overline{\Pi}_{L_{u,\mathrm{right}}}$ with inputs $g'_u(x_{u_R})$ and $(g_u(x_{u_L}), \xi)$). Then we have $c(\Pi') = c(\overline{\Pi}_{L_{v,\mathrm{left}}}|\mathcal{I}'_\xi)$ and $c(\Pi'') = c(\overline{\Pi}_{L_{v,\mathrm{right}}}|\mathcal{I}'_\xi)$ by the definitions. We suppose for simplicity that $c(\Pi') \leq c(\Pi'')$ (the other case is similar). Then we have $c(\Pi') \leq c_{\mathrm{two}}$ by the definition of c_{two}. Now we have $\Pi(g_u(U_{I_{u_L}}), g'_u(U_{I_{u_R}}), \xi) = \Pi'(U_{I_{u_L}}, U_{I_{u_R}})$ and $\Pi((g_u \times g'_u)(\overline{g}(U_{I_u})), \xi) = \Pi'(\overline{g}(U_{I_u}))$, therefore it suffices to evaluate the statistical distance between $\Pi'(U_{I_{u_L}}, U_{I_{u_R}})$ and $\Pi'(\overline{g}(U_{I_u}))$. By Lemma 2, the statistical distance is not larger than

$$\frac{\lambda(\mathcal{G}_{\mathrm{dp}(u)+1})}{2\delta_{\mathrm{dp}(u)+1}}\sqrt{kc_{\mathrm{two}}} + \rho(|I_u|, \nu_{\mathrm{dp}(u)+1}\delta_{\mathrm{dp}(u)+1}) + \rho(\nu_{\mathrm{dp}(u)+1}, |I_{u_L}|) + \rho(\nu_{\mathrm{dp}(u)+1}, |I_{u_R}|) \tag{32}$$

regardless of the choice of ξ. This implies that the statistical distance between $\Pi(g^{X_{h-1}}((U_{I_v})_{v\in X_{h-1}}))$ and $\Pi(g^{X_h}((U_{I_v})_{v\in X_h}))$ is also bounded by (32). Note that $\mathrm{dp}(u) \leq \mathrm{dp}(T)-1$ in this case, therefore $|I_u| = \nu_{\mathrm{dp}(u)}$.

In summing up the values in (32) over all the specified vertices $u = u_t$ for $t \in [s]$,

- for the first two terms $\frac{\lambda(\mathcal{G}_{\mathrm{dp}(u)+1})}{2\delta_{\mathrm{dp}(u)+1}}\sqrt{kc_{\mathrm{two}}} + \rho(\nu_{\mathrm{dp}(u)}, \nu_{\mathrm{dp}(u)+1}\delta_{\mathrm{dp}(u)+1})$, for each $0 \leq h \leq \mathrm{dp}(T)-1$, the case $\mathrm{dp}(u) = h$ appears precisely $|V_h(T)|$ times;
- for the remaining two terms $\rho(\nu_{\mathrm{dp}(u)+1}, |I_{u_L}|) + \rho(\nu_{\mathrm{dp}(u)+1}, |I_{u_R}|)$, it sums up $\rho(\nu_{\mathrm{dp}(v)}, |I_v|)$ for all vertices v of T other than the root of T, and we have $|I_v| = n_i$ if v is a leaf P_i of T, while we have $\rho(\nu_{\mathrm{dp}(v)}, |I_v|) = \rho(\nu_{\mathrm{dp}(v)}, \nu_{\mathrm{dp}(v)}) = 0$ if v is not a leaf of T.

Hence the resulting sum is equal to the right-hand side of (31), concluding the proof of Theorem 2. □

6 On pseudorandomization using INW PRGs

Based on the results in the previous sections, in this section we give quantitative evaluations of advantages of distinguishers for INW PRGs based on the above-mentioned bucket-brigade model. More precisely, as the bound for the statistical distance in Theorem 2 is a function of the (conditional) communication capacity of the multi-party protocol, the main task in this section is to estimate, in terms of the memory size $|M|$

for a distinguisher, the (conditional) communication capacity of the multi-party protocol $\Pi(\mathsf{A})$ associated to the distinguisher A under the bucket-brigade model. The communication capacity, hence the resulting advantage, becomes smaller as the memory size is decreasing. A numerical example for some existing information-theoretically secure scheme will be given in Section 7.

6.1 Advantages of distinguishers in bucket-brigade model

In principle, the (conditional) communication capacity appeared in Theorem 2 can be estimated for the multi-party protocols $\Pi(\mathsf{A})$ associated to algorithms A described in bucket-brigade model with arbitrary configuration graphs. However, the analysis of the case of arbitrary configuration graphs is too complicated, therefore here we focus on a special case that the random tape is a one-way tape of circular shape. In terms of configuration graphs, this means that the configuration graph $\mathcal{G}(\mathsf{A})$ for the random tape is the directed cycle $1 \to 2 \to \cdots \to \mu \to 1$, where $\mu \geq 2$ is the number of cells in the random tape (hence $\Pi(\mathsf{A})$ is a μ-party protocol).

To evaluate the communication capacity of $\Pi = \Pi(\mathsf{A})$, we introduce the following quantity: Let N_{read} denote the maximum, taken over all $i \in [\mu]$, of the number of steps during the communication phase of Π in which player P_i sends some message via a channel of the form $C_{i,i'}$ with $i' \in [\mu] \cup \{\perp\}$. Intuitively, this means that in an execution of the algorithm A, the content of each cell of the random tape is read at most N_{read} times. Now we have the following result:

Lemma 6. *In the above setting, suppose further that the initial memory state* $\mathsf{mem}_0 \in M$ *for* $\Pi = \Pi(\mathsf{A})$ *is a constant value; consequently, the joint input space* $\mathcal{I}$ *for* Π *is identified with* $R_1 \times \cdots \times R_\mu$*. Then the quantity* c_{two} *defined by* (30) *in the case* $\mathcal{I}' = \mathcal{I}$ *associated to* $\Pi = \Pi(\mathsf{A})$ *is bounded by* $c_{\mathrm{two}} \leq C(\mu, |M|)$*, where*

$$C(\mu, |M|) := ((\mu - 1)|M|^2 + 1)\frac{|M|^{2N_{\mathrm{read}}} - 1}{|M|^2 - 1} \quad . \tag{33}$$

Proof. First, by the construction of the tree T and the definitions of the subsets $L_{v,\mathrm{left}}$ and $L_{v,\mathrm{right}}$ of leaves of T, it suffices to show that $c(\overline{\Pi}_{\mathcal{P}_1}) \leq C(\mu, |M|)$ for any partition $\mathcal{P}(\Pi) = \mathcal{P}_1 \cup \mathcal{P}_2$ of $\mathcal{P}(\Pi) = \{P_1, \ldots, P_\mu\}$ satisfying that $\mathcal{P}_1 = \{P_h, P_{h+1}, \ldots, P_{h'-1}, P_{h'}\}$ for some $1 \leq h \leq h' \leq \mu - 1$ (note that we have supposed $P_\mu \notin \mathcal{P}_1$).

To count the communication patterns of $\overline{\Pi}_{\mathcal{P}_1}$, first we consider the case that $h \geq 2$, hence $P_1, P_\mu \notin \mathcal{P}_1$. Suppose that a communication phase of Π halts at $(s\mu+t+1)$-th step, where $1 \leq t \leq \mu$ and $0 \leq s \leq N_{\mathrm{read}}-1$, therefore the last (non-empty) message in the communication phase is sent via a channel $C_{t,\perp}$ (recall that the configuration graph $\mathcal{G}(\mathsf{A})$ is a cycle). Now we divide the argument into the following subcases:

Case 1-1: $h \geq 2$**,** $1 \leq t \leq h - 1$**.** By the shape of the configuration graph $\mathcal{G}(\mathsf{A})$, the channel for Π used at each, say, j-th step is uniquely determined by j, therefore the communication patterns for Π is expressed by a sequence $m_1, m_2, \ldots, m_{s\mu+t}$ of messages $m_j \in M$ sent at j-th step. Moreover, in the definition of $c(\overline{\Pi}_{\mathcal{P}_1})$, only the messages sent in Π from a player in $\mathcal{P}_1$ to a player in $\mathcal{P}_2$ and the messages sent from a player in $\mathcal{P}_2$ to a player in $\mathcal{P}_1$ are concerned and the other messages are ignored. In the present case, the former ones are $m_{h'}, m_{\mu+h'}, \ldots, m_{(s-1)\mu+h'}$ (all sent from $P_{h'}$ to $P_{h'+1}$ via the channel $C_{h',h'+1}$), s messages in total, and the latter ones are $m_{h-1}, m_{\mu+h-1}, \ldots, m_{(s-1)\mu+h-1}$ (all sent from P_{h-1} to P_h via the channel $C_{h-1,h}$), s messages in total. Therefore, the number of communication patterns counted by $c(\overline{\Pi}_{\mathcal{P}_1})$ in this case is not larger than $|M|^{2s}$.

Case 1-2: $h \geq 2$**,** $h \leq t \leq h'$**.** We apply an argument similar to Case 1-1. Now, among the messages $m_1, m_2, \ldots, m_{s\mu+t}$ sent during the communication phase of Π, the messages sent from a player in $\mathcal{P}_1$ to a player in $\mathcal{P}_2$ are $m_{h'}, m_{\mu+h'}, \ldots, m_{(s-1)\mu+h'}$ (from $P_{h'}$ to $P_{h'+1}$ via $C_{h',h'+1}$) and $m_{s\mu+t}$ (sent from P_t to P_μ via the channel $C_{t,\perp}$), $s+1$ messages in total. On the other hand, the messages sent from a player in $\mathcal{P}_2$ to a player in $\mathcal{P}_1$ are $m_{h-1}, m_{\mu+h-1}, \ldots, m_{s\mu+h-1}$ (from P_{h-1} to P_h via $C_{h-1,h}$), $s+1$ messages in total. Therefore, the number of communication patterns counted by $c(\overline{\Pi}_{\mathcal{P}_1})$ in this case is not larger than $|M|^{2s+2}$.

Case 1-3: $h \geq 2$, $h' + 1 \leq t \leq \mu$. We apply an argument similar to Case 1-1 and Case 1-2. Now, among the messages $m_1, m_2, \ldots, m_{s\mu+t}$ sent during the communication phase of Π, the messages sent from a player in $\mathcal{P}_1$ to a player in $\mathcal{P}_2$ are $m_{h'}, m_{\mu+h'}, \ldots, m_{s\mu+h'}$ (from $P_{h'}$ to $P_{h'+1}$ via $C_{h',h'+1}$), $s+1$ messages in total. On the other hand, the messages sent from a player in $\mathcal{P}_2$ to a player in $\mathcal{P}_1$ are $m_{h-1}, m_{\mu+h-1}, \ldots, m_{s\mu+h-1}$ (from P_{h-1} to P_h via $C_{h-1,h}$), $s+1$ messages in total. Therefore, the number of communication patterns counted by $c(\overline{\Pi}_{\mathcal{P}_1})$ in this case is not larger than $|M|^{2s+2}$.

By summing up the above results over all $0 \leq s \leq N_{\mathrm{read}} - 1$ and $1 \leq t \leq \mu$, we have

$$c(\overline{\Pi}_{\mathcal{P}_1}) \leq \sum_{s=0}^{N_{\mathrm{read}}-1} ((h-1)|M|^{2s} + (\mu - h + 1)|M|^{2s+2}) = (h - 1 + (\mu - h + 1)|M|^2)\frac{|M|^{2N_{\mathrm{read}}} - 1}{|M|^2 - 1} \tag{34}$$

(recall that we have assumed $|M| \geq 2$). Under the current condition $h \geq 2$, the coefficient $h - 1 + (\mu - h + 1)|M|^2$ takes the maximal value at $h = 2$, which is $(\mu - 1)|M|^2 + 1$. Hence the desired relation $c(\overline{\Pi}_{\mathcal{P}_1}) \leq C(\mu, |M|)$ holds in the case $h \geq 2$.

Secondly, we consider the case that $h = 1$, hence $P_1 \in \mathcal{P}_1$ and $P_\mu \notin \mathcal{P}_1$. In the same way as above, we suppose that a communication phase of Π halts at $(s\mu+t+1)$-th step, where $1 \leq t \leq \mu$ and $0 \leq s \leq N_{\mathrm{read}}-1$. We divide the argument into the following subcases:

Case 2-1: $h = 1$, $1 \leq t \leq h'$. Now, among the messages $m_1, m_2, \ldots, m_{s\mu+t}$ sent during the communication phase of Π, the messages sent from a player in $\mathcal{P}_1$ to a player in $\mathcal{P}_2$ are $m_{h'}, m_{\mu+h'}, \ldots, m_{(s-1)\mu+h'}$ (from $P_{h'}$ to $P_{h'+1}$ via $C_{h',h'+1}$) and $m_{s\mu+t}$ (sent from P_t to P_μ via $C_{t,\perp}$), $s+1$ messages in total. On the other hand, the messages sent from a player in $\mathcal{P}_2$ to a player in $\mathcal{P}_1$ are $m_\mu, m_{2\mu}, \ldots, m_{s\mu}$ (from P_μ to P_1 via $C_{\mu,1}$), s messages in total. Therefore, the number of communication patterns counted by $c(\overline{\Pi}_{\mathcal{P}_1})$ in this case is not larger than $|M|^{2s+1}$.

Case 2-2: $h = 1$, $h' + 1 \leq t \leq \mu$. Now, among the messages $m_1, m_2, \ldots, m_{s\mu+t}$ sent during the communication phase of Π, the messages sent from a player in $\mathcal{P}_1$ to a player in $\mathcal{P}_2$ are $m_{h'}, m_{\mu+h'}, \ldots, m_{s\mu+h'}$ (from $P_{h'}$ to $P_{h'+1}$ via $C_{h',h'+1}$), $s+1$ messages in total. On the other hand, the messages sent from a player in $\mathcal{P}_2$ to a player in $\mathcal{P}_1$ are $m_\mu, m_{2\mu}, \ldots, m_{s\mu}$ (from P_μ to P_1 via $C_{\mu,1}$), s messages in total. Therefore, the number of communication patterns counted by $c(\overline{\Pi}_{\mathcal{P}_1})$ in this case is not larger than $|M|^{2s+1}$.

By summing up the above results over all $0 \leq s \leq N_{\mathrm{read}} - 1$ and $1 \leq t \leq \mu$, we have

$$c(\overline{\Pi}_{\mathcal{P}_1}) \leq \sum_{s=0}^{N_{\mathrm{read}}-1} \mu|M|^{2s+1} = \mu|M|\frac{|M|^{2N_{\mathrm{read}}} - 1}{|M|^2 - 1} \ . \tag{35}$$

Now the conditions $\mu \geq 2$ and $|M| \geq 2$ imply that $\mu|M| \leq (\mu - 1)|M|^2 + 1$, therefore the inequality $c(\overline{\Pi}_{\mathcal{P}_1}) \leq C(\mu, |M|)$ also holds in this case. Hence the proof of Lemma 6 is concluded. □

From now, we construct INW PRG $g^{\mathrm{INW}} \colon [\nu_0] \to [n_1] \times \cdots \times [n_\mu]$ that pseudorandomizes the random input tape for $\Pi(\mathsf{A})$, where for each $i \in [\mu]$, we put $n_i := |R_i|$ and identify R_i with $[n_i]$ (see Section 4.3 for the notations). We take a rooted binary tree T in such a way that its depth $\mathrm{dp}(T)$ attains the minimal possible value $\lceil \log_2 \mu \rceil$ and the leaves $P_1, \ldots, P_\mu$ of T (identified with the μ players of $\Pi(\mathsf{A})$) are arranged in this order from left to right. This implies that, for any vertex v of T, the set of leaves of the subtree $T_{\wedge v}$ of T with root v are of the form $\{P_h, P_{h+1}, \ldots, P_{h'-1}, P_{h'}\}$ with $1 \leq h \leq h' \leq \mu$. Then we define g^{INW} as in Section 5.3 according to the tree T. We put $\Pi := \Pi(\mathsf{A})$ and $\mathsf{A}_{\mathrm{Rnd}} := \Pi(U_{[n_1]}, \ldots, U_{[n_\mu]})$ (where, for simplicity, the part $\mathsf{mem}_0 \in M$ of the input for Π is made implicit), which means the output distribution of the algorithm A with contents of the random tape given uniformly at random. On the other hand, we put $\mathsf{A}_{\mathrm{PRnd}} := \Pi(g^{\mathrm{INW}}(U_{[\nu_0]}))$, which means the output distribution of A with the contents of the random tape given by the PRG g^{INW}. Moreover, let N_{out} denote the number of possible outputs of Π (or equivalently,

of $\mathcal{A}$). Then the statistical distance $\Delta(\mathsf{A}_{\mathrm{Rnd}}, \mathsf{A}_{\mathrm{PRnd}})$, which is the advantage of the distinguisher, can be evaluated by using Theorem 2 and the above result:

Theorem 3. *In the above setting,* $\Delta(\mathsf{A}_{\mathrm{Rnd}}, \mathsf{A}_{\mathrm{PRnd}})$ *is not larger than*

$$\begin{aligned}&\sum_{h=0}^{\mathrm{dp}(T)-2} 2^h \left(\frac{\lambda(\mathcal{G}_{h+1})}{2\delta_{h+1}} \sqrt{N_{\mathrm{out}} C(\mu, |M|)} + \rho(\nu_h, \nu_{h+1}\delta_{h+1}) \right) \\ &+ (\mu - 2^{\mathrm{dp}(T)-1}) \left(\frac{\lambda(\mathcal{G}_{\mathrm{dp}(T)})}{2\delta_{\mathrm{dp}(T)}} \sqrt{N_{\mathrm{out}} C(\mu, |M|)} + \rho(\nu_{\mathrm{dp}(T)-1}, \nu_{\mathrm{dp}(T)}\delta_{\mathrm{dp}(T)}) \right) + \sum_{i=1}^{\mu} \rho(\nu_{\mathrm{dp}(v_i)}, n_i)\end{aligned} \tag{36}$$

(see (33) *for the definition of* $C(\mu, |M|)$*, and see Section 5.3 for notations). Hence, if for every* h *we have* $\lambda(\mathcal{G}_h)/(2\delta_h) \leq \varepsilon_g$ *for a common* $\varepsilon_g \in \mathbb{R}$*, and for every* h *and* i *we have* $\rho(\nu_h, \nu_{h+1}\delta_{h+1}) \leq \varepsilon_\rho$ *and* $\rho(\nu_{\mathrm{dp}(v_i)}, n_i) \leq \varepsilon_\rho$ *for a common* $\varepsilon_\rho \in \mathbb{R}$*, then we have*

$$\Delta(\mathsf{A}_{\mathrm{Rnd}}, \mathsf{A}_{\mathrm{PRnd}}) \leq (\mu - 1)\varepsilon_g \sqrt{N_{\mathrm{out}} C(\mu, |M|)} + (2\mu - 1)\varepsilon_\rho \ . \tag{37}$$

Proof. The latter inequality (37) follows from the former part of the statement and direct calculation, therefore we prove the former part from now. As the choice of $\mathsf{mem}_0 \in M$ in the input phase of Π is independent of the other parts $r_1, \ldots, r_\mu$ of the input, we may assume without loss of generality (due to the triangle inequality) that mem_0 is a constant value. Then Lemma 6 can be applied, and it follows that $c_{\mathrm{two}} \leq C(\mu, |M|)$ where we put $\mathcal{I}' = \mathcal{I}$ in the definition of c_{two}. Now we use Theorem 2 with $\mathcal{I}' = \mathcal{I}$ (note that $k = N_{\mathrm{out}}$). By the construction of the binary tree T, we have $|V_h(T)| = 2^h$ for $0 \leq h \leq \mathrm{dp}(T) - 2$ and $|V_{\mathrm{dp}(T)-1}| = \mu - 2^{\mathrm{dp}(T)-1}$. Therefore, the bound in the statement is derived by using the inequality $c_{\mathrm{two}} \leq C(\mu, |M|)$ and by substituting these values of $|V_h(T)|$ into Theorem 2. □

6.2 Asymptotic evaluation of sufficient seed lengths

Based on the arguments in Section 2.2 and Section 6.1, in this subsection we give an asymptotic evaluation of a sufficient seed length of the INW PRG g^{INW} for pseudorandomization of an information-theoretically secure protocol Π. By Theorem 3, if a class Class of distinguishers D is defined by the constraints on the quantities μ, $|M|$ and N_{read} associated to D (where μ is equal to the length of the original random tape for the protocol Π), then g^{INW} is $(\mathsf{Class}, \varepsilon_{\mathsf{Class}})$-secure with $\varepsilon_{\mathsf{Class}}$ given by (36) (note that $N_{\mathrm{out}} = 2$ in this case, as each distinguisher outputs either 0 or 1). Therefore, if we want to bound the difference of attack success probabilities in random and pseudorandom cases by ε, then Corollary 1 implies that it suffices to show that $n \cdot \varepsilon_{\mathsf{Class}} \leq \varepsilon$, or equivalently $\varepsilon_{\mathsf{Class}} \leq \varepsilon/n$, where n is the number of components of the partition of the set X introduced in Section 2.2.

Here we introduce some assumptions for simplicity. First, we assume that we have chosen the regular graphs $\mathcal{G}_1, \ldots, \mathcal{G}_{\mathrm{dp}(T)}$ used in the construction of g^{INW}, where $\mathrm{dp}(T) = \lceil \log_2 \mu \rceil$, in such a way that the degrees $\delta_1, \ldots, \delta_{\mathrm{dp}(T)}$ are a common value δ and $\nu_h = \nu_{h+1}\delta$ for every $1 \leq h \leq \mathrm{dp}(T) - 1$. Secondly, we assume that the second largest eigenvalues $\lambda(\mathcal{G}_h)$ of $\mathcal{G}_h$ satisfy that $\lambda(\mathcal{G}_h) \leq 2\sqrt{\delta - 1}$ for every $1 \leq h \leq \mathrm{dp}(T)$, i.e., all $\mathcal{G}_h$ are Ramanujan graphs (see e.g., [8]). On the other hand, we put $R := \max_{1 \leq i \leq \mu} |R_i|$ (e.g., $R = 2$ in the case of a binary random tape). Then we have the following result:

Theorem 4. *In the above setting, if the parameters are chosen in such a way that*

$$\delta = \lceil 8n^2\varepsilon^{-2}\mu^2 C(\mu, |M|) \rceil \, , \ \nu_{\mathrm{dp}(T)} = \lceil n\varepsilon^{-1}\mu R \rceil \, , \ \nu_0 = 2^{\nu_0^\dagger} \, , \ \nu_0^\dagger = \lceil \log_2(n^2\varepsilon^{-2}\mu^2\delta^{\mathrm{dp}(T)}R) \rceil \ , \tag{38}$$

where $\mathrm{dp}(T) = \lceil \log_2 \mu \rceil$ *and* $C(\mu, |M|)$ *is as defined in* (33)*, and every algorithm* $\mathsf{A}_{i,f}$ *specified in Corollary 1 belongs to the class* Class*, then the difference of attack success probabilities in random and pseudorandom cases is bounded by* ε*. Moreover, when the length* μ *of the original random tape for the protocol* Π *and the memory size* $|M|$ *for the algorithms* $\mathsf{A}_{i,f}$ *tend to* ∞ *while the alphabet size* R *for the random tape is constant, the asymptotic seed length* $\nu_0^\dagger$ *of the INW PRG* g^{INW} *is given by*

$$\nu_0^\dagger \sim 3(\log_2 \mu)^2 + 2N_{\mathrm{read}} \log_2 |M| \log_2 \mu + 2\log_2(n/\varepsilon) \log_2 \mu \quad (\mu, |M| \to \infty). \tag{39}$$

Proof. For the first part, it suffices to show that the value $\varepsilon_{\mathsf{Class}}$ in (36) is not larger than ε/n under the current choice of parameters. First, the assumption that $\delta_{h+1} = \delta$ and $\nu_h = \nu_{h+1}\delta$ for every $1 \le h \le \mathrm{dp}(T) - 1$ implies that $\rho(\nu_h, \nu_{h+1}\delta_{h+1}) = 0$ for every $1 \le h \le \mathrm{dp}(T) - 1$. On the other hand, we notice the following inequality for the function ρ;

$$\rho(n', m') = \frac{(n' \mod m') \cdot (m' - (n' \mod m'))}{n'm'} \le \frac{(m'/2)^2}{n'm'} = \frac{m'}{4n'} \tag{40}$$

(note that $0 \le (n' \bmod m') \le m'$). Then by the assumption that $R \ge |R_i| = n_i$ for every $i \in [\mu]$ and the condition that $\nu_h \ge \nu_{h+1}$ for every $0 \le h \le \mathrm{dp}(T) - 1$, for every $i \in [\mu]$, we have

$$\rho(\nu_{\mathrm{dp}(v_i)}, n_i) \le \frac{n_i}{4\nu_{\mathrm{dp}(v_i)}} \le \frac{R}{4\nu_{\mathrm{dp}(T)}} \le \frac{R}{4n\varepsilon^{-1}\mu R} = \frac{\varepsilon}{4n\mu} \tag{41}$$

(where we used (38) to deduce the third inequality). The relation $\nu_1 = \nu_{\mathrm{dp}(T)}\delta^{\mathrm{dp}(T)-1}$ implies also that

$$\begin{aligned} \rho(\nu_0, \nu_1\delta_1) \le \frac{\nu_1\delta_1}{4\nu_0} = \frac{\nu_{\mathrm{dp}(T)}\delta^{\mathrm{dp}(T)}}{4\nu_0} &\le \frac{(n\varepsilon^{-1}\mu R + 1)\delta^{\mathrm{dp}(T)}}{4n^2\varepsilon^{-2}\mu^2\delta^{\mathrm{dp}(T)}R} \\ &\le \frac{2n\varepsilon^{-1}\mu R\delta^{\mathrm{dp}(T)}}{4n^2\varepsilon^{-2}\mu^2\delta^{\mathrm{dp}(T)}R} = \frac{\varepsilon}{2n\mu} \le \frac{\varepsilon}{4n} \end{aligned} \tag{42}$$

(where we used (38) to deduce the second inequality; recall that we have assumed $\mu \ge 2$). Moreover, we have $\lambda(\mathcal{G}_h) \le 2\sqrt{\delta - 1}$ for every $1 \le h \le \mathrm{dp}(T)$ by the assumption. By substituting these relations to (36) and by using the value N_{out} mentioned above, we have

$$\begin{aligned} \varepsilon_{\mathsf{Class}} &\le \frac{\sqrt{\delta-1}}{\delta}\sqrt{2C(\mu,|M|)} + \frac{\varepsilon}{4n} + \sum_{h=1}^{\mathrm{dp}(T)-2} 2^h \frac{\sqrt{\delta-1}}{\delta}\sqrt{2C(\mu,|M|)} \\ &\quad + (\mu - 2^{\mathrm{dp}(T)-1})\frac{\sqrt{\delta-1}}{\delta}\sqrt{2C(\mu,|M|)} + \sum_{i=1}^{\mu}\frac{\varepsilon}{4n\mu} \\ &= (\mu-1)\frac{\sqrt{\delta-1}}{\delta}\sqrt{2C(\mu,|M|)} + \frac{\varepsilon}{2n} \le \mu\delta^{-1/2}\sqrt{2C(\mu,|M|)} + \frac{\varepsilon}{2n} \ . \end{aligned} \tag{43}$$

By using (38) further, it follows that the first term of the right-hand side is not larger than $\varepsilon/(2n)$, therefore we have $\varepsilon_{\mathsf{Class}} \le \varepsilon/n$, as desired. Hence the first part of the claim holds.

For the second part of the claim, first note that $C(\mu, |M|) \sim \mu|M|^{2N_{\mathrm{read}}}$ in the limit case $\mu, |M| \to \infty$. Therefore we have $\delta \sim 8n^2\varepsilon^{-2}\mu^3|M|^{2N_{\mathrm{read}}}$, and consequently

$$\log_2\delta \sim 3 + 2\log_2(n/\varepsilon) + 3\log_2\mu + 2N_{\mathrm{read}}\log_2|M| \sim 2\log_2(n/\varepsilon) + 3\log_2\mu + 2N_{\mathrm{read}}\log_2|M| \ . \tag{44}$$

This and the property $\mathrm{dp}(T) \sim \log_2\mu$ imply that

$$\begin{aligned} \nu_0^\dagger &\sim 2\log_2(n/\varepsilon) + 2\log_2\mu + \mathrm{dp}(T)\log_2\delta + \log_2 R \\ &\sim 2\log_2(n/\varepsilon) + 2\log_2\mu + 2\log_2\mu\log_2(n/\varepsilon) + 3(\log_2\mu)^2 + 2N_{\mathrm{read}}\log_2\mu\log_2|M| + \log_2 R \\ &\sim 2\log_2\mu\log_2(n/\varepsilon) + 3(\log_2\mu)^2 + 2N_{\mathrm{read}}\log_2\mu\log_2|M| \end{aligned} \tag{45}$$

(note that $2\log_2(n/\varepsilon) + 2\log_2\mu\log_2(n/\varepsilon) \sim 2\log_2\mu\log_2(n/\varepsilon)$ and $2\log_2\mu + 3(\log_2\mu)^2 + \log_2 R \sim 3(\log_2\mu)^2$). Hence the proof of Theorem 4 is concluded. □

7 Example for existing schemes

In this section, we apply the above general result to the case of specific existing schemes in order to estimate the performance of our pseudorandomization technique further and to give comparison to the preceding result in [11, 12].

7.1 Summary of the comparison

We consider an existing information-theoretically secure scheme and the associated security game described in Section 7.2 below. The choice of the security game as well as the relevant parameters are the same as the ones used in the numerical example of the preceding result [11]. We deal with the seven choices of parameters, and Table 1 shows the estimated seed lengths of INW PRGs g^{INW} based on our result, together with the original bit lengths μ of the random input and the seed lengths calculated in the example in [11] (by using the PRGs against time-bounded distinguishers proposed by Farashahi, Schoenmakers and Sidorenko [4]). The last two rows of the table show the ratios of our seed lengths compared to the original bit lengths μ and the results in [11].

Table 1: Comparison of estimated seed lengths with the preceding result [11] (where μ denotes the original bit lengths of the random input)

N	10^3	10^4	10^5	10^6	10^7	10^8	10^9
m	614	702	789	877	964	1052	1139
original μ	9.21E6	1.05E8	1.18E9	1.31E10	1.44E11	1.57E12	1.70E13
length in [11]	6.87E6	9.72E6	1.33E7	1.75E7	2.25E7	2.83E7	3.51E7
our length $\nu_0^\dagger$	1.28E5	1.63E5	2.09E5	2.53E5	3.10E5	3.63E5	4.21E5
ratio to μ	1.39E-2	1.55E-3	1.77E-4	1.93E-5	2.15E-6	2.31E-7	2.48E-8
ratio to [11]	1.86E-2	1.68E-2	1.57E-2	1.45E-2	1.38E-2	1.28E-2	1.20E-2

We give two remarks on the comparison. First, as explained in the following subsections, our estimate of seed lengths is based on Theorem 4, where we assumed ideal choices of regular graphs $\mathcal{G}_1, \ldots, \mathcal{G}_{\mathrm{dp}(T)}$ to construct the INW PRGs g^{INW}, which have not been practically assured so far. Due to the assumption, a very explicit comparison of the values in Table 1 would not make sense. However, we still expect from the table that the required seed lengths based on our result would be at least competitive to the ones in [11], despite the removal of hardness assumptions in contrast to the result in [11] that was based on DDH assumptions. Secondly, in our numerical evaluation of seed lengths, only one choice of the partition $X = \bigcup_{i=1}^{n} X_i$ of the set X of the adversary's inputs among the various possible partitions, which is the same as the one appeared in [11, 12], is used due to the technical difficulty. Therefore, there is a room to improve the required seed lengths in our result further by investigating better choices of the partition of the set X.

In the following subsections, we describe the details of the numerical evaluation of our seed lengths.

7.2 Security game for an information-theoretically secure scheme

The information-theoretically secure scheme studied in this section as an example is a fingerprint code proposed by Nuida et al. [10], which is the same as the one used in [11]. The parameters, e.g., the number $c = 3$ of the adversaries, are also chosen in the same way as [11]. Here we omit some details which are less important in the present example; see [11] for those omitted details.

In the present case of the fingerprint code, the security game is described as follows, where N denotes the number of users (i.e., codewords in the code), m denotes the code length, and $1 \leq i_1 < i_2 < i_3 \leq N$ are arbitrarily chosen three indices:

1. For each $j \in [m]$, a bit $\pi_j \in \{0, 1\}$ is chosen independently and uniformly at random.
2. For each $i \in [N]$ and $j \in [m]$, a bit $w_{i,j} \in \{0, 1\}$ is chosen independently in such a way that $Pr[w_{i,j} = 1] = p^{(\pi_j)}$, where $0 < p^{(0)} < 1$ and $p^{(1)} = 1 - p^{(0)}$ are certain parameters with 15-bit accuracy (i.e., the fractional part of the binary expression of $p^{(0)}$ consists of 15 bits).
3. The adversary receives three codewords $w_{i_t} := (w_{i_t,j})_{j \in [m]}$, $1 \leq t \leq 3$.

4. The adversary outputs a word $y \in \{0,1,?\}^m$ (where '?' is an extra symbol), called a *pirated word*. (In fact, the word y is required to satisfy a constraint called Marking Assumption [2]. Our argument here does not depend on the constraint.)

5. For each $i \in [N]$, a *score* $\mathsf{sc}_i = \sum_{j=1}^m \mathsf{sc}_{i,j}$ is calculated by

$$\mathsf{sc}_{i,j} := \begin{cases} u_{\pi_j} & \text{if } y_j = 1 \text{ and } w_{i,j} = 1 \ , \\ -u_{1-\pi_j} & \text{if } y_j = 1 \text{ and } w_{i,j} = 0 \ , \\ -u_{\pi_j} & \text{if } y_j \neq 1 \text{ and } w_{i,j} = 1 \ , \\ u_{1-\pi_j} & \text{if } y_j \neq 1 \text{ and } w_{i,j} = 0 \ , \end{cases} \tag{46}$$

   where

$$\begin{aligned} u_0 &:= 1.931793212890625 = (1.111011101000101)_2 \ , \\ u_1 &:= 0.5176544189453125 = (0.1000010010000101)_2 \ , \end{aligned} \tag{47}$$

   each having 15-bit accuracy.

6. Let $i_* \in [N]$ be the index for which sc_{i_*} is the largest among all sc_i with $i \in [N]$ (if sc_i takes the maximum at two or more indices, then let i_* be the last one among them). Then the result of the security game is 1 ("attack succeeded") if $i_* \notin \{i_1, i_2, i_3\}$, and 0 otherwise.

In the notations of Section 2.2, we have $X = (\{0,1\}^m)^3 = \{0,1\}^{3m}$ and $G = \{0,1,?\}^m$.

7.3 Implementations of distinguishers in bucket-brigade model

In the above setting, the algorithms $\mathsf{A}_{i,f}$ in Corollary 1 can be implemented in bucket-brigade model in the following manner. We use the partition $X = \bigcup_{\omega \in \{0,1\}^{3m}} X_\omega$, $X_\omega := \{\omega\}$ of the set X, therefore the number of components is $n := 2^{3m}$. For each $\omega \in \{0,1\}^{3m}$, any deterministic function $f\colon X_\omega \to G = \{0,1,?\}^m$ is identified with the element $y := f(\omega) \in G$. Now for the algorithm $\mathsf{A}_{i,f} = \mathsf{A}_{\omega,y}$, our implementation uses $\mu := (15N+1)m$ random bits $r_{0,j}$ ($j \in [m]$) and $r_{i,j,h}$ ($i \in [N]$, $j \in [m]$, $1 \le h \le 15$). (Note that the fractional parts of the auxiliary values $p^{(0)}$, $p^{(1)}$, u_0 and u_1 are represented by 15 bits.) The configuration graph $\mathcal{G}(\mathsf{A}_{\omega,y})$ of the random tape consists of the edges $P_{0,j} \to P_{0,j+1}$ ($1 \le j \le m-1$), $P_{0,m} \to P_{1,1,1}$, $P_{i,j,h} \to P_{i,j,h+1}$ ($i \in [N]$, $j \in [m]$, $1 \le h \le 14$), $P_{i,j,15} \to P_{i,j+1,1}$ ($i \in [N]$, $1 \le j \le m-1$) and $P_{i,m,15} \to P_{i+1,1,1}$ ($1 \le i \le N-1$), and each cell of the random tape is read at most once (that is, $N_{\mathrm{read}} = 1$). Hence the first and the last cells are $P_{0,1}$ and $P_{N,m,15}$, respectively. Moreover, we define the format of each memory state $\mathsf{mem} \in M$ by

$$\mathsf{mem} = (\pi_1, \ldots, \pi_m, \mathsf{sc}, \mathsf{sc}_{\max}, \mathsf{flag.word}, \mathsf{flag.index}) \ , \tag{48}$$

where each component has the following property:

- $(\pi_1, \ldots, \pi_m) \in \{0,1\}^m$: The component stores the bits $\pi_1, \ldots, \pi_m$ appearing in the security game.
- $(\mathsf{sc}, \mathsf{sc}_{\max})$: The component stores the score sc of the currently considered index $i \in [N]$ and the maximal score $\mathsf{sc}_{\max}$. By the definition of scores and the bit lengths of u_0 and u_1, both $2^{15}\mathsf{sc}$ and $2^{15}\mathsf{sc}_{\max}$ are integers from $-2^{15}mu_0$ to $2^{15}mu_0$, therefore the number of possibilities of this component is at most $(2^{16}mu_0 + 1)^2$.
- $\mathsf{flag.word} \in \{0,1,2\}$: First we notice that, the random variable $w_{i,j}$ with $Pr[1 \leftarrow w_{i,j}] = p^{(\pi_j)}$ and $Pr[0 \leftarrow w_{i,j}] = 1 - p^{(\pi_j)}$ is realized by using the 15 bits $r_{i,j,h}$ ($1 \le h \le 15$), by comparing the value $(0.r_{i,j,1}r_{i,j,2}\cdots r_{i,j,15})_2$ with $p^{(0)}$. Intuitively, the cases $\mathsf{flag.word} = 0$, $\mathsf{flag.word} = 1$ and $\mathsf{flag.word} = 2$, respectively, mean "the value of $w_{i,j}$ is determined as 0", "the value of $w_{i,j}$ is determined as 1" and "the value of $w_{i,j}$ is not yet determined", respectively.

- $\mathsf{flag.index} \in \{0, 1\}$: The cases $\mathsf{flag.index} = 0$ and $\mathsf{flag.index} = 1$, respectively, mean "the codeword w_{i_*} of the highest score satisfies $i_* \in \{i_1, i_2, i_3\}$" and "the codeword w_{i_*} of the highest score satisfies $i_* \notin \{i_1, i_2, i_3\}$", respectively.

In this setting, a μ-party protocol $\Pi = \Pi(\mathsf{A}_{\omega,y})$ given by the bucket-brigade model can be defined in the following manner (note that the parameters ω and y are hard-coded into the protocol and not included in the memory state). For the **input phase**, the input for Π consists of the initial memory state mem and the random bits $r_{0,j}$ ($j \in [m]$) and $r_{i,j,h}$ ($i \in [N]$, $j \in [m]$, $1 \le h \le 15$). The components of mem are initially set as $(\pi_1, \dots, \pi_m) := (0, \dots, 0)$, $(\mathsf{sc}, \mathsf{sc}_{\max}) := (0, -mu_0)$, $\mathsf{flag.word} := 2$ and $\mathsf{flag.index} := 1$.

For the **communication phase**, the player $P_{0,j}$ ($j \in [m]$) updates the component π_j of mem by $\pi_j := r_{0,j}$, and then sends the updated memory state mem to $P_{0,j+1}$ (when $j < m$) or $P_{1,1,1}$ (when $j = m$). On the other hand, if the player $P_{i,j,h}$ ($i \in [N]$, $j \in [m]$, $1 \le h \le 15$) has received mem via a channel which is not of the second type $C_{*,\perp}$ (i.e., this is not the end of the communication phase), then the protocol proceeds as follows, where $\omega \in \{0,1\}^{3m}$ is identified with the triple $(\omega_1, \omega_2, \omega_3)$, $\omega_1, \omega_2, \omega_3 \in \{0,1\}^m$:

1. When $j = 1$ and $h = 1$, the player initializes sc by $\mathsf{sc} := 0$.
2. When $h = 1$, the player initializes $\mathsf{flag.word}$ by $\mathsf{flag.word} := 2$.
3. When $\mathsf{flag.word} = 2$,
   - if $r_{i,j,h} = 0$ and h-th bit of the fractional part of $p^{(0)}$ is 1 (which implies that $(0.r_{i,j,1} \cdots r_{i,j,15})_2 < p^{(0)}$), then the player updates $\mathsf{flag.word}$ by $\mathsf{flag.word} := 1 - \pi_j$;
   - if $r_{i,j,h} = 1$ and h-th bit of the fractional part of $p^{(0)}$ is 0 (which implies that $(0.r_{i,j,1} \cdots r_{i,j,15})_2 > p^{(0)}$), then the player updates $\mathsf{flag.word}$ by $\mathsf{flag.word} := \pi_j$;
   - otherwise, the player does not change $\mathsf{flag.word}$.
4. When $h = 15$ and $\mathsf{flag.word} = 2$ (which implies that $(0.r_{i,j,1} \cdots r_{i,j,15})_2 = p^{(0)}$), the player updates $\mathsf{flag.word}$ by $\mathsf{flag.word} := \pi_j$.
5. When $h = 15$, if $i = i_t$ with $1 \le t \le 3$ and $\mathsf{flag.word} \neq \omega_{t,j}$ (which implies that $w_{i_t} \neq \omega_t$ and hence $(w_{i_1}, w_{i_2}, w_{i_3}) \neq \omega$), then the player sends a message $(0, 0, \dots, 0) \in M$ to $P_{N,m,15}$ via the unique channel of the form $C_{*,\perp}$ (i.e., this is the final message sent during the communication phase) and skips the steps below.
6. When $h = 15$, the player calculates $\mathsf{sc}_{i,j}$ as in (46), where $\mathsf{flag.word}$ plays the role of $w_{i,j}$, and then updates sc by $\mathsf{sc} := \mathsf{sc} + \mathsf{sc}_{i,j}$.
7. When $j = m$ and $h = 15$, if $\mathsf{sc} \ge \mathsf{sc}_{\max}$, then the player updates $\mathsf{flag.index}$ by $\mathsf{flag.index} := 0$ in the case $i \in \{i_1, i_2, i_3\}$ and by $\mathsf{flag.index} := 1$ in the case $i \notin \{i_1, i_2, i_3\}$, respectively.
8. When $i = N$, $j = m$ and $h = 15$, the player sends a message $(\mathsf{flag.index}, 0, \dots, 0) \in M$ to $P_{N,m,15}$ via the unique channel of the form $C_{*,\perp}$ (i.e., this is the final message sent during the communication phase) and skips the step below.
9. The player sends the updated memory state mem to the next player (via the channel of the first type).

Finally, for the **output phase**, if the last player $P_{N,m,15}$ has received, via a channel $C_{*,\perp}$ of the second type, the final message of the form $(b, 0, 0, \dots, 0)$ with $b \in \{0,1\}$, then the player outputs the bit b.

It is straightforward to check that the above protocol indeed implements the algorithm $\mathsf{A}_{\omega,y}$. Here the parameters are given by $\mu = (15N + 1)m$, $|M| = 6 \cdot 2^m (2^{16} m u_0 + 1)^2$ and $N_{\mathrm{read}} = 1$.

Remark 3. We notice that, in the preceding result [11, 12], it is required to theoretically evaluate the time complexity of some algorithms that are counterparts of $\mathsf{A}_{\omega,y}$ above, which seems practically difficult or complicated (cf., Section 4.3 of [11]). In contrast, our theoretical evaluation of $\mathsf{A}_{\omega,y}$ given above seems simpler than that in [11], which also shows an advantage of our proposed technique.

7.4 Estimation of seed lengths

In the numerical example given in the preceding paper [11], the parameters N and m given in the first and the second rows of Table 1 were used. On the other hand, it was aimed in the example in [11] that the difference of the attack success probabilities between random and pseudorandom cases is bounded by $\varepsilon := 10^{-6}$. Here we adopt the same parameters N, m and ε, while we put $R := 2$ in Theorem 4. Moreover, by virtue of the result in Section 7.3, we put $\mu := (15N + 1)m$, $|M| := 6 \cdot 2^m(2^{16}mu_0 + 1)^2$, $N_{\mathrm{read}} := 1$ and $n := 2^{3m}$. Then the value $C(\mu, |M|)$ in (33) is $(\mu - 1)|M|^2 + 1$, therefore the values in the statement of Theorem 4 are estimated as

$$\delta = \lceil 8n^2\varepsilon^{-2}\mu^2 C(\mu, |M|)\rceil \le 8n^2\varepsilon^{-2}\mu^2 \cdot \mu|M|^2 = 8n^2\varepsilon^{-2}\mu^3|M|^2 \tag{49}$$

and

$$\begin{aligned}\nu_0^\dagger &= \lceil \log_2(n^2\varepsilon^{-2}\mu^2\delta^{\mathrm{dp}(T)}R)\rceil \\ &\le \lceil \log_2(2 \cdot 8^{\lceil \log_2 \mu\rceil}n^{2\lceil \log_2 \mu\rceil+2}\varepsilon^{-2\lceil \log_2 \mu\rceil-2}\mu^{3\lceil \log_2 \mu\rceil+2}|M|^{2\lceil \log_2 \mu\rceil})\rceil \\ &= \lceil 1 + 3\lceil \log_2 \mu\rceil + (2\lceil \log_2 \mu\rceil + 2)(3m + 6\log_2 10) + (3\lceil \log_2 \mu\rceil + 2)\log_2 \mu + 2\lceil \log_2 \mu\rceil \log_2 |M|\rceil \ .\end{aligned} \tag{50}$$

Now we have $3m + 6\log_2 10 \le 3m + 20$ and

$$\begin{aligned}\log_2 |M| = \log_2(6 \cdot 2^m(2^{16}mu_0 + 1)^2) &= m + \log_2(6 \cdot (126602m + 1)^2) \\ &\le m + \log_2(8 \cdot (126603m)^2) = m + 3 + 2\log_2(126603m) \ ,\end{aligned} \tag{51}$$

therefore we have

$$\begin{aligned}\nu_0^\dagger &\le 1 + 3\lceil \log_2 \mu\rceil + (2\lceil \log_2 \mu\rceil + 2)(3m + 20) + (3\lceil \log_2 \mu\rceil + 2)\lceil \log_2 \mu\rceil \\ &\qquad + 2\lceil \log_2 \mu\rceil(m + 3 + 2\lceil \log_2(126603m)\rceil) \\ &\le 3\lceil \log_2 \mu\rceil^2 + (8m + 51 + 4\lceil \log_2(126603m)\rceil)\lceil \log_2 \mu\rceil + 6m + 41 \ .\end{aligned} \tag{52}$$

The resulting values of the right-hand side of (52) are given in Table 1.

8 Conclusion

In this paper, we revisited a preceding work by Nuida and Hanaoka [11, 12] on pseudorandomization of information-theoretically secure schemes. First, we gave a proof technique for the security of such pseudorandomization, which admits tuning of some parameters in the proof and contains the argument appeared in [12] as a special case. Secondly, we pointed out that, although the argument and numerical examples in [11, 12] are mainly based on the use of PRGs against time-bounded distinguishers and consequently some hardness assumptions are required, the use of other kinds of PRGs such as INW PRGs against memory-bounded distinguishers can remove the requirement of the hardness assumptions. We also gave a precise formulation of a computational model which suits the quantitative evaluation of memory costs of the distinguishers for INW PRGs. Finally, we gave a numerical comparison of the required seed lengths in our result with those in the numerical example in [11], and showed that the reduction effect of required seed lengths by our pseudorandomization result is still competitive with that of [11, 12], despite the removal of the required hardness assumptions.

Acknowledgments. The author would like to thank Professor Takeshi Koshiba for a fruitful discussion on a previous version of the work, and also the author's colleagues, especially Dr. Goichiro Hanaoka and Dr. Takahiro Matsuda, for their precious comments. The author would also like to thank the anonymous referees for previous submissions of the paper for their detailed comments.

References

[1] L. Babai, N. Nisan, M. Szegedy, *Multi-party protocols and logspace-hard pseudorandom sequences*, in: Proc. STOC 1989, ACM, 1989, pp.1–11

[2] D. Boneh, J. Shaw, *Collusion-secure fingerprinting for digital data*, IEEE Transactions on Information Theory, vol.44, 1998, pp.1897–1905

[3] B. Dubrov, Y. Ishai, *On the randomness complexity of efficient sampling*, in: Proc. STOC 2006, ACM, 2006, pp.711–720

[4] R. R. Farashahi, B. Schoenmakers, A. Sidorenko, *Efficient pseudorandom generators based on the DDH assumption*, in: Proc. PKC 2007, LNCS 4450, 2007, pp.426–441

[5] S. Hoory, N. Linial, A. Wigderson, *Expander graphs and their applications*, Bull. Amer. Math. Soc., vol.43, no.4, 2006, pp.439–561

[6] R. Impagliazzo, N. Nisan, A. Wigderson, *Pseudorandomness for network algorithms*, in: Proc. STOC 1994, ACM, 1994, pp.356–364

[7] N. Koblitz, A. Menezes, *Another look at non-uniformity*, preprint, `http://eprint.iacr.org/2012/359`

[8] A. Lubotzky, R. Phillips, P. Sarnak, *Explicit expanders and the Ramanujan conjectures*, in: Proc. STOC 1986, ACM, 1986, pp.240–246

[9] N. Nisan, *Pseudo-random sequences for space bounded computation*, Combinatorica, vol.12, no.4, 1992, pp.449–461

[10] K. Nuida, S. Fujitsu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, H. Imai, *An improvement of discrete Tardos fingerprinting codes*, Designs, Codes and Cryptography, vol.52, no.3, 2009, pp.339–362

[11] K. Nuida, G. Hanaoka, *An improvement of pseudorandomization against unbounded attack algorithms – the case of fingerprint codes*, in: Proc. ICITS 2009, LNCS 5973, 2010, pp.213–230

[12] K. Nuida, G. Hanaoka, *On the security of pseudorandomized information-theoretically secure schemes*, in: Proc. ICITS 2009, LNCS 5973, 2010, pp.56–73