Keeping or Losing Tiny-Error Correctness of Cryptosystems Implemented by Secure Pseudorandom Generators

Koji Nuida[1][2]

[1] The University of Tokyo, Japan
(nuida@mist.i.u-tokyo.ac.jp)
[2] National Institute of Advanced Industrial Science and Technology (AIST), Japan

August 1, 2018

Abstract

Randomness is essential but expensive resource for cryptography, and secure (and efficient) implementations of randomness using pseudorandom generators (PRGs) are much concerned in this area. On the other hand, implementations of randomness without losing the correctness of the underlying cryptosystem should be important but seem to be less concerned in the literature. The results in this paper show that the problem of the correct implementation of randomness in cryptosystems is in general non-trivial even by using secure PRGs. Namely, we construct two examples with the following properties:

- There are a secure and correct public key encryption (PKE) scheme (with negligible decryption error probability) and a secure PRG satisfying that, implementing the key generation algorithm by using the PRG makes the scheme incorrect. The reason of this phenomenon is that, the standard formulation of correctness of PKE schemes does in general *not* imply that erroneous keys (that yield non-negligible decryption error probability for some plaintext) are efficiently detectable.
- There are a secure and correct PKE scheme and a PRG secure against *uniform* distinguishers, satisfying that, implementing the encryption algorithm by using the PRG makes the scheme incorrect. The reason of this phenomenon is that, when a PKE scheme is incorrect, a plaintext that yields non-negligible decryption error probability is in general *not* efficiently samplable by a uniform algorithm; hence security of the PRG against *non-uniform* distinguishers is required. We also discuss a possibility to avoid the reliance on PRGs secure against non-uniform distinguishers.

1 Introduction

Randomness is a fundamental resource for cryptography; theoretical designs of cryptosystems usually assume the use of a large amount of ideally random bits. As it is fairly expensive in practice to generate (almost) ideal random bits, a practical way of *securely* implementing the randomness, i.e., cryptographically secure *pseudorandom generators* (*PRGs*), has been well studied in cryptography. On the other hand, *correctness* of cryptosystems is a further important property than security; an insecure cryptosystem may be problematic in the presence of adversaries, while an incorrect cryptosystem may be problematic even in the absence of adversaries. Now we note that cryptosystems may err with non-zero probability, and such errors with negligible probabilities are often tolerated by rigorous formulations of correct cryptosystems. However, a way of *correctly* implementing the randomness in cryptography (i.e., keeping the negligible error probability) seems to be less concerned in contrast to the case of security. This paper aims at focusing on the correct implementation of randomness in cryptography, and giving a caution that a naive use of a secure PRG may be in general *not* sufficient to preserve the correctness of an implemented cryptosystem. We note that this paper discusses only the case of public key encryption (PKE) schemes as a simplest and most typical example, though similar observations might be applicable to other kinds of cryptographic schemes as well.

Digest of technical results. Roughly speaking, in this paper we construct (under certain reasonably weak assumptions) two secure (IND-CPA) PKE schemes Π_1, Π_2 and two PRGs $\mathcal{R}_1, \mathcal{R}_2$ where

- Π_1 and Π_2 are correct (in some appropriate sense);
- $\mathcal{R}_1$ is secure against any probabilistic polynomial-time (PPT) distinguisher (either uniform or non-uniform), and $\mathcal{R}_2$ is secure against PPT uniform distinguishers;

satisfying the following properties;

- when implementing the key generation algorithm by using the PRG $\mathcal{R}_1$, the scheme Π_1 will be incorrect;
- when implementing the encryption algorithm by using the PRG $\mathcal{R}_2$, the scheme Π_2 will be incorrect.

The result shows that, when a negligibly small but non-zero error probability exists in either a key generation algorithm or an encryption algorithm, the use of a secure PRG to implement this component of a PKE scheme is in general *not* sufficient to preserve the correctness of the scheme. See Theorem 1 in Section 3 and Theorem 2 in Section 4 for the details of these results.

More backgrounds on correctness. The safest way of defining the correctness of a PKE scheme is to require the following: for *any* pair of a public key and a secret key, and for *any* plaintext, the decryption of a random ciphertext for the plaintext results in the original plaintext with probability *one*. This kind of correctness is often called the *perfect correctness*. For example, Definition 5.1.1 of Goldreich's famous book [9] adopts this formulation. It is obvious that implementations of randomness for perfectly correct schemes do not cause any problem for preserving the correctness. We also note that, recently Bitansky and Vaikuntanathan [4] proposed a generic conversion method for a large class of cryptographic schemes, including PKE schemes, that obtains a perfectly correct scheme from a not perfectly correct scheme (see below for some related topics).

On the other hand, there are also different ways of defining the correctness of a PKE scheme in the literature, which allow negligible but non-zero error probabilities. For example, a comment given after Definition 5.1.1 of the aforementioned book [9] says:

> *Definition 5.1.1 may be relaxed in several ways without significantly harming its usefulness. For example, we may relax Condition (2) and allow a negligible decryption error (e.g.,* $\mathsf{Pr}[D_d(E_e(\alpha)) \neq \alpha] < 2^{-n}$*). Alternatively, one may postulate that Condition (2) holds for all but a negligible measure of the key-pairs generated by* $G(1^n)$*. At least one of these relaxations is essential for some suggestions of (public-key) encryption schemes.*

The first part of the quoted comment concerns the existence of decryption errors for any fixed key pair and any fixed plaintext. Such a formulation of correctness is also found in, e.g., papers [2, 6, 14, 22] from recent leading conferences in cryptography. The second part of the quoted comment concerns the existence of "erroneous" key pairs, which may cause decryption errors. Such a formulation of correctness is also found in, e.g., a recent book by Katz and Lindell [15]; Definition 11.1 of the book says:

> *It is required that, except possibly with negligible probability over* (pk, sk) *output by* $\mathsf{Gen}(1^n)$*, we have* $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m$ *for any (legal) message* m*.*

As shown above, such relaxed versions of correctness for PKE schemes have been popular in the area of cryptography. Here we emphasize that, these two relaxed formulations of correctness can be unified (roughly) into the following form: for almost all key pairs, and for *any* choice of plaintext, the probability of decryption error for this plaintext is negligible. In other words: we say (roughly) that a key pair is erroneous, if for *some* plaintext, this key pair yields non-negligible probability of decryption error. Then the aforementioned relaxed correctness is equivalent to that the fraction of the erroneous key pairs is negligible.

First lesson: Importance of detectability of erroneous keys. The first part of our technical result shows that, when implementing the key generation algorithm for a PKE scheme by using even a secure PRG, the ratio of the erroneous key pairs may increase intolerably, e.g., from exponentially rare to all, in the case of our example. The reason of such non-negligible increase of erroneous keys by using PRGs is as follows: it is *not* an intrinsic property of PKE schemes that efficient detection of erroneous keys is possible. Namely, if there is a polynomial-time recognizable set $\mathcal{K}^\dagger$ of keys that involves all erroneous keys (as well as, possibly, some non-erroneous keys) and if a random key belongs to this set $\mathcal{K}^\dagger$ with negligible probability, then implementing the key generation algorithm by using a secure PRG will ensure that the probability for a key to belong to $\mathcal{K}^\dagger$ is still negligible, meaning that the scheme remains correct. However, the existence of such a set $\mathcal{K}^\dagger$ is not required by the standard formulation of the notion of PKE schemes (though such a set does exist at least for most of the known PKE schemes), and our example in this paper is certainly the case where such an efficiently recognizable set $\mathcal{K}^\dagger$ does not exist. This observation suggests that, when designing a new PKE scheme, it is important to make the erroneous keys efficiently detectable (we note that, this does *not* mean that such a detection process for erroneous keys should be included in the key generation algorithm; just the *existence* of such an efficient detection process works for this purpose).

Second lesson: Implicit non-uniformity in correctness. The second part of our technical result shows that, when implementing the encryption algorithm for a PKE scheme by using even a PRG secure against uniform distinguishers, a non-erroneous key pair may become erroneous; in fact, this happens for all key pairs in the case of our example. Roughly speaking, the reason of such non-negligible increase of decryption error probability is that, the formulation of correctness relies implicitly on the notion of *non-uniform* computation, therefore the security of a PRG against *uniform* distinguishers only is in general not sufficient to preserve the correctness.

Recall that a non-uniform algorithm is endowed with, in addition to the input, an auxiliary advice string, which depends solely on the input length and need not be efficiently computable. In contrast, a uniform algorithm is not endowed with such advice. Now we imagine that, implementing the encryption algorithm of a PKE scheme by using a PRG makes some originally non-erroneous key pair erroneous; namely, the decryption error probability for some plaintext is non-negligibly increased. In this case, one may naively expect that, an output of the PRG and a random string could be efficiently distinguished (which would contradict the security of the PRG) by taking these key pair and plaintext, trying encryption and decryption by using the target random string (either output by the PRG or truly random), and then observing if decryption error occurs. However, we should note that *such a plaintext* (as well as such a key pair) *may in general be not efficiently found*, therefore the virtual distinguisher constructed above is not a PPT uniform algorithm in general. Hence the security of a PRG against uniform distinguishers may be insufficient.

The same argument above also implies that, such non-negligible increase of decryption error probability will not occur when the security of the PRG is against *non-uniform* distinguishers (which may be given the key pair and the plaintext as advice). However, it has been pointed out, e.g., by Koblitz and Menezes [16], that developing and practically using cryptographic primitives (such as PRGs) with security against non-uniform adversaries require stronger computational hardness assumptions for non-uniform algorithms. This affects the evaluation of the practical security level for such primitives, due to not just the potentially stronger power of non-uniform algorithms in comparison to uniform ones for solving the underlying hard problems, but also some concrete attempts in the literature to utilize the power of non-uniformity for breaking the security. For example, the authors of [16] gave non-uniform attacks against some message authentication codes and leakage-resilient stream ciphers. There are also previous results using non-uniformity towards breaking PRGs (e.g., [5, 20]) or solving hard problems on which PRGs can be based (e.g., [3]). Although the latter are yet superpolynomial-time attacks and hence do not mean total break of the security, the existence of such attacks should still be concerned when choosing a practically reasonable security parameter.

Due to these facts, it might be desirable if we do not rely on PRGs secure against non-uniform distinguishers. For this purpose we point out that, the known technique in cryptography of constructing a PRG by taking bitwise XOR of two PRGs with security of different types (e.g., [4, 7, 19]) is also effective in the present case. Namely, we choose a PRG $\mathcal{R}_1$ secure against uniform distinguishers, and another PRG $\mathcal{R}_2$

that fools a certain class of functions constructed from the encryption and decryption algorithms for a given PKE scheme. We show (Theorem 3 in Section 4.2) that, by using the PRG $\mathcal{R}(s_1, s_2) = \mathcal{R}_1(s_1) \oplus \mathcal{R}_2(s_2)$ to implement the encryption algorithm, the security is preserved owing to the security of $\mathcal{R}_1$, while the correctness is also preserved owing to the choice of $\mathcal{R}_2$. We note that, in contrast to requiring for a PRG to fool non-uniform and *unknown* distinguishers, it should be relatively easier to develop the PRG $\mathcal{R}_2$ that fools non-uniform but *known* algorithms only. Several construction techniques for PRGs in the area of derandomization (e.g., [18, 21]) would be effective to construct such a PRG against known algorithms.

Related Work: Immunization towards perfect correctness. Needless to say, an ultimate countermeasure for preventing the loss of correctness by the use of PRGs is to make the original scheme itself completely error-free. A direction of such an "immunization" method to eliminate errors is to modify each individual scheme; a famous example in this direction is the work by Goldreich, Goldwasser, and Halevi [10] to establish an error-free version of Ajtai–Dwork lattice-based cryptosystem [1].

Another direction is to establish a generic method to convert a cryptographic scheme having some errors into a perfectly correct scheme with the same functionality. For this direction, Dwork, Naor, and Reingold [7] proposed a method to eliminate decryption errors in PKE schemes for almost all key pairs (though a negligible but non-zero fraction of erroneous keys may remain). Holenstein and Renner [12], and Lin and Tessaro [17] also realized a similar kind of results by different approaches. We note that these results do not eliminate errors perfectly (e.g., for the key generation algorithm) and hence the obtained, almost perfectly correct PKE schemes may still suffer from the loss of correctness by the use of PRGs as in our examples. On the other hand, a recent work by Bitansky and Vaikuntanathan [4] achieved a generic conversion that eliminates errors in not only encryption but also key generation, hence obtaining perfectly correct PKE schemes (their result is also applicable to many other kinds of cryptographic schemes). However, the method of [4] has large overhead and might be not suitable for practical purposes. We note also that, those conversion methods were motivated by e.g., the facts that decryption errors in a PKE scheme may be utilized to break the scheme (e.g., [13]) and the existence of decryption errors (even though with negligible probability) will be problematic when used as a building block of some other cryptographic scheme. These preceding results did not concern the loss of correctness by the use of PRGs as discussed in the present paper.

2 Preliminaries

2.1 Basic Notations and Settings

In this paper, we say that a function $\varepsilon(\lambda) \in [0, 1]$ of an integer $\lambda \geq 1$ is *negligible*, if for any integer $k \geq 1$, there exists an integer $\lambda_0 \geq 1$ satisfying $\varepsilon(\lambda) < \lambda^{-k}$ for any $\lambda > \lambda_0$. We let a *positive polynomial* mean a non-zero polynomial with non-negative coefficients. For a function $f(\lambda)$ of positive integer λ, we say that $f(\lambda)$ is *polynomially bounded*, if $f(\lambda) \leq \mathsf{poly}(\lambda)$ for some positive polynomial $\mathsf{poly}(\lambda)$; and we say that $f(\lambda)$ is *polynomial-time computable*, if there is a deterministic polynomial-time algorithm $\mathcal{A}$ that for an input 1^λ outputs $f(\lambda)$.

For a probability distribution D, we write $a \hookleftarrow D$ to indicate that the element a is chosen according to the distribution D. Let $U[X]$ denote the uniform distribution on a set X. For two probability distributions X and Y over a (finite) set Z, their *statistical distance* $\Delta(X, Y)$ is defined by

$$\Delta(X, Y) \overset{\mathrm{def}}{=} \frac{1}{2} \sum_{z \in Z} \left| \Pr[z \hookleftarrow X] - \Pr[z \hookleftarrow Y] \right| = \max_{E \subseteq Z} \left(\Pr_{z \hookleftarrow X}[z \in E] - \Pr_{z \hookleftarrow Y}[z \in E] \right) .$$

It is known that $\Delta(f(X, D), f(Y, D)) \leq \Delta(X, Y)$ for any function f and any probability distribution D independent of X and Y. We say that probability distributions $X = X_\lambda$ and $Y = Y_\lambda$ indexed by a parameter λ are *exponentially close*, if there is a function $\varepsilon(\lambda) = 2^{-\Omega(\lambda)}$ for which we have $\Delta(X, Y) \leq \varepsilon(\lambda)$ for any λ.

In this paper, we suppose that the set of random tapes for a probabilistic algorithm is of the form $\{0, 1\}^{L(\lambda)}$ for some value $L(\lambda)$ depending on the security parameter λ. For any probabilistic algorithm $\mathcal{A}$

with input x and random tape r, we may write $\mathcal{A}(x;r)$ instead of $\mathcal{A}(x)$ in order to emphasize the choice of the random tape. We often abbreviate the term "probabilistic polynomial-time" to "PPT". In this paper, for simplifying the argument, we adopt a convention about non-uniform algorithms in a way that an advice for a non-uniform algorithm depends solely on the security parameter λ. By using an appropriate padding to the input, our convention here can be made consistent with the rigorous convention in computational complexity theory where an advice for a non-uniform algorithm depends solely on the input length for the algorithm. An advice $z = z_\lambda$ for an algorithm $\mathcal{A}$ may be either made implicit in notation or indicated by using a notation such as $\mathcal{A}^{(z_\lambda)}$.

2.2 Pseudorandom Generators

In this paper, we define a *pseudorandom generator* (*PRG*) to be a deterministic polynomial-time algorithm, with security parameter 1^λ and a seed $s \in \{0,1\}^{\ell_{\mathrm{in}}(\lambda)}$ as input, that outputs an element of $\{0,1\}^{\ell_{\mathrm{out}}(\lambda)}$, where $\ell_{\mathrm{in}}(\lambda)$ and $\ell_{\mathrm{out}}(\lambda)$ are polynomially bounded and polynomial-time computable functions satisfying that $\lambda \leq \ell_{\mathrm{in}}(\lambda) < \ell_{\mathrm{out}}(\lambda)$ and $\ell_{\mathrm{in}}(\lambda)$ is a strictly increasing function[1]. A *distinguisher* for a PRG $\mathcal{R}$ is an algorithm $\mathcal{D}$ (either uniform or non-uniform) that is given 1^λ and some $r \in \{0,1\}^{\ell_{\mathrm{out}}(\lambda)}$ as input and outputs $\mathcal{D}(1^\lambda, r) \in \{0,1\}$. The *advantage* of a distinguisher $\mathcal{D}$ for a PRG $\mathcal{R}$ is defined by

$$\mathsf{Adv}_{\mathcal{R},\mathcal{D}}(1^\lambda) \overset{\mathrm{def}}{=} \left|\Pr[\mathcal{D}(1^\lambda, \mathcal{R}(1^\lambda, U[\{0,1\}^{\ell_{\mathrm{in}}(\lambda)}])) = 1] - \Pr[\mathcal{D}(1^\lambda, U[\{0,1\}^{\ell_{\mathrm{out}}(\lambda)}]) = 1]\right| \enspace .$$

Definition 1. Let $\mathcal{R}$ be a PRG. We say that $\mathcal{R}$ is *secure against uniform* (respectively, *non-uniform*) *distinguishers*, if $\mathsf{Adv}_{\mathcal{R},\mathcal{D}}(1^\lambda)$ is negligible for any PPT uniform (respectively, non-uniform) distinguisher $\mathcal{D}$.

The following lemma can be proved by a standard hybrid argument.

Lemma 1. *Let $\mathcal{R}$ be a PRG with input length $\ell_{\mathrm{in}}(\lambda)$ and output length $\ell_{\mathrm{out}}(\lambda)$. Let $\rho(\lambda) \geq 1$ and $\nu(\lambda) \geq 0$ be polynomially bounded and polynomial-time computable functions both of that are weakly increasing. We define a PRG $\mathcal{R}^{\rho,\nu}$ with input length $\rho(\lambda) \cdot \ell_{\mathrm{in}}(\lambda) + \nu(\lambda)$ and output length $\rho(\lambda) \cdot \ell_{\mathrm{out}}(\lambda) + \nu(\lambda)$ by*

$$\mathcal{R}^{\rho,\nu}(1^\lambda, s_1, \dots, s_{\rho(\lambda)}, r) \overset{\mathrm{def}}{=} (\mathcal{R}(1^\lambda, s_1), \dots, \mathcal{R}(1^\lambda, s_{\rho(\lambda)}), r)$$

where $s_1, \dots, s_{\rho(\lambda)} \in \{0,1\}^{\ell_{\mathrm{in}}(\lambda)}$ and $r \in \{0,1\}^{\nu(\lambda)}$. If $\mathcal{R}$ is secure against uniform (respectively, non-uniform) distinguishers, then $\mathcal{R}^{\rho,\nu}$ is also secure against uniform (respectively, non-uniform) distinguishers.

The following lemma can also be proved by a standard argument (cf., Theorem 3.3.3 of [8]).

Lemma 2. *Let $\mathcal{R}$ be a PRG with input length $\ell_{\mathrm{in}}(\lambda)$ and output length $\ell_{\mathrm{out}}(\lambda) = \ell_{\mathrm{in}}(\lambda) + 1$. Let $\rho(\lambda)$ be a polynomially bounded and polynomial-time computable function satisfying $\rho(\lambda) > \ell_{\mathrm{in}}(\lambda)$. We define a PRG $\mathcal{R}^\rho$ with input length $\ell_{\mathrm{in}}(\lambda)$ and output length $\rho(\lambda)$ by*

$$\mathcal{R}^\rho(1^\lambda, s) \overset{\mathrm{def}}{=} \sigma_1\sigma_2 \cdots \sigma_{\rho(\lambda)} \in \{0,1\}^{\rho(\lambda)}$$

where we recursively define $x_0 = s$ and $\mathcal{R}(1^\lambda, x_{i-1}) = \sigma_i x_i$ (i.e., x_i and σ_i are the last $\ell_{\mathrm{in}}(\lambda)$ bits and the first bit of $\mathcal{R}(1^\lambda, x_{i-1}) \in \{0,1\}^{\ell_{\mathrm{in}}(\lambda)+1}$, respectively). If $\mathcal{R}$ is secure against uniform (respectively, non-uniform) distinguishers, then $\mathcal{R}^\rho$ is also secure against uniform (respectively, non-uniform) distinguishers.

[1] The constraint here for input lengths of PRGs is seemingly different from a standard convention where the input length is equal to the security parameter λ. However, our style here can be made consistent with the standard convention by regarding the value $\lambda' = \ell_{\mathrm{in}}(\lambda)$ as a new "security parameter" and then "interpolating" the discontinuously chosen security parameters λ' by considering some harmless scheme with missing security parameters between $\ell_{\mathrm{in}}(\lambda)$ and $\ell_{\mathrm{in}}(\lambda+1)$. This does not affect the essence of our results, and we adopt the convention as in the main text in order to avoid such inessential intricacy.

2.3 Public Key Encryption

In this paper, a *public key encryption* (*PKE*) *scheme* means a triple $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms satisfying the following syntax. The *key generation algorithm* Gen is an algorithm that, given a security parameter 1^λ as input, outputs a pair $(\mathsf{pk}, \mathsf{sk})$ of a public key pk (including the security parameter 1^λ and specifications of a plaintext space $\mathcal{M}$ and a ciphertext space $\mathcal{C}$) and a secret key sk (supposed to implicitly include pk). The *encryption algorithm* Enc is an algorithm that, given pk and a plaintext $m \in \mathcal{M}$ as input, outputs a ciphertext $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m)$. The *decryption algorithm* Dec is an algorithm that, given sk and a ciphertext $c \in \mathcal{C}$ as input, outputs either a plaintext $m' \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c)$ or a decryption failure symbol $\perp \not\in \mathcal{M}$.

In this paper, we consider the correctness of PKE schemes with various amounts of decryption errors.

Definition 2. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme. Let $\alpha(\lambda)$ and $\beta(\lambda)$ be functions of security parameter λ. We say that Π is $\alpha(\lambda)$*-key* $\beta(\lambda)$*-correct*, if for any security parameter λ, there is a set $\mathcal{K}^\dagger$ with the following properties:

- We have $\Pr_{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)}[(\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}^\dagger] \leq 1 - \alpha(\lambda)$.
- If $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and $(\mathsf{pk}, \mathsf{sk}) \not\in \mathcal{K}^\dagger$, then we have $\Pr[\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m)) \neq m] \leq 1 - \beta(\lambda)$ for any plaintext m.

For the term "$\alpha(\lambda)$-key", we also say "all-key" when $\alpha(\lambda) = 1$, "almost all-key" when $\alpha(\lambda) = 1 - 2^{-\Omega(\lambda)}$, and "overwhelming-key" when $1 - \alpha(\lambda)$ is negligible. On the other hand, for the term "$\beta(\lambda)$-correct", we also say "perfectly-correct" when $\beta(\lambda) = 1$, "almost perfectly-correct" when $\beta(\lambda) = 1 - 2^{-\Omega(\lambda)}$, and "overwhelmingly-correct" when $1 - \beta(\lambda)$ is negligible.

For example, all-key perfect-correctness in the sense above corresponds to the perfect correctness in a usual sense, while overwhelming-key overwhelming-correctness in the sense above corresponds to the relaxed version of the correctness (with negligible decryption error probability) in a usual sense.

On the other hand, we introduce the following terminology relevant to *incorrectness* of PKE schemes.

Definition 3. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme. Let λ be a security parameter and let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$. Let $\alpha(\lambda)$ be a function of λ. We say that the key pair $(\mathsf{pk}, \mathsf{sk})$ is *somewhere* $\alpha(\lambda)$*-erroneous*, if we have $\Pr[\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m)) \neq m] \geq \alpha(\lambda)$ for some plaintext m.

This paper does not much concern the security of PKE schemes; we only consider IND-CPA security as a simple and widely recognized notion. We recall the definition as follows.

Definition 4. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme. We consider the following *IND-CPA game* between a challenger and an adversary $\mathcal{A}$ with common security parameter λ:

1. The challenger generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and sends pk to the adversary.
2. The adversary executes $\mathcal{A}(\mathsf{submit}, 1^\lambda, \mathsf{pk})$ to generate a tuple (m_0, m_1, st) of plaintexts $m_0, m_1 \in \mathcal{M}$ and a record of the internal state st. Then the adversary sends m_0 and m_1 to the challenger.
3. The challenger chooses $b^* \in \{0, 1\}$ uniformly at random, generates $c^* \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_{b^*})$, and sends c^* to the adversary.
4. The adversary executes $\mathcal{A}(\mathsf{guess}, 1^\lambda, \mathsf{pk}, \mathsf{st}, c^*)$ to generate a bit $b \in \{0, 1\}$.

We say that the adversary *wins* the game if $b = b^*$. We define the *advantage* of the adversary by $\mathsf{Adv}_{\mathcal{A}}^{\Pi}(1^\lambda) \overset{\mathrm{def}}{=} |\Pr[b = b^*] - 1/2|$ where the probability is determined according to the game above. We say that the PKE scheme Π is *IND-CPA*, if $\mathsf{Adv}_{\mathcal{A}}^{\Pi}(1^\lambda)$ is negligible for any PPT adversary $\mathcal{A}$.

The following lemma can be proved by a standard hybrid argument (cf., Section 5.2.5.3 of [9]).

Lemma 3. *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with* 1*-bit plaintext space* $\mathcal{M} = \{0,1\}$ *for any security parameter* λ*. Let* $\rho(\lambda)$ *be a polynomially bounded and polynomial-time computable function. We consider the following PKE scheme* $\Pi^\rho = (\mathsf{Gen}^\rho, \mathsf{Enc}^\rho, \mathsf{Dec}^\rho)$*:*

- $\mathsf{Gen}^\rho(1^\lambda)$ *outputs the same public/secret keys* $(\mathsf{pk}, \mathsf{sk})$ *as* $\mathsf{Gen}(1^\lambda)$ *except that the plaintext space is* $\mathcal{M}^\rho = \{0,1\}^{\rho(\lambda)}$ *and the ciphertext space is* $\mathcal{C}^\rho = \mathcal{C}^{\rho(\lambda)}$*.*
- $\mathsf{Enc}^\rho_{\mathsf{pk}}(m_1, \dots, m_{\rho(\lambda)})$ *outputs* $(c_1, \dots, c_{\rho(\lambda)})$ *where* $m_i \in \{0,1\}$ *and* $c_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_i)$ *for each* $i \in \{1, \dots, \rho(\lambda)\}$*.*
- $\mathsf{Dec}^\rho_{\mathsf{sk}}(c_1, \dots, c_{\rho(\lambda)})$ *outputs* $\perp$ *if at least one of* $m'_i \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c_i)$ *is* $\perp$*, otherwise outputs* $(m'_1, \dots, m'_{\rho(\lambda)})$*.*

If Π *is IND-CPA, then so is* Π^ρ*. Moreover, if* Π *is all-key perfectly-correct, then so is* Π^ρ *as well.*

3 Incorrect PKE Schemes with Pseudorandom Keys

In the following text, when Π is a PKE scheme and $\mathcal{R}$ is a PRG to generate the random tape for the key generation algorithm, we denote by $\Pi \circ_{\mathrm{Gen}} \mathcal{R}$ the PKE scheme obtained from Π in a way that the key generation algorithm uses the output of $\mathcal{R}$ (for uniformly random seed) as its random tape.

In this section, we prove the following theorem:

Theorem 1. *Assume that*

- *there is a PRG that is secure against uniform (respectively, non-uniform) distinguishers; and*
- *there is an IND-CPA and all-key perfectly-correct PKE scheme with* 1*-bit plaintext space.*

Then there are an IND-CPA PKE scheme Π *and a PRG* $\mathcal{R}$ *that is secure against uniform (respectively, non-uniform) distinguishers, satisfying that*

- Π *is almost all-key perfectly-correct; but*
- *all key pairs for* $\Pi \circ_{\mathrm{Gen}} \mathcal{R}$ *are somewhere* 1*-erroneous.*

Namely, though the PKE scheme Π is almost perfectly correct and the PRG $\mathcal{R}$ is secure, the PKE scheme will no longer be correct when the PRG is used in the key generation algorithm. We note that, in contrast to the (in)correctness concerned in the theorem, the security of the PRG $\mathcal{R}$ ensures by a standard proof strategy that the IND-CPA security of the PKE scheme Π is preserved when the pseudorandom output of $\mathcal{R}$ is used in the key generation algorithm.

3.1 Proof of the Theorem

From now, we prove Theorem 1. First, by applying Lemma 2 to the PRG $\mathcal{R}_0$ in the hypothesis of the theorem, we may assume without loss of generality that the input length $\ell_{0,\mathrm{in}}(\lambda)$ and the output length $\ell_{0,\mathrm{out}}(\lambda)$ of $\mathcal{R}_0$ satisfy $\ell_{0,\mathrm{out}}(\lambda) \geq \ell_{0,\mathrm{in}}(\lambda) + \lambda$. On the other hand, by applying Lemma 3 to the PKE scheme in the hypothesis of the theorem, we may assume without loss of generality that there is an IND-CPA and all-key perfectly-correct PKE scheme $\Pi_0 = (\mathsf{Gen}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ with plaintext space $\{0,1\}^{\ell_{0,\mathrm{in}}(\lambda)}$. Let $L(\lambda)$ be the length of the random tape for Gen_0, which is polynomially bounded as Gen_0 is PPT. By adding (if necessary) some extra random bits which are actually not used, we may also assume without loss of generality that $L(\lambda)$ is a weakly increasing and polynomial-time computable function.

Now we define a PKE scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ in the following manner.

- The random tape for Gen is of length $\ell_{0,\mathrm{out}}(\lambda) + L(\lambda)$. Given security parameter 1^λ and random tape $(r_1, r_2) \in \{0,1\}^{\ell_{0,\mathrm{out}}(\lambda)} \times \{0,1\}^{L(\lambda)}$, Gen generates $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{Gen}_0(1^\lambda; r_2)$ except that an extra symbol $\perp$ is added to the ciphertext space, and then output $\mathsf{pk} \leftarrow (\mathsf{pk}_0, r_1)$ and $\mathsf{sk} \leftarrow \mathsf{sk}_0$.

- Given $\mathsf{pk} = (\mathsf{pk}_0, r_1)$ and a plaintext $m \in \{0,1\}^{\ell_{0,\mathrm{in}}(\lambda)}$, Enc first computes $r' \leftarrow \mathcal{R}_0(1^\lambda, m)$, and outputs $\perp$ if $r' = r_1$. On the other hand, if $r' \neq r_1$, then Enc outputs the output of $(\mathsf{Enc}_0)_{\mathsf{pk}_0}(m)$.
- Given a ciphertext c, Dec outputs $\perp$ if $c = \perp$; otherwise outputs $(\mathsf{Dec}_0)_{\mathsf{sk}_0}(c)$.

Let $\mathcal{K}^\dagger$ be the set of the key pairs $(\mathsf{pk}, \mathsf{sk})$ for the PKE scheme Π satisfying that the component r_1 of pk is equal to $\mathcal{R}_0(1^\lambda, s)$ for some $s \in \{0,1\}^{\ell_{0,\mathrm{in}}(\lambda)}$. Then by the construction of Π, if the key pair is not in the set $\mathcal{K}^\dagger$, then the output distribution of Enc is equal to that of Enc_0 with the same plaintext and the corresponding public key. Moreover, the ratio of the elements in $\{0,1\}^{\ell_{0,\mathrm{out}}(\lambda)}$ of the form $\mathcal{R}_0(1^\lambda, s)$ for some $s \in \{0,1\}^{\ell_{0,\mathrm{in}}(\lambda)}$ among the whole set is at most $2^{\ell_{0,\mathrm{in}}(\lambda)}/2^{\ell_{0,\mathrm{out}}(\lambda)} \leq 2^{-\lambda}$ since $\ell_{0,\mathrm{out}}(\lambda) \geq \ell_{0,\mathrm{in}}(\lambda) + \lambda$, therefore we have $\Pr[(\mathsf{pk}, \mathsf{sk}) \in \mathcal{K}^\dagger] \leq 2^{-\lambda}$. This argument and the IND-CPA security of Π_0 imply that Π is also IND-CPA. Moreover, the all-key perfect-correctness of Π_0 implies that Π is $(1 - 2^{-\lambda})$-key 1-correct, hence almost all-key perfectly-correct.

Now we define a PRG $\mathcal{R}$ as follows: the input length is $\ell_{\mathrm{in}}(\lambda) = \ell_{0,\mathrm{in}}(\lambda) + L(\lambda)$, and the output length is $\ell_{\mathrm{out}}(\lambda) = \ell_{0,\mathrm{out}}(\lambda) + L(\lambda)$ (note that these functions satisfy the constraint for the length functions for PRGs). Given an input $(s_1, s_2) \in \{0,1\}^{\ell_{0,\mathrm{in}}(\lambda)} + \{0,1\}^{L(\lambda)}$, the output of $\mathcal{R}$ is $(\mathcal{R}_0(1^\lambda, s_1), s_2)$. Then Lemma 1 implies that $\mathcal{R}$ is secure against uniform (respectively, non-uniform) distinguishers as well as $\mathcal{R}_0$. Moreover, when a random tape of the form $(\mathcal{R}_0(1^\lambda, s_1), s_2)$ is used in the key generation algorithm for Π, the construction of Π implies that the encryption algorithm Enc with plaintext $m = s_1$ always outputs the ciphertext $\perp$, therefore this key pair is somewhere 1-erroneous. As any key pair for $\Pi \circ_{\mathrm{Gen}} \mathcal{R}$ is of such a form, it follows that all key pairs for $\Pi \circ_{\mathrm{Gen}} \mathcal{R}$ are somewhere 1-erroneous. This completes the proof of Theorem 1.

3.2 Discussion

Theorem 1 shows that, when implementing a probabilistic key generation algorithm for a PKE scheme Π by using a PRG $\mathcal{R}$, it is in general *not* ensured that the resulting scheme $\Pi \circ_{\mathrm{Gen}} \mathcal{R}$ is correct even if the original scheme Π is almost all-key perfectly-correct (which is usually of practically acceptable level of correctness) and the PRG $\mathcal{R}$ is cryptographically secure. On the other hand, it is obvious that the scheme $\Pi \circ_{\mathrm{Gen}} \mathcal{R}$ will be correct (more precisely, all-key $\beta(\lambda)$-correct) if Π is *all-key* $\beta(\lambda)$-correct. This suggests a significant difference between all-key correctness and not all-key correctness, and also practical importance (from the viewpoint of implementation using PRGs) of avoiding erroneous key pairs in designing PKE schemes.

A main reason of the unexpected loss of correctness shown by Theorem 1 is that, it is in general *not* efficiently checkable whether or not a given key pair for a PKE scheme is erroneous. Namely, if the subset $\mathcal{K}^\dagger$ of possibly erroneous keys appeared in Definition 2 were efficiently recognizable (i.e., with efficient algorithm for membership test), the use of a secure PRG in the key generation algorithm would preserve the correctness with only negligible increase of probability for a key pair to be a member of $\mathcal{K}^\dagger$. This suggests the importance of designing PKE schemes in a way that erroneous keys (if any) can be efficiently detected. (We note that, this does not claim that the detection process for erroneous keys should be included in the actual key generation algorithm. Just the existence of such an efficient detection process is sufficient for ensuring the correctness when implementing the key generation algorithm by using a secure PRG.)

4 Incorrect PKE Schemes with Pseudorandom Encryption

In the following text, when Π is a PKE scheme and $\mathcal{R}$ is a PRG to generate the random tape for the encryption algorithm, we denote by $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$ the PKE scheme obtained from Π in a way that the encryption algorithm uses the output of $\mathcal{R}$ (for uniformly random seed) as its random tape.

In this section, we prove the following theorem:

Theorem 2. *Assume that*

- *there is a PRG that is secure against uniform distinguishers but is not secure against non-uniform distinguishers; and*

- *there is an IND-CPA and all-key perfectly-correct PKE scheme with* 1*-bit plaintext space.*

Then for any constant $\varepsilon > 0$, there are an IND-CPA PKE scheme Π and a PRG $\mathcal{R}$ that is secure against uniform distinguishers, satisfying that

- Π *is all-key almost perfectly-correct and all-key $(1 - \varepsilon)$-correct; but*
- *all key pairs for $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$ are somewhere $\eta(\lambda)$-erroneous for a common non-negligible function $\eta(\lambda)$.*

Namely, though the PKE scheme Π is almost perfectly correct and the PRG $\mathcal{R}$ is secure (against uniform distinguishers), the PKE scheme will no longer be correct when the PRG is used in the encryption algorithm. We note that, in contrast to the (in)correctness concerned in the theorem, the security of the PRG $\mathcal{R}$ against uniform distinguishers ensures by a standard proof strategy that the IND-CPA security of the PKE scheme Π is preserved when the pseudorandom output of $\mathcal{R}$ is used in the encryption algorithm.

4.1 Proof of the Theorem

From now, we prove Theorem 2. Let $\mathcal{R}_0$ denote the PRG as in the hypothesis of the theorem with input length $\ell_{0,\mathrm{in}}(\lambda)$ and output length $\ell_{0,\mathrm{out}}(\lambda)$. This $\mathcal{R}_0$ is secure against uniform distinguishers but is not secure against non-uniform distinguishers. In particular, there is a PPT non-uniform distinguisher $\mathcal{D}$ for $\mathcal{R}_0$, with some advice depending solely on the security parameter λ, that has non-negligible advantage $\mathsf{Adv}_{\mathcal{R}_0,\mathcal{D}}(1^\lambda)$. In more detail, there are an integer $k \geq 1$ and an infinite set Λ of positive integers satisfying that $\mathsf{Adv}_{\mathcal{R}_0,\mathcal{D}}(1^\lambda) > \lambda^{-k}$ for every $\lambda \in \Lambda$. Now, as $\mathcal{D}$ is PPT, the computational complexity of $\mathcal{D}$ is bounded by a positive polynomial $Q(\lambda)$. Then, we may assume without loss of generality that, the advice a_λ for $\mathcal{D}$, associated to security parameter λ, that yields the non-negligible advantage has length at most $Q(\lambda)$; for each $\lambda \in \Lambda$, we have $a_\lambda \in \bigcup_{0 \leq i \leq Q(\lambda)} \{0,1\}^i$ and

$$\begin{aligned} \mathsf{Adv}_{\mathcal{R}_0,\mathcal{D}^{(a_\lambda)}}(1^\lambda) &= \left| \Pr[\mathcal{D}^{(a_\lambda)}(1^\lambda, \mathcal{R}_0(1^\lambda, U[\{0,1\}^{\ell_{0,\mathrm{in}}(\lambda)}])) = 1] - \Pr[\mathcal{D}^{(a_\lambda)}(1^\lambda, U[\{0,1\}^{\ell_{0,\mathrm{out}}(\lambda)}]) = 1] \right| \\ &> \lambda^{-k} \enspace . \end{aligned} \tag{1}$$

Moreover, we may also assume without loss of generality that the random tape for $\mathcal{D}$ is of length $Q(\lambda)$.

On the other hand, by applying Lemma 3 to the PKE scheme in the hypothesis of the theorem, we may assume without loss of generality that there is an IND-CPA and all-key perfectly-correct PKE scheme $\Pi_0 = (\mathsf{Gen}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ with plaintext space $\{0,1\}^{Q(\lambda)}$. Let $L(\lambda)$ be the length of the random tape for Enc_0, which is polynomially bounded as Enc_0 is PPT. By adding (if necessary) some extra random bits which are actually not used, we may also assume without loss of generality that $L(\lambda)$ is a weakly increasing and polynomial-time computable function.

Now we define a PKE scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ in the following manner. The output $(\mathsf{pk}, \mathsf{sk})$ of the key generation algorithm is the same as Π_0 except that the ciphertext space includes an extra symbol $\perp$. For the decryption algorithm, $\mathsf{Dec}_{\mathsf{sk}}(c)$ outputs $(\mathsf{Dec}_0)_{\mathsf{sk}}(c)$ if $c \in \mathcal{C}$, while it outputs $\perp$ if $c = \perp$. On the other hand, for the encryption algorithm, by using a parameter $\rho(\lambda)$ specified in the argument below, the random tape is given by

$$\vec{r} \overset{\mathrm{def}}{=} (\vec{r}_0, \vec{r}_1, \vec{r}_{\mathcal{D}}, r_{\mathrm{Enc}}, r_{\mathrm{Adv}})$$

where $\vec{r}_0 = (r_{0,1}, \dots, r_{0,\rho(\lambda)})$ and $\vec{r}_1 = (r_{1,1}, \dots, r_{1,\rho(\lambda)})$ with each $r_{i,j} \in \{0,1\}^{\ell_{0,\mathrm{out}}(\lambda)}$,

$$\vec{r}_{\mathcal{D}} = (r_{\mathcal{D},0,1}, \dots, r_{\mathcal{D},0,\rho(\lambda)}, r_{\mathcal{D},1,1}, \dots, r_{\mathcal{D},1,\rho(\lambda)}) \text{ with each } r_{\mathcal{D},i,j} \in \{0,1\}^{Q(\lambda)} \enspace ,$$

$r_{\mathrm{Enc}} \in \{0,1\}^{L(\lambda)}$, and $r_{\mathrm{Adv}} \in \{0,1\}^{\lfloor \log_2 Q(\lambda) \rfloor + \lambda + \gamma(\lambda)}$ where

$$\gamma(\lambda) \overset{\mathrm{def}}{=} \rho(\lambda) \cdot \max\{\ell_{0,\mathrm{out}}(\lambda') - \ell_{0,\mathrm{out}}(\lambda) \mid 1 \leq \lambda' \leq \lambda\} \geq 0 \enspace .$$

(We note that this additional factor $\gamma(\lambda)$ is introduced in order to let the input length function $\ell_{\mathrm{in}}(\lambda)$ for the PRG $\mathcal{R}$ constructed later satisfy the constraint that it should be a strictly increasing function.) Then the encryption algorithm $\mathsf{Enc}_{\mathsf{pk}}(m)$ for a given plaintext $m \in \{0,1\}^{Q(\lambda)}$ is executed as follows:

1. The algorithm computes $\nu \leftarrow r_{\mathrm{Adv}} \bmod (Q(\lambda)+1)$ where r_{Adv} is identified with an integer via the binary representation of integers. Let $m_\nu \in \{0,1\}^\nu$ denote the first ν bits of m.
2. For $i \in \{0,1\}$, the algorithm executes the algorithm $\mathcal{D}^{(m_\nu)}(1^\lambda, r_{i,j}; r_{\mathcal{D},i,j})$ with advice m_ν and random tape $r_{\mathcal{D},i,j}$ to obtain a bit $b_{i,j}$ for $j = 1, \dots, \rho(\lambda)$, and counts the number μ_i of the indices j with $b_{i,j} = 1$.
3. For a parameter $\theta(\lambda)$ specified in the argument below, if $|\mu_0 - \mu_1| \leq \theta(\lambda)$ then the algorithm outputs $(\mathsf{Enc}_0)_{\mathsf{pk}}(m; r_{\mathrm{Enc}})$; otherwise the algorithm outputs $\bot$.

We will use the parameters $\rho(\lambda)$ and $\theta(\lambda)$ both of that are polynomially bounded and polynomial-time computable; we suppose these properties in the current argument. Then the algorithm Enc is PPT. In order to choose $\rho(\lambda)$ and $\theta(\lambda)$, we refer to the following special case of Hoeffding's Inequality [11]:

Lemma 4. *Let $X_1, \dots, X_n$ be independent random variables, each taking the value 1 with probability p and the value 0 with probability $1-p$ for a common p. Then for any $t > 0$, we have*

$$\Pr\left[\left|\frac{X_1 + \cdots + X_n}{n} - p\right| \geq t\right] \leq 2\exp\left(-2nt^2\right) \enspace .$$

When the random tape $\vec{r}$ for Enc is uniformly random, for any given m_ν, the output bits of $\mathcal{D}^{(m_\nu)}(1^\lambda, r_{i,j}) = \mathcal{D}^{(m_\nu)}(1^\lambda, r_{i,j}; r_{\mathcal{D},i,j})$ with $i \in \{0,1\}$ and $j \in \{1, \dots, \rho(\lambda)\}$ are independent of each other and follow the same probability distribution; let p denote the probability for taking the value 1 in this distribution. Now if $|\mu_0 - \mu_1| > \theta(\lambda)$, then either $|\mu_0 - \rho(\lambda) \cdot p| > \theta(\lambda)/2$ or $|\mu_1 - \rho(\lambda) \cdot p| > \theta(\lambda)/2$ must hold. Moreover, Lemma 4 (with $n = \rho(\lambda)$ and $t = \dfrac{\theta(\lambda)}{2\rho(\lambda)}$) implies

$$\Pr\left[|\mu_i - \rho(\lambda) \cdot p| > \frac{\theta(\lambda)}{2}\right] \leq 2\exp\left(-2\rho(\lambda) \cdot \left(\frac{\theta(\lambda)}{2\rho(\lambda)}\right)^2\right) = 2\exp\left(-\frac{\theta(\lambda)^2}{2\rho(\lambda)}\right) \enspace .$$

Hence, for any plaintext m, we have

$$\Pr[\mathsf{Enc}_{\mathsf{pk}}(m) = \bot] \leq 4\exp\left(-\frac{\theta(\lambda)^2}{2\rho(\lambda)}\right) \enspace . \tag{2}$$

On the other hand, we define a PRG $\mathcal{R}$ as follows: the seed for $\mathcal{R}$ is

$$\vec{s}_{\mathrm{PRG}} \overset{\mathrm{def}}{=} (\vec{s}, \vec{r}_1, \vec{r}_{\mathcal{D}}, r_{\mathrm{Enc}}, r_{\mathrm{Adv}})$$

where $\vec{s} = (s_1, \dots, s_{\rho(\lambda)})$ with each $s_j \in \{0,1\}^{\ell_{0,\mathrm{in}}(\lambda)}$, and the remaining components $\vec{r}_1$, $\vec{r}_{\mathcal{D}}$, r_{Enc}, and r_{Adv} are the same as in the random tape for Enc. Its input length $\ell_{\mathrm{in}}(\lambda)$ is given by

$$\ell_{\mathrm{in}}(\lambda) = \rho(\lambda) \cdot (\ell_{0,\mathrm{in}}(\lambda) + \ell_{0,\mathrm{out}}(\lambda) + 2Q(\lambda)) + L(\lambda) + \lfloor \log_2 Q(\lambda) \rfloor + \lambda + \gamma(\lambda) \enspace .$$

By the definition of $\gamma(\lambda)$, assuming (as shown below) that $\rho(\lambda)$ is weakly increasing, $\rho(\lambda) \cdot \ell_{0,\mathrm{out}}(\lambda) + \gamma(\lambda)$ is weakly increasing as well. This implies that $\mathcal{R}$ satisfies the constraint for the input and output lengths for PRGs. Now, given a seed $\vec{s}_{\mathrm{PRG}}$, the algorithm $\mathcal{R}$ replaces each component s_j of $\vec{s}$ with the output $r_{0,j}$ of $\mathcal{R}_0(1^\lambda, s_j)$ and keeps the remaining part of $\vec{s}_{\mathrm{PRG}}$ unchanged. Then by Lemma 1, $\mathcal{R}$ is a PRG secure against uniform distinguishers as well as $\mathcal{R}_0$.

In order to analyze the behavior of Enc when the random tape is generated by $\mathcal{R}$, we use the following lemma:

Lemma 5. *Let M, N be two positive integers. Put $\delta = M/N - \lfloor M/N \rfloor$, hence $0 \leq \delta < 1$. Moreover, we set $U_M = U[\{0, \dots, M-1\}]$ and $U_N = U[\{0, \dots, N-1\}]$. Then we have*

$$\Delta(U_M \bmod N, U_N) = \frac{\delta(1-\delta)}{M/N} \leq \frac{N}{4M} \enspace .$$

Proof. The latter part follows from the fact that $\delta(1-\delta)$ attains the maximum value $1/4$ at $\delta = 1/2$. For the former part, we note that

$$\Pr[U_M \bmod N = a] = \begin{cases} \dfrac{\lfloor M/N \rfloor + 1}{M} > \dfrac{1}{N} & \text{for } 0 \leq a \leq M - \lfloor M/N \rfloor N - 1 \ , \\ \dfrac{\lfloor M/N \rfloor}{M} \leq \dfrac{1}{N} & \text{for } M - \lfloor M/N \rfloor N \leq a \leq N - 1 \ . \end{cases}$$

This implies that

$$\begin{aligned} \Delta(U_M \bmod N, U_N) &= (M - \lfloor M/N \rfloor N) \cdot \left(\frac{\lfloor M/N \rfloor + 1}{M} - \frac{1}{N} \right) \\ &= N\delta \cdot \left(\frac{M/N - \delta + 1}{M} - \frac{1}{N} \right) = N\delta \cdot \frac{-\delta + 1}{M} = \frac{\delta(1-\delta)}{M/N} \ . \end{aligned}$$

Hence the assertion holds. □

We consider the case that the security parameter λ is in the set Λ, the plaintext is $m(\lambda) \overset{\text{def}}{=} a_\lambda 0^{Q(\lambda) - |a_\lambda|}$ (i.e., the correct advice a_λ is a prefix of $m(\lambda)$), and the random tape for Enc is generated by $\mathcal{R}$ with uniformly random seed $\vec{s}_{\text{PRG}}$. By Lemma 5, the statistical distance of the probability distribution of the value ν in the algorithm Enc from the uniform distribution over $\{0, \dots, Q(\lambda)\}$ is bounded by

$$\frac{Q(\lambda) + 1}{4 \cdot 2^{\lfloor \log_2 Q(\lambda) \rfloor + \lambda + \gamma(\lambda)}} \leq \frac{2Q(\lambda)}{4 \cdot 2^{\log_2 Q(\lambda) - 1 + \lambda}} = \frac{1}{2^\lambda} \ .$$

Hence we have $m_\nu = m(\lambda)_\nu = a_\lambda$ with probability at least $(Q(\lambda)+1)^{-1} - 2^{-\lambda}$. Conditioned on this case $m_\nu = a_\lambda$, the output bits of $\mathcal{D}^{(m_\nu)}(1^\lambda, \mathcal{R}_0(1^\lambda, s_j); r_{\mathcal{D},0,j})$ with $j \in \{1, \dots, \rho(\lambda)\}$ and $\mathcal{D}^{(m_\nu)}(1^\lambda, r_{1,j}; r_{\mathcal{D},1,j})$ with $j \in \{1, \dots, \rho(\lambda)\}$ are independent of each other. Moreover, the output bits $\mathcal{D}^{(m_\nu)}(1^\lambda, \mathcal{R}_0(1^\lambda, s_j); r_{\mathcal{D},0,j})$, $j \in \{1, \dots, \rho(\lambda)\}$, follow the same probability distribution; let p_0 be the common probability of taking the value 1. Similarly, the output bits $\mathcal{D}^{(m_\nu)}(1^\lambda, r_{1,j}; r_{\mathcal{D},1,j})$, $j \in \{1, \dots, \rho(\lambda)\}$, follow the same probability distribution; let p_1 be the common probability of taking the value 1. Now Eq.(1) implies that $|p_0 - p_1| > \lambda^{-k}$. According to this inequality, we set

$$\theta(\lambda) \overset{\text{def}}{=} \frac{\lambda^{-k}}{2} \rho(\lambda) \ .$$

Due to the choice of the threshold $\theta(\lambda)$, if $|\mu_0 - \mu_1| \leq \theta(\lambda)$, then either $|\mu_0 - \rho(\lambda) \cdot p_0| > \theta(\lambda)/2$ or $|\mu_1 - \rho(\lambda) \cdot p_1| > \theta(\lambda)/2$ must hold. Moreover, Lemma 4 (with $n = \rho(\lambda)$ and $t = \dfrac{\theta(\lambda)}{2\rho(\lambda)}$) implies

$$\Pr\left[|\mu_i - \rho(\lambda) \cdot p_i| > \frac{\theta(\lambda)}{2} \right] \leq 2 \exp\left(-\frac{\theta(\lambda)^2}{2\rho(\lambda)} \right) \ .$$

Hence, in the case where the random tape for Enc is generated by $\mathcal{R}$, we have

$$\Pr[\mathsf{Enc}_{\mathsf{pk}}(m(\lambda)) = \bot] \geq \max\{0, (Q(\lambda)+1)^{-1} - 2^{-\lambda}\} \cdot \left(1 - 4 \exp\left(-\frac{\theta(\lambda)^2}{2\rho(\lambda)} \right) \right) \ . \tag{3}$$

When we set

$$\rho(\lambda) \overset{\text{def}}{=} 8\lambda^{2k} \left(\lceil \log(1/\varepsilon) \rceil + 2\lambda \right) \ ,$$

both $\rho(\lambda)$ and $\theta(\lambda) = \rho(\lambda)\lambda^{-k}/2 = 4\lambda^k \left(\lceil \log(1/\varepsilon) \rceil + 2\lambda \right)$ are polynomially bounded and polynomial-time computable. We have

$$4 \exp\left(-\frac{\theta(\lambda)^2}{2\rho(\lambda)} \right) = 4 \exp\left(-\left(\lceil \log(1/\varepsilon) \rceil + 2\lambda \right) \right) \leq 4e^{\log \varepsilon - 2\lambda} = 4\varepsilon \cdot e^{-2\lambda} \ .$$

Since $4e^{-2\lambda} \leq 4e^{-2} < 1$, this implies that the right-hand side of Eq.(2) is at most ε and is exponentially small. Hence, when the encryption algorithm uses an ideally random tape, the PKE scheme Π is all-key almost perfectly-correct and all-key $(1-\varepsilon)$-correct. The same argument also implies (due to the construction of Π) that the distributions of ciphertexts in Π and in Π_0 for the same plaintext are exponentially close to each other, therefore Π is IND-CPA as well as Π_0. On the other hand, the right-hand side of Eq.(3) is at least

$$\max\{0, (Q(\lambda)+1)^{-1} - 2^{-\lambda}\} \cdot (1 - 4\varepsilon \cdot e^{-2\lambda}) = \Omega(Q(\lambda)^{-1}) \ .$$

Now the function $\eta(\lambda)$ defined by

$$\eta(\lambda) = \begin{cases} \max\{0, (Q(\lambda)+1)^{-1} - 2^{-\lambda}\} \cdot (1 - 4\varepsilon \cdot e^{-2\lambda}) & \text{if } \lambda \in \Lambda \ , \\ 0 & \text{if } \lambda \notin \Lambda \end{cases}$$

is non-negligible (as Λ is an infinite set). When the encryption algorithm uses a random tape generated by the PRG $\mathcal{R}$, if $\lambda \in \Lambda$, then for any key pair, the decryption error occurs for the plaintext $m(\lambda)$ with probability at least $\eta(\lambda)$ by the argument above; i.e., this key pair is somewhere $\eta(\lambda)$-erroneous. On the other hand, if $\lambda \notin \Lambda$, then any key pair is automatically somewhere $\eta(\lambda)$-erroneous as $\eta(\lambda) = 0$. Hence, for the PKE scheme $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$, all key pairs are somewhere $\eta(\lambda)$-erroneous for the non-negligible function $\eta(\lambda)$. This completes the proof of Theorem 2.

4.2 Discussion

Theorem 2 shows that, when implementing a probabilistic encryption algorithm for a PKE scheme Π by using a PRG $\mathcal{R}$, it is in general *not* ensured that the resulting scheme $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$ is correct even if the original scheme Π is all-key almost perfectly-correct (which is usually of practically acceptable level of correctness) and the PRG $\mathcal{R}$ is secure against *uniform* distinguishers. On the other hand, as mentioned in the introduction, the correctness will be ensured when a PRG secure against *non-uniform* distinguishers is used instead; however, reliance on such a non-uniform PRG may cause a less efficient choice of a security parameter for providing an enough security level in practical implementations. In this section, we discuss a possible alternative countermeasure to avoid such an unexpected loss of correctness when using a PRG (we note that, designing PKE schemes with (overwhelming-key) *perfect-correctness* can trivially avoid such an issue and therefore is practically important).

The strategy discussed here, which has also been mentioned in the introduction, is preparing two PRGs with indistinguishability properties of different types and then combining these PRGs by taking the bitwise XOR of the outputs. Similar techniques have been used by previous works in the area of cryptography in different contexts; such a technique was called "dual-mode PRG" in [19]. Specializing to our present case, let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme, where the random tape for the encryption algorithm is of length $L(\lambda)$. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be (polynomial-time) PRGs with input lengths $\ell_{1,\mathrm{in}}(\lambda)$ and $\ell_{2,\mathrm{in}}(\lambda)$, respectively, and common output length $L(\lambda)$, satisfying that $\ell_{1,\mathrm{in}}(\lambda) + \ell_{2,\mathrm{in}}(\lambda) < L(\lambda)$. Then we define a PRG $\mathcal{R}$ with input length $\ell_{\mathrm{in}}(\lambda) = \ell_{1,\mathrm{in}}(\lambda) + \ell_{2,\mathrm{in}}(\lambda)$ and output length $L(\lambda)$ by

$$\mathcal{R}(1^\lambda, \vec{s}) \overset{\mathrm{def}}{=} \mathcal{R}_1(1^\lambda, s_1) \oplus \mathcal{R}_2(1^\lambda, s_2) \quad \text{for} \quad \vec{s} = (s_1, s_2) \in \{0,1\}^{\ell_{1,\mathrm{in}}(\lambda)} \times \{0,1\}^{\ell_{2,\mathrm{in}}(\lambda)}$$

where $\oplus$ denotes the bitwise XOR.

In order to state the result, we introduce the following auxiliary notation. For any security parameter λ, any key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$, any plaintext m, any $r_1 \in \{0,1\}^{L(\lambda)}$, and any random tape r' for Dec, we define a function $\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,r_1,r'} \colon \{0,1\}^{L(\lambda)} \to \{0,1\}$ by

$$\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,r_1,r'}(r_2) \overset{\mathrm{def}}{=} \begin{cases} 0 & \text{if } \mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m; r_1 \oplus r_2); r') = m \ , \\ 1 & \text{if } \mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m; r_1 \oplus r_2); r') \neq m \ . \end{cases}$$

Then we have the following result.

Theorem 3. *Let the PKE scheme Π and the PRG $\mathcal{R}$ be as above.*

1. *If Π is IND-CPA and $\mathcal{R}_1$ is secure against uniform distinguishers, then $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$ is also IND-CPA.*
2. *Suppose that Π is $\alpha(\lambda)$-key $\beta(\lambda)$-correct. Let $\eta(\lambda)$ be a function. Suppose moreover that $\mathcal{R}_2$ satisfies*
$$\left|\Pr\left[\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,r_1,r'}(\mathcal{R}_2(1^\lambda, U[\{0,1\}^{\ell_{2,\mathrm{in}}(\lambda)}])) = 1\right] - \Pr\left[\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,r_1,r'}(U[\{0,1\}^{L(\lambda)}]) = 1\right]\right| \le \eta(\lambda)$$
for any function $\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,r_1,r'}$ defined above. Then $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$ is $\alpha(\lambda)$-key $(\beta(\lambda) - \eta(\lambda))$-correct.

Proof. For the first assertion, let $\mathcal{A}$ be any PPT adversary for the IND-CPA game for $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$. We consider the following distinguisher $\mathcal{D}$ for the PRG $\mathcal{R}_1$: Given 1^λ and $r_1 \in \{0,1\}^{L(\lambda)}$, the distinguisher emulates the IND-CPA game for Π where the adversary is the $\mathcal{A}$ and the encryption algorithm of the challenger uses $r_1 \oplus \mathcal{R}_2(1^\lambda, s_2)$ with uniformly random seed s_2 as its random tape, and outputs 1 if the adversary wins the emulated game and otherwise outputs 0. Note that $\mathcal{D}$ is PPT. Now, when $r_1 = \mathcal{R}_1(1^\lambda, s_1)$ with uniformly random seed s_1, the emulated game is identical to the IND-CPA game for $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$, therefore we have

$$\mathsf{Adv}_{\mathcal{A}}^{\Pi \circ_{\mathrm{Enc}} \mathcal{R}}(1^\lambda) = \left|\Pr[\mathcal{D}(1^\lambda, \mathcal{R}_1(1^\lambda, U[\{0,1\}^{\ell_{1,\mathrm{in}}(\lambda)}])) = 1] - 1/2\right| \enspace .$$

On the other hand, when r_1 is uniformly random, $r_1 \oplus \mathcal{R}_2(1^\lambda, s_2)$ is also uniformly random, therefore the emulated game is identical to the IND-CPA game for Π. Hence we have

$$\mathsf{Adv}_{\mathcal{A}}^{\Pi}(1^\lambda) = \left|\Pr[\mathcal{D}(1^\lambda, U[\{0,1\}^{L(\lambda)}]) = 1] - 1/2\right| \enspace .$$

As $\mathcal{D}$ is PPT and $\mathcal{R}_1$ is secure against uniform distinguishers, $\Pr[\mathcal{D}(1^\lambda, \mathcal{R}_1(1^\lambda, U[\{0,1\}^{\ell_{1,\mathrm{in}}(\lambda)}])) = 1]$ and $\Pr[\mathcal{D}(1^\lambda, U[\{0,1\}^{L(\lambda)}]) = 1]$ have negligible difference, therefore $\mathsf{Adv}_{\mathcal{A}}^{\Pi \circ_{\mathrm{Enc}} \mathcal{R}}(1^\lambda)$ and $\mathsf{Adv}_{\mathcal{A}}^{\Pi}(1^\lambda)$ also have negligible difference. As Π is IND-CPA and $\mathcal{A}$ is PPT, it follows that $\mathsf{Adv}_{\mathcal{A}}^{\Pi}(1^\lambda)$ is negligible, so is $\mathsf{Adv}_{\mathcal{A}}^{\Pi \circ_{\mathrm{Enc}} \mathcal{R}}(1^\lambda)$. Hence $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$ is also IND-CPA.

For the second assertion, let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ be not in the set $\mathcal{K}^\dagger$ in Definition 2, and let m be any plaintext. Then, by the construction of the function $\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,r_1,r'}$, we have

$$\begin{aligned}
\mathrm{err}_{\mathrm{PRG}} &\overset{\mathrm{def}}{=} \Pr_{\vec{s} \leftarrow U[\{0,1\}^{\ell_{\mathrm{in}}(\lambda)}]}\left[\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m; \mathcal{R}(1^\lambda, \vec{s}))) \neq m\right] \\
&= \sum_{s_1 \in \{0,1\}^{\ell_{1,\mathrm{in}}(\lambda)}} 2^{-\ell_{1,\mathrm{in}}(\lambda)} \Pr_{r'}\left[\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,\mathcal{R}_1(1^\lambda,s_1),r'}(\mathcal{R}_2(1^\lambda, U[\{0,1\}^{\ell_{2,\mathrm{in}}(\lambda)}])) = 1\right] \enspace .
\end{aligned}$$

By the hypothesis, we have

$$\begin{aligned}
\mathrm{err}_{\mathrm{PRG}} &\le 2^{-\ell_{1,\mathrm{in}}(\lambda)} \sum_{s_1 \in \{0,1\}^{\ell_{1,\mathrm{in}}(\lambda)}} \left(\Pr_{r'}\left[\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,\mathcal{R}_1(1^\lambda,s_1),r'}(U[\{0,1\}^{L(\lambda)}]) = 1\right] + \eta(\lambda)\right) \\
&= \eta(\lambda) + 2^{-\ell_{1,\mathrm{in}}(\lambda)} \sum_{s_1 \in \{0,1\}^{\ell_{1,\mathrm{in}}(\lambda)}} \Pr_{r'}\left[\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,\mathcal{R}_1(1^\lambda,s_1),r'}(U[\{0,1\}^{L(\lambda)}]) = 1\right] \enspace .
\end{aligned}$$

Moreover, as $\mathcal{R}_1(1^\lambda, s_1) \oplus r_2$ is uniformly random if $r_2 \in \{0,1\}^{L(\lambda)}$ is uniformly random, it follows from the construction of the function $\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,r_1,r'}$ that, for any s_1,

$$\Pr_{r'}\left[\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,\mathcal{R}_1(1^\lambda,s_1),r'}(U[\{0,1\}^{L(\lambda)}]) = 1\right] = \Pr_{r \leftarrow U[\{0,1\}^{L(\lambda)}]}[\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m; r)) \neq m] \enspace .$$

The right-hand side is at most $1 - \beta(\lambda)$ by the hypothesis for the correctness of Π. Therefore, we have

$$\mathrm{err}_{\mathrm{PRG}} \le \eta(\lambda) + 2^{-\ell_{1,\mathrm{in}}(\lambda)} \sum_{s_1 \in \{0,1\}^{\ell_{1,\mathrm{in}}(\lambda)}} (1 - \beta(\lambda)) = 1 - (\beta(\lambda) - \eta(\lambda)) \enspace .$$

Hence $\Pi \circ_{\mathrm{Enc}} \mathcal{R}$ is $\alpha(\lambda)$-key $(\beta(\lambda) - \eta(\lambda))$-correct. This completes the proof of Theorem 3. □

Owing to the theorem, instead of developing a PRG secure against non-uniform distinguishers, our task has been reduced to develop both a PRG secure against uniform distinguishers (which is standard in cryptography) and a "special-purpose" PRG that fools the explicit family of functions $\mathcal{F}_{\lambda,\mathsf{pk},\mathsf{sk},m,r_1,r'}$ defined above. Since the complexity of those functions is bounded well and is almost the sum of the complexity of the encryption and decryption algorithms of the given PKE scheme, it might be hopeful (especially when the PKE scheme has an efficient structure) to develop such a special-purpose PRG by e.g., using techniques in the area of derandomization such as [18, 21].

References

[1] M. Ajtai, C. Dwork: A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. In: Proceedings of STOC 1997, ACM, pp.284–293, 1997.

[2] B. Auerbach, M. Bellare, E. Kiltz: Public-Key Encryption Resistant to Parameter Subversion and Its Realization from Efficiently-Embeddable Groups. In: Proceedings of PKC 2018 (Part I), LNCS vol.10769, pp.348–377, 2018.

[3] D. J. Bernstein, T. Lange: Non-Uniform Cracks in the Concrete: The Power of Free Precomputation. In: Proceedings of ASIACRYPT 2013 (Part II), LNCS vol.8270, pp.321–340, 2013.

[4] N. Bitansky, V. Vaikuntanathan: A Note on Perfect Correctness by Derandomization. In: Proceedings of EUROCRYPT 2017 (Part II), LNCS vol.10211, pp.592–606, 2017.

[5] A. De, L. Trevisan, M. Tulsiani: Time Space Tradeoffs for Attacks against One-Way Functions and PRGs. In: Proceedings of CRYPTO 2010, LNCS vol.6223, pp.649–665, 2010.

[6] Y. Deng: Magic Adversaries Versus Individual Reduction: Science Wins Either Way. In: Proceedings of EUROCRYPT 2017 (Part II), LNCS vol.10211, pp.351–377, 2017.

[7] C. Dwork, M. Naor, O. Reingold: Immunizing Encryption Schemes from Decryption Errors. In: Proceedings of EUROCRYPT 2004, LNCS vol.3027, pp.342–360, 2004.

[8] O. Goldreich: Foundations of Cryptography, Volume I. Cambridge University Press, 2001.

[9] O. Goldreich: Foundations of Cryptography, Volume II. Cambridge University Press, 2004.

[10] O. Goldreich, S. Goldwasser, S. Halevi: Eliminating Decryption Errors in the Ajtai–Dwork Cryptosystem. In: Proceedings of CRYPTO 1997, LNCS vol.1294, pp.105–111, 1997.

[11] W. Hoeffding: Probability Inequalities for Sums of Bounded Random Variables. Journal of the American Statistical Association, vol.58, no.301, pp.13–30, 1963.

[12] T. Holenstein, R. Renner: One-Way Secret-Key Agreement and Applications to Circuit Polarization and Immunization of Public-Key Encryption. In: Proceedings of CRYPTO 2005, LNCS vol.3621, pp.478–493, 2005.

[13] N. Howgrave-Graham, P. Q. Nguyen, D. Pointcheval, J. Proos, J. H. Silverman, A. Singer, W. Whyte: The Impact of Decryption Failures on the Security of NTRU Encryption. In: Proceedings of CRYPTO 2003, LNCS vol.2729, pp.226–246, 2003.

[14] Z. Huang, J. Lai, W. Chen, M. H. Au, Z. Peng, J. Li: Hedged Nonce-Based Public-Key Encryption: Adaptive Security Under Randomness Failures. In: Proceedings of PKC 2018 (Part I), LNCS vol.10769, pp.253–279, 2018.

[15] J. Katz, Y. Lindell: Introduction to Modern Cryptography, Second Edition. Taylor & Francis Group, 2015.

[16] N. Koblitz, A. Menezes: Another Look at Non-Uniformity. Groups Complexity Cryptology, vol.5, no.2, pp.117–139, 2013.

[17] H. Lin, S. Tessaro: Amplification of Chosen-Ciphertext Security. In: Proceedings of EUROCRYPT 2013, LNCS vol.7881, pp.503–519, 2013.

[18] N. Nisan, A. Wigderson: Hardness vs Randomness. Journal of Computer and System Sciences, vol.49, no.2, pp.149–167, 1994.

[19] K. Nuida: How to Use Pseudorandom Generators in Unconditional Security Settings. In: Proceedings of ProvSec 2014, LNCS vol.8782, pp.291–299, 2014.

[20] K. Pietrzak, M. Skorski: Non-Uniform Attacks Against Pseudoentropy. In: Proceedings of ICALP 2017, LIPICS vol.80, Article no.39, 2017.

[21] R. Shaltiel, C. Umans: Simple Extractors for All Min-Entropies and a New Pseudo-Random Generator. In: Proceedings of FOCS 2001, IEEE, pp.648–657, 2001.

[22] J. Zhang, Y. Yu: Two-Round PAKE from Approximate SPH and Instantiations from Lattices. In: Proceedings of ASIACRYPT 2017 (Part III), LNCS vol.10626, pp.37–67, 2017.