How to Use Pseudorandom Generators in Unconditional Security Settings

Koji Nuida

National Institute of Advanced Industrial Science and Technology (AIST),
Tsukuba, Ibaraki 305-8568, Japan
k.nuida@aist.go.jp

Abstract. Cryptographic pseudorandom generators (PRGs) can reduce the randomness complexity of computationally secure schemes. Nuida and Hanaoka (IEEE Trans. IT 2013) developed a security proof technique against computationally unbounded adversaries under the use of cryptographic PRGs. However, their proof assumed unproven hardness of the underlying problem for the cryptographic PRG. In the paper, we realize a *fully unconditional* security proof, by extending the previous result to "non-cryptographic" PRGs such as the one by Impagliazzo, Nisan and Wigderson (STOC 1994) based on graph theory rather than one-way functions. In fact, our proof technique is effective only for some restricted class of schemes; then we also propose a "dual-mode" modification of the PRG to prove computational security even for schemes outside the class, while keeping the unconditional security for schemes in the class.

Keywords: Pseudorandom generators, information-theoretic security.

1 Introduction

Cryptographic pseudorandom generators (PRGs) can generate randomness for computationally secure schemes. On the other hand, when the original scheme is information-theoretically secure, it was expected that the security is degraded to computational. Recently, Nuida and Hanaoka [14] developed a security proof technique under the use of a cryptographic PRG, *where the computational power of adversaries are not assumed to be bounded.* However, their proof still assumed *the unproven hardness of an underlying computational problem for the PRG* (e.g., the hardness of the Decisional Diffie–Hellman (DDH) problem, for the PRG in [4] used in the numerical example of [14]). The aim of the work is to remove the latter kind of assumptions, realizing a *fully unconditional* security proof.

1.1 Our Contributions

In the paper, we remove the unproven assumptions in the previous result and realize a fully unconditional security, by extending the result in [14] to "non-cryptographic" PRGs. We use the PRG by Impagliazzo, Nisan and Wigderson

S.S.M. Chow et al. (Eds.): ProvSec 2014, LNCS 8782, pp. 291–299, 2014.

[7], hereafter called an *INW PRG*, whose indistinguishability is based on *unconditionally provable* graph-theoretic properties rather than one-way functions associated to cryptographic PRGs.

In fact, our proof technique (as well as the previous result in [14]) is effective only for some restricted class of schemes, and no security is guaranteed for schemes outside the class. To resolve the issue, we also propose a technique of combining the INW PRG with a cryptographic PRG, in such a way that the security under the use of the resulting PRG is at least computational even for schemes outside the class, while the unconditional security is kept for schemes in the class. We call the resulting PRG a *dual-mode PRG*. Such a hybrid property is also potentially useful when the security notion for the original scheme involves both information-theoretically secure parts and computationally secure parts.

One may feel that, whenever the randomness complexity of an information-theoretically secure scheme can be reduced by our technique using the INW PRGs, the randomness complexity could also be reduced by modifying the individual scheme directly. We emphasize that, even if it is true, our result provides a *unified* way to reduce the randomness complexity, hence is still meaningful.

1.2 Related Work

Dubrov and Ishai (Sect. 3.2.1 of [3]) also mentioned that the randomness complexity of some cryptographic processes can be unconditionally decreased by using PRGs proposed in the same paper. However, the possible applications of their result are restricted in comparison to our result; indeed, their result only corresponds to Theorem 1 in the paper, but not to more general Theorem 2.

Our construction of dual-mode PRGs has in fact a flavor similar to several "indistinguishability amplification" results such as Yao's XOR lemma (e.g., [10]). However, in contrast to those *quantitative* security improvements, our dual-mode PRGs focus on *qualitative* properties (i.e., hybrid security property).

1.3 Organization of the Paper

In Sect. 2, we summarize the proof technique of the previous work [14] and its problem, and then propose a solution by using the INW PRG. In Sect. 3, we summarize the construction and properties of the INW PRGs. In Sect. 4, we give a numerical example of our result. In Sect. 5, we propose the dual-mode PRGs. Finally, in Sect. 6, we discuss other potential applications of our proposed techniques. See the full version of the paper for some omitted details.

1.4 Notations and Terminology

A *directed edge* of an (undirected) graph is an edge, with distinction of the two end vertices as the source and the destination. We say that a graph is *δ-regular*, if each vertex is adjacent to precisely δ edges. For a binary rooted tree T, let $r(T)$, $V(T)$ and $L(T)$ denote the root of T, the set of vertices of T, and the set of leaves of T ordered from left to right, respectively. For each $v \in V(T)$, let $v^{\uparrow}$, $v_{\leftarrow}$ and $v_{\rightarrow}$ denote its parent vertex, left child vertex and right child vertex (if exist), respectively. For two random variables $\mathcal{R}_1, \mathcal{R}_2$, let $\varDelta(\mathcal{R}_1, \mathcal{R}_2)$ denote their statistical distance; $\varDelta(\mathcal{R}_1, \mathcal{R}_2) := (1/2) \sum_x |\Pr[x \leftarrow \mathcal{R}_1] - \Pr[x \leftarrow \mathcal{R}_2]|$. For any map F, let $[F]$ denote an algorithm to compute the value of F.

2 A Framework for Our Unconditional Security Proof

Here we explain the previous proof technique in [14] on which our result is based. We consider the following abstract security game for a cryptographic scheme:

1. The *challenger* generates an object α by using an output of a random source $r \leftarrow \mathcal{R}$. We denote the function to compute α by F_1; i.e., $F_1(r) = \alpha$.
2. The *adversary* obtains some information $\beta \in B$ on α and give it to the attack algorithm $\mathcal{A}$. We denote the function to compute β by F_2; i.e., $F_2(\alpha) = \beta$. Then the adversary sends the output $\gamma \in C$ of $\mathcal{A}(\beta)$ to the challenger.
3. The challenger decides, from γ and α, whether the adversary wins or not. We denote the function to make the decision by F_3; i.e., $F_3(\alpha, \gamma) = 1$ if the adversary wins, and $F_3(\alpha, \gamma) = 0$ otherwise.

The success probability of the adversary's attack (i.e., the winning probability of the adversary in the game above) relative to random source $\mathcal{R}$ is defined by

$$\begin{aligned} \mathsf{Succ}_{\mathcal{A},\mathcal{R}} &:= \Pr_{r \leftarrow \mathcal{R}}[\alpha = F_1(r);\ \beta = F_2(\alpha);\ \gamma \leftarrow \mathcal{A}(\beta);\ \delta = F_3(\alpha, \gamma) \colon \delta = 1] \\ &= \Pr_{r \leftarrow \mathcal{R}}[\beta = F_2(F_1(r));\ \gamma \leftarrow \mathcal{A}(\beta);\ \delta = F_3(F_1(r), \gamma) \colon \delta = 1] \ . \end{aligned}$$

Let $\mathcal{R}_U$ denote the uniformly random source, and let $\mathcal{R}_P$ denote the output distribution of a given PRG. We suppose that the scheme is secure if $\mathcal{R}_U$ is used, i.e., $\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U}$ is sufficiently small for any possible attack algorithm $\mathcal{A}$. Our goal here is to prove that the scheme is still secure if the PRG $\mathcal{R}_P$ is used; i.e., $\mathsf{Succ}_{\mathcal{A},\mathcal{R}_P}$ is sufficiently small. For the purpose, it suffices to show that $|\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U} - \mathsf{Succ}_{\mathcal{A},\mathcal{R}_P}|$ is sufficiently small for any possible $\mathcal{A}$. For each $\beta_0 \in B$ and $\gamma_0 \in C$, let $\mathcal{F}_{\beta_0,\gamma_0}$ denote the map with input r that outputs 1 if $F_2(F_1(r)) = \beta_0$ and $F_3(F_1(r), \gamma_0) = 1$, and outputs 0 otherwise. Then the argument in [14] implies that $\mathsf{Succ}_{\mathcal{A},\mathcal{R}} = \sum_{\beta_0 \in B,\, \gamma_0 \in C} \Pr[\gamma_0 \leftarrow \mathcal{A}(\beta_0)] \Pr[\mathcal{F}_{\beta_0,\gamma_0}(\mathcal{R}) = 1]$ and

$$\begin{aligned} &|\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U} - \mathsf{Succ}_{\mathcal{A},\mathcal{R}_P}| \\ &\leq \sum_{\beta_0 \in B,\, \gamma_0 \in C} \Pr[\gamma_0 \leftarrow \mathcal{A}(\beta_0)] \left| \Pr[\mathcal{F}_{\beta_0,\gamma_0}(\mathcal{R}_U) = 1] - \Pr[\mathcal{F}_{\beta_0,\gamma_0}(\mathcal{R}_P) = 1] \right| \ . \end{aligned}$$

Now we introduce the following two conditions, where we fix values T and ε:

Condition 1. The PRG is indistinguishable in the following sense; if the complexity of an algorithm $\mathcal{D}$ with 1-bit output is bounded by T, then we have

$$\varDelta(\mathcal{D}(\mathcal{R}_U), \mathcal{D}(\mathcal{R}_P)) = \left| \Pr[\mathcal{D}(\mathcal{R}_U) = 1] - \Pr[\mathcal{D}(\mathcal{R}_P) = 1] \right| \leq \varepsilon \ .$$

Condition 2. The map $\mathcal{F}_{\beta_0,\gamma_0}$ for any $\beta_0 \in B$ and $\gamma_0 \in C$ above satisfies that the complexity of the algorithm $[\mathcal{F}_{\beta_0,\gamma_0}]$ is bounded by T.

Note that *these two conditions are independent of the choice of the adversary's attack algorithm* $\mathcal{A}$; e.g., $\mathcal{A}$ *may have unbounded complexity.* From now, suppose that the two conditions above are satisfied. Then we have the following bound:

$$\begin{aligned} |\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U} - \mathsf{Succ}_{\mathcal{A},\mathcal{R}_P}| &\le \sum_{\beta_0 \in B,\, \gamma_0 \in C} \Pr[\gamma_0 \leftarrow \mathcal{A}(\beta_0)] \cdot \varepsilon \\ &= \varepsilon \sum_{\beta_0 \in B} \sum_{\gamma_0 \in C} \Pr[\gamma_0 \leftarrow \mathcal{A}(\beta_0)] = \varepsilon \sum_{\beta_0 \in B} 1 = |B| \cdot \varepsilon \ , \end{aligned} \tag{1}$$

which is *independent of the attack algorithm* $\mathcal{A}$, and is effective if $|B| \cdot \varepsilon$ is sufficiently small (note that $|B|$ depends heavily on the individual scheme).

2.1 Our First Contribution: Using "Non-Cryptographic" PRGs

We point out that, the argument in [14] supposed to use cryptographic PRGs against distinguishers with bounded *time complexity*; consequently, *Condition 1 requires some unproven assumptions* (cf., P=NP? Problem), though (1) itself was derived without any assumptions on the complexity of the attack algorithm $\mathcal{A}$. In other words, *the security proof in [14] will be ineffective if an efficient algorithm to distinguish the cryptographic PRG from random is found.*

To resolve the issue, we use "non-cryptographic" PRGs (less frequently used in cryptography), especially the one by Impagliazzo, Nisan and Wigderson [7] based on expander graphs, hereafter called an *INW PRG*. The underlying complexity measure is close to the space complexity rather than the time complexity, and *the hardness to distinguish the PRG from random is unconditionally provable* by graph-theoretic facts. Now Condition 1 becomes provable as well, therefore our security proof under the use of the PRG is also made unconditional.

3 Impagliazzo–Nisan–Wigderson PRG

In the section, we summarize the construction and properties of the INW PRGs denoted by G^{INW}. Let ℓ_{INW} denote its seed length. The output set of G^{INW} is $\boldsymbol{R} = \prod_{v \in L(T)} R_v$ where R_v is some set indexed by the leaves v of a binary rooted tree T. For each $v \in V(T) \setminus L(T)$, let $\mathit{\Gamma}_v$ be a δ_v-regular graph with ν_v vertices, and define R_v to be the set of the directed edges of $\mathit{\Gamma}_v$ ($\delta_v \nu_v$ edges in total). See Sect. 1.4 for some notations. We say that a map $f \colon X \to Y$ is *most balanced*, if $|f^{-1}(y_1)| - |f^{-1}(y_2)| \in \{-1, 0, 1\}$ for any $y_1, y_2 \in Y$. Then for each $v \in V(T) \setminus L(T)$, we define a map $G_v^{\mathrm{INW}} \colon R_v \to R_{v_\leftarrow} \times R_{v_\rightarrow}$ in the following manner. Given a directed edge $x_v \in R_v$ of $\mathit{\Gamma}_v$, let $y_{v_\leftarrow}$ and $y_{v_\rightarrow}$ denote its source and destination vertices, respectively. Then we map $y_{v_\leftarrow}$ and $y_{v_\rightarrow}$ to an element $x_{v_\leftarrow} \in R_{v_\leftarrow}$ and an element $x_{v_\rightarrow} \in R_{v_\rightarrow}$ by fixed, most balanced maps $V(\mathit{\Gamma}_v) \to R_{v_\leftarrow}$ and $V(\mathit{\Gamma}_v) \to R_{v_\rightarrow}$, respectively. Now we define $G_v^{\mathrm{INW}}(x_v) := (x_{v_\leftarrow}, x_{v_\rightarrow})$.

Then G^{INW} is constructed as follows. Given a seed $s \in \{0,1\}^{\ell_{\mathrm{INW}}}$, first we map s to an element $x_{r(T)} \in R_{r(T)}$ by a fixed, most balanced map $\{0,1\}^{\ell_{\mathrm{INW}}} \to R_{r(T)}$. Secondly, we determine the elements $x_v \in R_v$ for $v \in V(T) \setminus \{r(T)\}$ by successively applying the maps G_u^{INW} for $u \in V(T) \setminus L(T)$ in an ascending order with respect to the depth of u. Finally, we define $G^{\mathrm{INW}}(s) := (x_v)_{v \in L(T)} \in \boldsymbol{R}$.

To evaluate the indistinguishability of the INW PRG quantitatively, here we fix a concrete computational model associated to the tree T as follows:

- In the model, an algorithm is equipped with a common memory M which can take one of $|M|$ possible states, as well as a processor associated to each leaf of T (identified with the leaf itself) that has unbounded computational power and unbounded local memory. Given an input $\boldsymbol{r} = (r_v)_{v \in L(T)}$ for the algorithm, the component r_v is distributed to $v \in L(T)$ at the beginning.
- The execution of the algorithm consists of μ rounds, where μ is a parameter. For each round, the first (leftmost) leaf is activated first, and each leaf is activated after the execution of the previous leaf ends. Each leaf first reads the current state of the common memory M, decides the new memory state by using the current state of M and the local memory state of the leaf, and updates the state of M accordingly (the local memory state is also updated).
- Finally, after the final round ends, the output of the algorithm is decided according to the final state of the common memory M.

The *adjacency matrix* of graph Γ_v is a symmetric $\{0,1\}$-matrix of size ν_v, where the (i,j)-entry is 1 if and only if the i-th and the j-th vertices of Γ_v are adjacent. Since Γ_v is a δ_v-regular graph, if we order the eigenvalues of the adjacency matrix as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\nu_v}$, then $\lambda_1 = \delta_{\nu_v}$. Now we define $\lambda(\Gamma_v) := \max\{|\lambda_2|, |\lambda_{\nu_v}|\}$. On the other hand, it is known that, for any integers $n, m \geq 1$, the statistical distance between the uniform random variable on $[m] := \{1, \dots, m\}$ and the output of any most balanced map $[n] \to [m]$ with uniformly random input is $\rho(n,m) := (n \bmod m) \cdot (m - (n \bmod m))/(nm)$, where $(n \bmod m)$ is the remainder of n modulo m (see Lemma VI.1 of [14]). Now we have the following results (whose proofs are similar to the original paper [7] and are omitted due to the page limitation; see the full version for details):

Theorem 1. *Let $\mathcal{R}_U$ denote the uniform distribution on $\boldsymbol{R}$, and let $\mathcal{R}_P$ denote the output distribution of G^{INW} with uniformly random seed $s \in \{0,1\}^{\ell_{\mathrm{INW}}}$. Then for any algorithm $\mathcal{D}$ described in the computational model above, we have*

$$\Delta(\mathcal{D}(\mathcal{R}_U), \mathcal{D}(\mathcal{R}_P)) \leq |M|^{\mu} \sum_{v \in V(T) \setminus L(T)} \frac{\lambda(\Gamma_v)}{2\delta_v} + \Delta_{\mathsf{dist}} \ ,$$

$$\Delta_{\mathsf{dist}} := \rho(2^{\ell_{\mathrm{INW}}}, \nu_{r(T)}\delta_{r(T)}) + \sum_{\substack{v \in V(T) \setminus L(T) \\ v \neq r(T)}} \rho(\nu_{v^{\uparrow}}, \nu_v \delta_v) + \sum_{v \in L(T)} \rho(\nu_{v^{\uparrow}}, |R_v|) \ .$$

Theorem 2. *Let $\mathcal{R}_U$ and $\mathcal{R}_P$ be as in Theorem 1. In the situation of Sect. 2, suppose that the algorithm $[\mathcal{F}_{\beta_0,\gamma_0}]$ can be described in the computational model above with common memories of size bounded by $|M|$ and at most μ rounds*

Table 1. Comparison of seed lengths (here "Our result" shows the seed lengths by our result; "Plain" shows the originally used random bits; "[14]" shows the seed lengths in the previous result [14]; and the approximate values are written in scientific E notation)

N	10^3	10^4	10^5	10^6	10^7	10^8	10^9
m	614	702	789	877	964	1052	1139
Plain	9.21E6	1.05E8	1.18E9	1.31E10	1.44E11	1.57E12	1.70E13
[14]	6.87E6	9.72E6	1.33E7	1.75E7	2.25E7	2.83E7	3.51E7
Our result	3.09E5	4.90E5	6.65E5	8.67E5	1.14E6	1.40E6	1.68E6

for any $\beta_0 \in B$ and $\gamma_0 \in C$. Then, without any assumption on hardness of computational problems nor on the complexity of the algorithm $\mathcal{A}$, we have the following, where Δ_{dist} is defined as in Theorem 1:

$$|\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U} - \mathsf{Succ}_{\mathcal{A},\mathcal{R}_P}| \le |B| \cdot \left(|M|^{\mu} \sum_{v \in V(T) \setminus L(T)} \frac{\lambda(\Gamma_v)}{2\delta_v} + \Delta_{\mathsf{dist}} \right)$$

We note that the bounds in Theorems 1 and 2 become better when $\lambda(\Gamma_v)$ becomes smaller. A graph Γ_v is called a *Ramanujan graph*, if $\lambda(\Gamma_v) \le 2\sqrt{\delta_v - 1}$; this is known to almost attain the theoretical lower bound of $\lambda(\Gamma_v)$ (see e.g., Sect. 5.3 of [6]). For example, we can use Ramanujan graphs given by a part of the result by Morgenstern [12] (see the full version of the paper for details):

Proposition 1 ([12]). *For any positive integers L, D, there is an explicit construction of a $(2^D + 1)$-regular Ramanujan graph with $2^{6DL} - 2^{2DL}$ vertices.*

4 Example: Collusion-Secure Codes

In the section, we give a numerical example of our technique applied to a collusion-secure code in [13] (with the number $c = 3$ of corrupted users), which is the same as the example in [14] and has information-theoretic security. Roughly speaking, in the abstract security game in Sect. 2, α is the collection of m-bit words, one per each of the N users; β is the collection of the three words for the corrupted users; γ is a word of length m on an expanded alphabet $\{0, 1, ?\}$, where '?' means a bit erasure; and $F_3(\alpha, \gamma) = 1$ if and only if the "most suspicious" user determined from α and γ is not a corrupted user. See the numerical example in [14] for details. Then an analysis shows that, to bound the difference $|\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U} - \mathsf{Succ}_{\mathcal{A},\mathcal{R}_P}|$ by a value $\varepsilon_{\mathsf{diff}}$, the seed length for the INW PRG becomes $\ell_{\mathrm{INW}} \sim 12 \log_2 N(\log_2 N + 4m + \log_2(1/\varepsilon_{\mathsf{diff}}))$ when $N \to \infty$ (see the full version of the paper), while $\ell_{\mathrm{org}} := 15mN + m$ random bits are originally used in total. On the other hand, for the choices of $\varepsilon_{\mathsf{diff}} := 10^{-6}$ and other parameters as the numerical example in [14], the seed lengths ℓ_{INW} are calculated as in Table 1. This table shows that our seed lengths are much shorter than the originally required random bits and also significantly smaller than those in [14].

5 Our Second Contribution: Dual-Mode PRGs

We note that, in the situation of Sect. 2, no security is guaranteed if the set B (hence the right-hand side of (1)) is too large (though (1) itself holds unconditionally). To resolve the issue, in the section we propose a technique to modify the INW PRG in such a way that, by using the resulting PRG, the information-theoretic security is kept if B is sufficiently small, while at least computational security is guaranteed even if B is too large. For two random variables $\mathcal{R}_1, \mathcal{R}_2$ on the output set $\boldsymbol{R}$ of the INW PRG G^{INW}, let $\mathcal{R}_1 * \mathcal{R}_2$ denote the random variable on $\boldsymbol{R}$ computing the component-wise group operation $*$ for values of $\mathcal{R}_1$ and $\mathcal{R}_2$. We call $\mathcal{R}_P * \mathcal{R}_C$ the *dual-mode PRG*, where $\mathcal{R}_C$ denotes the output distribution of a cryptographic (computationally secure) PRG G^{comp}. Then we have the following result (deduced from the fact that both $\mathcal{R}_U * \mathcal{R}_C$ and $\mathcal{R}_P * \mathcal{R}_U$ are identical to $\mathcal{R}_U$; see the full version of the paper for details):

Theorem 3. *Under the same assumptions as Theorem 2, we have:*

- *The value* $|\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U} - \mathsf{Succ}_{\mathcal{A},\mathcal{R}_P * \mathcal{R}_C}|$ *satisfies the same inequality as the value* $|\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U} - \mathsf{Succ}_{\mathcal{A},\mathcal{R}_P}|$ *in Theorem 2.*
- *Suppose that the maps* F_1, F_2 *and* F_3 *in Sect. 2,* G^{INW} *and the operator* $*$ *in* $\boldsymbol{R}$ *are all polynomial-time computable. Then* $|\mathsf{Succ}_{\mathcal{A},\mathcal{R}_U} - \mathsf{Succ}_{\mathcal{A},\mathcal{R}_P * \mathcal{R}_C}|$ *is negligible for any probabilistic polynomial-time algorithm* $\mathcal{A}$.

As an example, we apply the dual-mode PRG to Shamir's k-out-of-n secret sharing scheme [17] over the field $\mathbb{F}_q$. Let k', $1 \leq k' < k$, denote the number of corrupted users. Then an analysis (see the full version of the paper) shows that, to bound the bias of the k' corrupted shares from uniform by $\varepsilon_{\mathsf{diff}} = k^{-\omega(1)}$ (negligible in k), the seed length for the part G^{INW} of the dual-mode PRG is

$$\ell_{\mathrm{INW}} \sim 12 \log_2 k (k' \log_2 q + \omega(1) \log_2 k) \quad (\text{when } k \to \infty) \ ,$$

having lower order than the number $\ell_{\mathrm{org}} \sim k \log_2 q$ of the originally used random bits if $k' = o(k / \log_2 k)$. Now the corrupted shares are statistically close to uniform (information-theoretic security) when at most k' users are corrupted, while these are computationally indistinguishable from uniform (at least computational security) even if more than k' (and at most $k-1$) users are corrupted.

6 Other Potential Applications

Finally, in this section, we discuss a possible application of our result to lossy encryption [1,9,15] with small randomness space. In [5], Hemenway and Ostrovsky showed that any lossy encryption scheme for which the randomness space for encryption is smaller than the plaintext space can be converted into a (slightly) lossy trapdoor function (e.g., [16]); and the latter is further converted (via other results in [8,11]) into various cryptographic primitives such as CCA-secure encryption and adaptive trapdoor functions. However, construction of such schemes with small randomness spaces is difficult; the only known construction so far (to the author's best knowledge) is the one based on the Damgård–Jurik cryptosystem [2]. Indeed, since the ciphertexts under a lossy key should be *statistically* indistinguishable, a naive strategy of reducing the randomness space by cryptographic PRGs is not effective. The author hopes that our *unconditional* proof technique using "non-cryptographic" PRGs is effective to resolve the problem; a detailed study is a future research topic.

Acknowledgments. The author thanks Takeshi Koshiba for a fruitful discussion on a previous version of the work; thanks the members of Shin-Akarui-Angou-Benkyo-Kai, especially Takahiro Matsuda for his suggestion of the content in Sect. 6 and many comments, Keita Emura and Goichiro Hanaoka for their precious comments; and thanks the anonymous referees for their comments.

References

1. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)
2. Damgård, I., Jurik, M., Nielsen, J.B.: A generalization of Paillier's public-key system with applications to electronic voting. Int. J. Inform. Sec. 9(6), 371–385 (2010)
3. Dubrov, B., Ishai, Y.: On the randomness complexity of efficient sampling. In: Proceedings of STOC 2006, pp. 711–720 (2006)
4. Farashahi, R.R., Schoenmakers, B., Sidorenko, A.: Efficient pseudorandom generators based on the DDH assumption. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 426–441. Springer, Heidelberg (2007)
5. Hemenway, B., Ostrovsky, R.: Building injective trapdoor functions from oblivious transfer. Electronic Colloquium on Computational Complexity, TR10-127, Revision 1 (2010), `http://eccc.hpi-web.de/report/2010/127/`
6. Hoory, S., Linial, N., Wigderson, A.: Expander graphs and their applications. Bull. Amer. Math. Soc. 43(4), 439–561 (2006)
7. Impagliazzo, R., Nisan, N., Wigderson, A.: Pseudorandomness for network algorithms. In: Proceedings of STOC 1994, pp. 356–364 (1994)
8. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
9. Kol, G., Naor, M.: Cryptography and game theory: Designing protocols for exchanging information. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 320–339. Springer, Heidelberg (2008)
10. Levin, L.A.: One-way functions and pseudorandom generators. Combinatorica 7(4), 357–363 (1987)
11. Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 296–311. Springer, Heidelberg (2010)
12. Morgenstern, M.: Existence and explicit constructions of q+1 regular Ramanujan graphs for every prime power q. J. Combin. Theory, Series B 62, 44–62 (1994)

13. Nuida, K., Fujitsu, S., Hagiwara, M., Kitagawa, T., Watanabe, H., Ogawa, K., Imai, H.: An improvement of discrete Tardos fingerprinting codes. Des. Codes Cryptography 52(3), 339–362 (2009)
14. Nuida, K., Hanaoka, G.: On the security of pseudorandomized information-theoretically secure schemes. IEEE Trans. Inform. Theory 59(1), 635–652 (2013)
15. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)
16. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of STOC 2008, pp. 187–196 (2008)
17. Shamir, A.: How to share a secret. Commun. ACM 22(11), 612–613 (1979)