A NOTE ON A YAO'S THEOREM ABOUT PSEUDORANDOM GENERATORS

STÉPHANE BALLET AND ROBERT ROLLAND

Abstract. The Yao's theorem gives an equivalence between the indistinguishability of a pseudorandom generator and the impredictability of the next bit from an asymptotic point of view. We present in this paper, with detailed proofs, some modified versions of the Yao's theorem which can be of interest for the study of practical systems. We study the case of one pseudorandom generator, then the case of a family of pseudorandom generators having the same fixed length and last an asymptotical version of the previous result. We compute in each case the cost of the reduction between the two algorithms.

1. Introduction

In [4] A. Yao defines for a family of pseudorandom generators depending on a security parameter the notion of indistinguishability to be the impossibility in an asymptotical context of building a uniform (in the sense of uniform Turing machine) probabilistic polynomial time algorithm able to distinguish between these pseudorandom generators and a true random generator. Next, he defines the notion of polynomial statistical test and then defines for a source S and a statistical test M what is the meaning of the following assertion: "the source S passes the statistical test M". It turns out that this meaning is roughly speaking the impredictability of the next bit knowing the first ones. He states the asymptotical equivalence by probabilistic polynomial reduction of the indistinguishability and the impredictability of the next bit.

Let us remark that the study of provable security notions can be done from different points of view:

(1) the study can be done in a static context, with given parameters. In this case, the sizes of the objects are fixed. Namely, we deal with a non-asymptotic study;

Date: November 9, 2009.

(2) on the contrary the study can be done in a dynamic context, namely the system depends on a variable parameter (the so-called security parameter) growing to infinity. It is the case when we consider the Blum Blum Shub pseudorandom generator family based on a modulus N of size k bits where k is a variable parameter (the security parameter). On such an asymptotic study, all the data depend on the security parameter k.

In the paper [4], the study is done in an asymptotical context. Let us note that unfortunately, there exist few books of cryptography introducing Yao's theorem and in our knowing always in asymptotical formulation. Good reference works on this topic (and on many other subjects related to complexity theory in cryptography) are the books by O. Goldreich [1], [2] and the book by D. Stinson [3].

In this paper, we follow the Yao's result in order to present modified version expressed in a static context. We give detailed proofs and then we compute the exact cost of the reductions between the notion of indistinguishability and the notion of impredicability of the next bit. We stress that this point of view can be of interest for a practical study of concrete pseudorandom generators with a fixed length. Last, we derive from the previous results an asymptotical result for families of pseudorandom generators having the same security parameter k, when k is growing to infinity.

In the section 1.1, we give the typographic conventions used in this paper and the main notations. In the section 2 we define what is a pseudorandom generator and define some probabilities going with a pseudorandom generator. Next, in section 3 we introduce the security notions in particular the notion of indistinguishability, the notion of impredicability and then we prove a static version of the Yao's theorem giving in the same time the costs of the reductions between these two notions. In section 4 we generalize the results of the section 3 to a family of pseudorandom generators with fixed parameters. In section 5 we derive from section 4 a detailed proof of a slight improvement of the asymptotic Yao's theorem stated in [4], [1], [2] and [3].

1.1. **Notations.**

1.1.1. *Typography.* We will denote the integers by the letters k, l, i, s, n, m. The algorithms will be denoted by $\mathcal{A}, \mathcal{B}, \mathcal{G}$. The vectors of $\{0,1\}^m$ (where m is an integer exponent) will be denoted by X, Y. For example $X = (x_1, x_2, \cdots, x_l)$ denotes a finite bit sequence $(x_i)_i$. The bits will be denoted by x_i, y_i, b. The subsets of $\{0,1\}^m$ will be written in

bold type: $\mathbf{U}, \mathbf{Y}, \mathbf{Z}$. In particular, if $Y = (y_1, y_2, \cdots, y_l)$ is an element of $\{0,1\}^l$, then $\mathbf{Y}$ will denote the subset $\{Y\}$ constituted by the unique element Y.

1.1.2. *Algorithms.* The arrow $\leftarrow$ will denote the following operations which can be distinguished by the context:

- **assignment** of a value to a variable, for examples:
$$X \leftarrow (x_1, x_2, \cdots, x_l),$$
$$b \leftarrow 1,$$
$$b \leftarrow \mathcal{A}(y_1, y_2, \cdots, y_l);$$
- **random assignment** to a variable according to the uniform distribution, for examples:
$$Y \leftarrow \{0,1\}^l$$
(we draw at random a binary vector according to the uniform distribution),
$$b \leftarrow \{0,1\}$$
(we draw a bit at random);
- **weighted random assignment** to a variable according to a probability δ, for example:
$$f \xleftarrow{\delta} \Gamma$$
(we draw at random according to the probability δ a function in a finite family Γ).

The other used notations, concerning the algorithm running or the random experiment running, are classical and can be easily understood.

2. Pseudorandom generators

2.1. Definition of a pseudorandom generator.

Definition 2.1. *A pseudorandom generator is a deterministic function f defined on a subset $\mathbf{U} \subseteq \{0,1\}^k$ into $\{0,1\}^l$ (where $k < l$) which maps a seed $X_0 \in \mathbf{U}$ (a secret seed) to a finite sequence of l bits:*
$$f(X_0) = (x_1, x_2, \cdots, x_l).$$

Generally the function f is built using a recursive computation, which outputs successively the bits x_i of $f(X_0)$. In a typical case we have a function u from $\{0,1\}^k$ into itself which computes recursively a secret internal state X_n from the initial value X_0:
$$X_n = u(X_{n-1}),$$

and a function v which from the input X_n outputs the bit x_n (or sometimes a few bits):

$$x_n = v(X_n).$$

So, we can compute the successive bits of $f(X_0) = (x_1, \cdots, x_l)$. If the functions u and v are well designed, an attacker knowing the first bits $x_1, x_2, \cdots, x_t$ (but not the seed X_0) cannot compute in practice the bit x_{t+1}.

Exemple 2.2 (Blum Blum Shub generator $x^2 \mod n$)**.** *Let n be a Blum integer (namely a product of two primes p and q which are equal to 3 modulo 4) and having k bits (for example $k = 2048$). From a seed having 128 bits (here $\mathbf{U} = \{0,1\}^{128}$) we define the sequence $X_i = X_{i-1}^2 \mod n$, and then $x_i = lsb(X_i) = X_i \mod 2$. This pseudorandom generator is the BBS generator (Blum Blum Shub).*

2.2. **Probabilities related to a pseudorandom generator.** Let f be a pseudorandom generator. We define some probabilities related to f. Then we give simple formulae involving these probabilities. If A is a finite set, we will denote by $\#A$ its cardinality.

Let us denote by $P_{\mathbf{U}}$ the uniform probability on $\mathbf{U}$, Π_j the uniform probability on $\{0,1\}^j$ and Q_f the image probability by the map f of $P_{\mathbf{U}}$. If $\mathbf{Y} \subseteq \{0,1\}^l$ then:

$$\Pi_l(\mathbf{Y}) = \frac{\#\mathbf{Y}}{2^l} \qquad Q_f(\mathbf{Y}) = P_{\mathbf{U}}\left(f^{-1}(\mathbf{Y})\right) = \frac{\#f^{-1}(\mathbf{Y})}{\#\mathbf{U}}.$$

Now let us fix an integer s such that $0 \leq s \leq l$. We want to build at random an element $(y_1, \cdots y_l) \in \{0,1\}^l$ in the following way:

Construction $(C_{f,s})$ **:**

(1) we draw at random $X_0 \in \mathbf{U}$ according to the uniform distribution on $\mathbf{U}$;
(2) we compute $f(X_0) = (x_1, \cdots, x_l)$ and we keep the s first bits $(y_1, \cdots, y_s)$ (where $y_1 = x_1, \cdots, y_s = x_s$)
(3) we complete these sequence of s bits by $l - s$ bits $(y_{s+1}, \cdots, y_l)$ taken at random in $\{0,1\}^{l-s}$ according to the uniform distribution.

We introduce a probability adapted to this construction, namely the probability to obtain an $Y = (y_1, y_2, \cdots, y_l)$ as output of the construction $(C_{f,s})$.

For any integer s such that $0 \leq s \leq l$ we define over $\{0,1\}^l$ the following probability $p_{f,s}$ by

$$p_{f,s}\Big(\{(y_1, y_2, \cdots, y_l)\}\Big) =$$

$$P_{\mathbf{U}}\Big(f^{-1}\left(\{(y_1, y_2, \cdots, y_s)\} \times \{0,1\}^{l-s}\right)\Big) \times \Pi_{l-s}\Big(\{(y_{s+1}, \cdots, y_l)\}\Big),$$

It follows from the definition of $P_{\mathbf{U}}$, Π_j et Q_f that:

$$(1)\quad p_{f,s}\Big(\{(y_1, y_2, \cdots, y_l)\}\Big) = \frac{1}{2^{l-s}} Q_f\Big(\{(y_1, y_2, \cdots, y_s)\} \times \{0,1\}^{l-s}\Big).$$

To simplify let us denote by $\mathbf{Y}$ the event

$$\mathbf{Y} = \{Y\} = \{(y_1, y_2, \cdots, y_l)\},$$

by $\mathbf{Y}_s$ the event "the s first components are $y_1 \cdots y_s$", namely

$$\mathbf{Y}_s = \{(y_1, y_2, \cdots, y_s)\} \times \{0,1\}^{l-s},$$

and by $\mathbf{Z}_{s+1}$ the event "the component of index $s+1$ is y_{s+1}", namely

$$\mathbf{Z}_{s+1} = \{0,1\}^s \times \{y_{s+1}\} \times \{0,1\}^{l-s-1}.$$

The formula (1), can be written

$$(2)\qquad p_{f,s}(\mathbf{Y}) = \frac{1}{2^{l-s}} Q_f(\mathbf{Y}_s).$$

From the definition of a conditionnal probability and from the equality

$$\mathbf{Y}_s \cap \mathbf{Z}_{s+1} = \mathbf{Y}_{s+1},$$

it follows

$$Q_f(\mathbf{Y}_s) \times Q_f(\mathbf{Z}_{s+1} | \mathbf{Y}_s) = Q_f(\mathbf{Y}_{s+1}),$$

and then using the formula (1) (or the formula (2)):

$$(3)\qquad p_{f,s}(\mathbf{Y}) \times Q_f(\mathbf{Z}_{s+1} | \mathbf{Y}_s) = \frac{1}{2}\, p_{f,s+1}(\mathbf{Y}),$$

namely, with the previous notations:

(4)

$$p_{f,s}\Big(\{(y_1, y_2, \cdots, y_l)\}\Big) \times$$

$$Q_f\Big(\{0,1\}^s \times \{y_{s+1}\} \times \{0,1\}^{l-s-1} | \{(y_1, y_2, \cdots, y_s)\} \times \{0,1\}^{l-s}\Big) =$$

$$\frac{1}{2}\, p_{f,s+1}\Big(\{(y_1, y_2, \cdots, y_l)\}\Big).$$

Remark 2.3. *For $s = 0$ we obtain $p_{f,0} = \Pi_l$ (all the bits are drawn according to the uniform distribution). For $s = l$ we obtain $p_{f,l} = Q_f$ (all the bits are computed with the pseudorandom generator).*

3. The security of a pseudorandom generator

3.1. **Definition of a secure pseudorandom generator.** Let us consider the following pseudorandom generator:

$$f : \mathbf{U} \subset \{0,1\}^k \to \{0,1\}^l \text{ where } k < l.$$

Let us recall that a probabilistic algorithm can be seen as a non-deterministic algorithm having for each input a probability on the set of the runs which can occur when we start from this input.

If $\mathcal{A}$ is a probabilistic algorithm which outputs one bit, We will denote by $\mu_{\mathcal{A}}(e)$ the probability of the output 1 when the input of $\mathcal{A}$ is e.

The following random experiment, retated to the construction $(C_{f,s})$ defined in the paragraph 2.2, involves a probabilistic algorithm $\mathcal{A}$ having for input a vector $Y \in \{0,1\}^l$ and which output one bit. Roughly speaking, this algorithm tries to distinguish the given pseudorandom generator f from a true random one. More precisely, it has for aim to recognize if an input Y comes from the pseudorandom generator f or for a true random generator.

Let us fix an integer s such that $0 \leq s \leq l$.

$\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{dist}}(\mathcal{A})$
  $X_0 \leftarrow \mathbf{U} \subseteq \{0,1\}^k$
  $X \leftarrow f(X_0)$
    (notation : $X = (x_1, \cdots, x_l)$)
  $Y_1 \leftarrow (x_1, x_2, \cdots, x_s)$
  $Y_2 \leftarrow \{0,1\}^{l-s}$
  $Y \leftarrow Y_1 || Y_2$
  $b \leftarrow \mathcal{A}(Y)$
  return b
End.

Let $q_{f,s}$ be the probability that the experiment

$$\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{dist}}(\mathcal{A})$$

returns $b = 1$. With the previous notations we have the following:

$$q_{f,s} = \sum_{Y \in \{0,1\}^l} p_{f,s}(\mathbf{Y}) \mu_{\mathcal{A}}(Y). \tag{5}$$

In particular, $q_{f,0}$ is the probability of the following event: we draw at random an element of $\{0,1\}^l$ according to the uniform distribution, we run the algorithm $\mathcal{A}$ on this element, and the output is 1. The probability $q_{f,l}$ is the probability of the following event: we draw at

random a seed X_0 in $\mathbf{U}$, we apply f to obtain an element of $\{0,1\}^l$ which becomes the input of the algorithm $\mathcal{A}$, and the output is 1.

Let us recall now the notion of advantage which permits to quantify the ability of $\mathcal{A}$ to distinguish f.

Definition 3.1. *The advantage of the algorithm $\mathcal{A}$ to distinguish f is*

$$Adv_f^{dist}(\mathcal{A}) = |q_{f,l} - q_{f,0}|.$$

Then we define a (T,ϵ)-distinguisher:

Definition 3.2. *Let f be a pseudorandom generator. Let T and ϵ be positive real numbers. A (T,ϵ)-distinguisher for f is a probabilistic algorithm $\mathcal{A}$ such that*

(1) the maximal running time of $\mathcal{A}$ is $\leq T$,
(2) the input of $\mathcal{A}$ is an element of $\{0,1\}^l$,
(3) the output of $\mathcal{A}$ is a bit b,
(4) the algorithm $\mathcal{A}$ can distinguish the pseudorandom generator from the uniform distribution, namely

$$Adv_f^{dist}(\mathcal{A}) > \epsilon.$$

We can now define the (T,ϵ)-security of f.

Definition 3.3. *The generator f is (T,ϵ)-secure, if it does not exist any (T,ϵ)-distinguisher for f, namely any probabilistic algorithm $\mathcal{A}$ with maximal running time $t(\mathcal{A}) \leq T$ has an advantage satisfying the inequality*

$$Adv_f^{dist}(\mathcal{A}) \leq \epsilon.$$

Remark 3.4. *In the advantage definition we can suppose that $q_{f,l} \geq q_{f,0}$, if not we can replace $\mathcal{A}$ by the complementary algorithm (which outputs 1 when the other ouputs 0 and vice versa). Using this remark we can avoid to use absolute value.*

3.2. **Impredictability of a pseudorandom generator.** Let us consider the following pseudorandom generator:

$$f : \mathbf{U} \subset \{0,1\}^k \to \{0,1\}^l.$$

Let $1 \leq s < l$. The following random experiment involves a probabilistic algorithm $\mathcal{B}$ having for input a sequence of s bits and for output a bit. Roughly speaking, this algorithm tries to predict the next bit produced by the pseudorandom generator f, namely the bit of index $s+1$.

$\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{pred}}(\mathcal{B})$
$X_0 \leftarrow \mathbf{U} \subseteq \{0,1\}^k$

$X \leftarrow f(X_0)$
  (notation : $X = (x_1, \cdots, x_l)$)
$Y \leftarrow (x_1, x_2, \cdots, x_s)$
$b \leftarrow \mathcal{B}(Y)$
if $b = x_{s+1}$
  then return 1
  else return 0
fi
End.

Let $r_{f,s}$ be the probability that the experiment $\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{pred}}(\mathcal{B})$ returns 1. With the previous notations:

$$r_{f,s} = \sum_{Y \in \{0,1\}^l, y_{s+1}=1} p_{f,s}(\mathbf{Y}) \mu_{\mathcal{B}}(Y) + \sum_{Y \in \{0,1\}^l, y_{s+1}=0} p_{f,s}(\mathbf{Y}) \left(1 - \mu_{\mathcal{B}}(Y)\right).$$

Definition 3.5. *The advantage of the algorithm $\mathcal{B}$ to predict the bit of index $(s+1)$ computed by f is:*

$$Adv_{f,s}^{pred}(\mathcal{B}) = \left| r_{f,s} - \frac{1}{2} \right|.$$

We can now define the notion of (T, s, ϵ)-prediction algorithm.

Definition 3.6. *Let f be a pseudorandom generator. Let T and ϵ be positive real numbers and s be an integer such that $1 \leq s < l$. A (T, s, ϵ)-prediction algorithm $\mathcal{B}$ is a probabilistic algorithm such that:*

(1) the maximal running time of $\mathcal{B}$ is $\leq T$,
(2) the input of $\mathcal{B}$ is an element of $\{0,1\}^s$,
(3) the output of $\mathcal{B}$ is a bit,
(4) the algorithm $\mathcal{B}$ can predict the next bit, namely

$$Adv_{f,s}^{pred}(\mathcal{B}) > \epsilon.$$

We define now the notion of (T, s, ϵ)-impredictable pseudorandom generator.

Definition 3.7. *Let f be a pseudorandom genrator. Let and s an integer such that $1 \leq s < l$. The generator f is (T, s, ϵ)-impredictable, if there does not exist any (T, s, ϵ)-prediction algorithm.*

3.3. **Yao's theorem, static version.** The Yao's theorem relates the notion of security to the notion of impredictability of the next bit. We express it in its non-asymptotic form. In this case, we give two results which can be considered respectively as a necessary condition and a sufficient condition to have the security of a generator f.

Theorem 3.8. *We consider the following pseudorandom generator:*

$$f : \mathbf{U} \subset \{0,1\}^k \to \{0,1\}^l.$$

If we have a

$$(T, s, \epsilon)\text{-}\textit{prediction algorithm}$$

for f, we can build a

$$(T + c, \epsilon)\text{-}\textit{distinguisher}$$

where c is the constant time needed to compare two bits.

Proof. Let $\mathcal{B}$ be a (T, s, ϵ)-prediction algorithm. We build a (T, ϵ)-distinguisher $\mathcal{A}$ in the following way:

$\mathcal{A}(x_1, x_2, \cdots x_l)$
  $b \leftarrow \mathcal{B}(x_1, x_2, \cdots x_s)$
  if $b = x_{s+1}$
    the return 1
    else return 0
  fi
End.

The probability to have $\mathcal{A}(f(X_0)) = 1$ is then $> 1/2 + \epsilon$ since $\mathcal{B}$ is a (T, s, ϵ)-prediction algorithm. But for a random

$$(y_1, y_2, \cdots, y_l) \in \{0,1\}^l,$$

the probability to have

$$\mathcal{A}(y_1, y_2, \cdots, y_l) = 1$$

is $1/2$. Moreover, to obtain the running time of the built distinguisher we just add to the running time of $\mathcal{B}$ the constant time c needed to compare b to x_{s+1} (to compare two bits). □

Theorem 3.9. *Let f be a pseudorandom generator:*

$$f : \mathbf{U} \subset \{0,1\}^k \to \{0,1\}^l.$$

Let us suppose that for all s such that $1 \leq s < l$, it does not exist any (T, s, ϵ)-prediction algorithm. Then f is $(T - (c_1 l + c_2), l\,\epsilon)$-secure where c_1 is the constant time needed to draw one bit at random, and c_2 is the constant time needed to test the value of a bit and then depending upon the value of this bit to return a bit or its complementary.

Proof. Let us suppose that f is not (T_1, η)-secure. Then there is a distinguisher algorithm $\mathcal{A}$ which has an running time $\leq T_1$ and an advantage $> \eta$. Let us consider the construction $(C_{f,s})$ defined in the

paragraph 2.2, and let us use the probabilities introduced in the paragraph 3. Even if it means changing the algorithm $\mathcal{A}$ by its complementary, we can suppose that $q_{f,l} - q_{f,0} > \eta$. Hence:

$$Adv_f^{dist}(\mathcal{A}) = q_{f,l} - q_{f,0} =$$
$$(q_{f,l} - q_{f,l-1}) + (q_{f,l-1} - q_{f,l-2}) + \cdots$$
$$+(q_{f,s} - q_{f,s-1}) + \cdots + (q_{f,1} - q_{f,0}) > \eta.$$

Then, there is an integer s such that $|q_{s+1} - q_s| > \eta/l$. Now, let us define the following algorithm $\mathcal{B}$:

$\mathcal{B}(z_1, z_2, \cdots z_s)$
  $(z_{s+1}, \cdots, z_l) \leftarrow \{0,1\}^{l-s}$
  $b \leftarrow \mathcal{A}(z_1, \cdots, z_l)$
  if $b = 1$
    then return z_{s+1}
    else return $\overline{z_{s+1}}$
  fi
End.

The running time of this algorithm is less than $T_1 + c_1 l + c_2$, where c_1 is the constant time needed to draw at random 1 bit, and c_2 the constant time needed to return z_s or $\overline{z_s}$ according to b. Let us prove now that the algorithm $\mathcal{B}$ is a $(T_1 + c_1 l + c_2, s, \eta/l)$-prediction algorithm. First, let us compute the probability $r_{f,s}$ such that the result of the experiment$\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{pred}}(\mathcal{B})$ is 1. To do that we nest the definition of $\mathcal{B}$ in the definition of the experiment $\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{pred}}(\mathcal{B})$.

We obtain the following experiment:

$\mathbf{Expt}_{\mathbf{f,s}}^{\mathbf{pred}}(\mathcal{B})$
  $X_0 \leftarrow \mathbf{U} \subset \{0,1\}^k$
  $Y \leftarrow f(X_0)$
    (notation : $Y = (x_1, \cdots, x_l)$)
  $Y_s \leftarrow (x_1, x_2, \cdots, x_s)$
  (Y_s is the input of $\mathcal{B}$,
    which only knows these componants)
  •**begin nesting of** $\mathcal{B}$
  $(z_{s+1}, \cdots, z_l) \leftarrow \{0,1\}^{l-s}$
  $b_1 \leftarrow \mathcal{A}(x_1, \cdots, x_s, z_{s+1}, \cdots, z_l)$
  if $b_1 = 1$
    then $b \leftarrow z_{s+1}$
    else $b \leftarrow \overline{z_{s+1}}$
  •**end nesting**

if $b = x_{s+1}$
then return 1
else return 0
fi
End.

This experiment will give us a mean to compute the probability $r_{f,s}$. We remark that the result of the experiment is 1 when $b = x_{s+1}$, namely in the two following cases:

(1) $b_1 = 1$ et $z_{z+1} = x_{s+1}$;
(2) $b_1 = 0$ et $\overline{z_{s+1}} = x_{s+1}$.

Let us use the simple notations yet introduced in the paragraph 2.2: $\mathbf{Y}$ is the event $\{(x_1, \cdots, x_l)\}$, $\mathbf{Y}_s$ denotes the event "the s first components are $x_1, \cdots x_s$", $\mathbf{Z}_{s+1}$ is the event "the component $s+1$ is z_{s+1}". Let us set $\nu_{\mathcal{A}}(Y) = 1 - \mu_{\mathcal{A}}(Y)$.

Then, $Q_f(\mathbf{Z}_{s+1}|\mathbf{Y}_s)$ is the conditionnal probability, when Y is built from a random seed using the pseudorandom generator, that the component $s+1$ of Y (namely x_{s+1}) is z_{s+1}, assuming that the s first componants are $(x_1, \cdots, x_s)$.

Hence:

$$r_{f,s} = \sum_{Y \in \{0,1\}^l} p_{f,s}(\mathbf{Y}) \Big(Q_f(\mathbf{Z}_{s+1}|\mathbf{Y}_s)\, \mu_{\mathcal{A}}(Y) + Q_f(\overline{\mathbf{Z}_{s+1}}|\mathbf{Y}_s)\, \nu_{\mathcal{A}}(Y) \Big) =$$

$$\sum_{Y \in \{0,1\}^l} p_{f,s}(\mathbf{Y}) \Big(Q_f(\mathbf{Z}_{s+1}|\mathbf{Y}_s)\, \mu_{\mathcal{A}}(Y) + (1 - Q_f(\mathbf{Z}_{s+1}|\mathbf{Y}_s))\, \nu_{\mathcal{A}}(Y) \Big).$$

Using the formula (4) we get:

$$r_{f,s} = \frac{1}{2} \sum_{Y \in \{0,1\}^l} p_{f,s+1}(\mathbf{Y}) \Big(\mu_{\mathcal{A}}(Y) - \nu_{\mathcal{A}}(Y) \Big) + \sum_{Y \in \{0,1\}^l} p_{f,s}(\mathbf{Y}) \nu_{\mathcal{A}}(Y) =$$

$$\frac{1}{2} \sum_{Y \in \{0,1\}^l} p_{f,s+1}(\mathbf{Y}) \Big(2\mu_{\mathcal{A}}(Y) - 1 \Big) + \sum_{Y \in \{0,1\}^l} p_{f,s}(\mathbf{Y})(1 - \mu_{\mathcal{A}}(Y)) =$$

$$\frac{1}{2} + \sum_{Y \in \{0,1\}^l} \Big(p_{f,s+1}(\mathbf{Y}) - p_{f,s}(\mathbf{Y}) \Big) \mu_{\mathcal{A}}(Y).$$

This equality and the use of the formula (5) give the following:

$$r_{f,s} = \frac{1}{2} + q_{f,s+1} - q_{f,s},$$

hence:

$$\left| r_{f,s} - \frac{1}{2} \right| > \frac{\eta}{l}.$$

Now we get the result by setting $T_1 = T - (c_1 l + c_2)$ and $\eta = l\epsilon$. □

Remark 3.10. Changing the direction of the prediction algorithm. *In the paragraph 3.2 we defined and used right prediction algorithms, namely, given the bits $(x_1, \cdots, x_s)$ the prediction algorithm computes the bit x_{s+1} (prediction of the next bit). In fact the same study, with the same results, can be done for left prediction algorithms, namely, for an algorithm which, given the bits $(x_{s+1}, \cdots, x_l)$, computes the bit x_s (prediction of the previous bit). In particular all the versions of Yao's theorem remain valid for left prediction algorithms.*

Remark 3.11. *Let $f : \{0,1\}^k \to \{0,1\}^l$ be a pseudorandom generator and s be an integer such that $1 \leq s < l-1$. In many practical examples we can say that if s' is an integer such that $s \leq s' < l$ then*

$$Adv^{pred}_{f,s'}(\mathcal{B}) \geq Adv^{pred}_{f,s}(\mathcal{B}).$$

For example let us consider the typical construction given in Subsection 2.1. Let u be a bijective function from $\{0,1\}^k$ onto itsel. The function u computes recursively a secret internal state X_n from the initial value X_0:

$$X_n = u(X_{n-1}).$$

Now a function v maps X_n to a bit x_n, then

$$f(X_0) = \left(v \circ u(X_0), v \circ u^2(X_0), \cdots, v \circ u^l(X_0) \right).$$

Suppose that $s' = s+1 < l$ and that we know the bits $(x'_1, x'_2, \cdots, x'_s)$ of $f(X'0)$. Then to compute the bit of index $s'+1$, we can forget the bit x'_1 and use an algorithm which knowing s bits, try to find the bit of index $s+1$. More precisely, let $X_0 = u(X'_0) = X_1$. Starting from the seed X_0 we can compute the s first terms of the pseudoandom sequence:

$$x_1 = v \circ u(X_0) = x'_2, \cdots, x_s = v \circ u^{s+1} = x'_{s'}.$$

As u is bijective, the probability repartition of X_0 is the same as the probability repartition of X'_0. Then

$$Adv^{pred}_{f,s+1}(\mathcal{B}) \geq Adv^{pred}_{f,s}(\mathcal{B}).$$

4. The security of a family of pseudo-random generators with same given size

We have considered the case of one pseudorandom generator f. But even in the non-asymptotic case where k and l are fixed, we have to study not only one, but a family (a finite family because k and l are fixed) Γ of function f defined on a subset $\mathbf{U}_f$ (which can depend on f) of $\{0,1\}^k$ with images in $\{0,1\}^l$. It is the case for the Blum Blum Shub algorithms: given the size of the modulus, we can consider all the possible modulus N having this size. Then we study algorithms which attack all the generators of the family.

4.1. Revisiting the previous notions in the case of a family of pseudo-random generators with same size.

In a realistic situation we must, in the random experiment which defines the attacker's advantage, draw at random the function f in the family Γ according to a probability δ.

So, we replace now the algorithms $\mathcal{A}$ and $\mathcal{B}$ of the previous section by algorithms whose inputs are a function $f \in \Gamma$ and an a vector. The random experiments $\mathbf{Expt}^{\mathbf{dist}}_{\mathbf{f,s}}(\mathcal{A})$ and $\mathbf{Expt}^{\mathbf{pred}}_{\mathbf{f,s}}(\mathcal{B})$ are replaced by the random experiments $\mathbf{Expt}^{\mathbf{dist}}_{\mathbf{\Gamma,s}}(\mathcal{A})$ and $\mathbf{Expt}^{\mathbf{pred}}_{\mathbf{\Gamma,s}}(\mathcal{B})$ where we draw at random not only the seed X_0, but also the function f itself.

The experiment $\mathbf{Expt}^{\mathbf{dist}}_{\mathbf{\Gamma,s}}(\mathcal{A})$ is given by the following scheme:

$\mathbf{Expt}^{\mathbf{dist}}_{\mathbf{\Gamma,s}}(\mathcal{A})$
  $f \overset{\delta}{\leftarrow} \Gamma$
  $X_0 \leftarrow \mathbf{U}_f \subseteq \{0,1\}^k$
  $X \leftarrow f(X_0)$
    (notation : $X = (x_1, \cdots, x_l)$)
  $Y_1 \leftarrow (x_1, x_2, \cdots, x_s)$
  $Y_2 \leftarrow \{0,1\}^{l-s}$
  $Y \leftarrow Y_1 || Y_2$
  $b \leftarrow \mathcal{A}(f, Y)$
  return b
End.

The probability q_s that the result of this experiment is 1 is

$$q_s = \sum_{f \in \Gamma} \delta(f) q_{f,s}.$$

The experiment $\mathbf{Expt}^{\mathbf{pred}}_{\mathbf{\Gamma,s}}(\mathcal{B})$ is given by the following scheme:

$\mathbf{Expt}^{\mathbf{pred}}_{\mathbf{\Gamma,s}}(\mathcal{B})$
$f \overset{\delta}{\leftarrow} \Gamma$
$X_0 \leftarrow \mathbf{U}_f \subseteq \{0,1\}^k$
$X \leftarrow f(X_0)$
(notation : $X = (x_1, \cdots, x_l)$)
$Y \leftarrow (x_1, x_2, \cdots, x_s)$
$b \leftarrow \mathcal{B}(f, Y)$
if $b = x_{s+1}$
then return 1
else return 0
fi
End.

The probability r_s that the result of this experiment is 1 is

$$r_s = \sum_{f \in \Gamma} \delta(f) r_{s,f}.$$

All the definitions of the advantages of the previous paragraph can be extended to this case, and the static Yao's theorems can be generalized. More precisely we can modify the definitions 3.1, 3.2, 3.3 and 3.5, 3.6, 3.7 in the following way:

Definition 4.1. *Let $\mathcal{A}$ be an algorithm having for inputs a pseudorandom generator $f \in \Gamma$ and a vector $Y \in \{0,1\}^l$ and for output a bit b. The advantage of the algorithm $\mathcal{A}$ to distinguish an element of the Γ family is:*

$$Adv^{dist}_{\Gamma}(\mathcal{A}) = |q_l - q_0|.$$

Definition 4.2. *Let Γ be a family of pseudorandom generators having the same size (i.e. the same parameters k and l). Let T and ϵ be positive real numbers. A (T, ϵ)-distinguisher for Γ is a probabilistic algorithm $\mathcal{A}$ such that:*

(1) the maximal running time of $\mathcal{A}$ is $\leq T$,
(2) the inputs of $\mathcal{A}$ are an element $f \in \Gamma$ and an element $Y \in \{0,1\}^l$,
(3) the output of $\mathcal{A}$ is a bit b,
(4) the algorithm $\mathcal{A}$ can distinguish the pseudorandom generator in Γ from the uniform distribution, namely

$$Adv^{dist}_{\Gamma}(\mathcal{A}) > \epsilon.$$

Definition 4.3. *The family Γ of pseudorandom generators (having the same size) is (T, ϵ)-secure, if it does not exist any (T, ϵ)-distinguisher for Γ.*

Definition 4.4. *Let s be an integer such that $1 \leq s < l$. Let $\mathcal{B}$ be an algorithm having for inputs a pseudorandom generator $f \in \Gamma$ and an element $Z \in \{0,1\}^s$.*

The advantage of the algorithm $\mathcal{B}$ to predict the bit of index $(s+1)$ computed by a random $f \in \Gamma$ is

$$Adv^{pred}_{\Gamma,s}(\mathcal{B}) = \left| r_s - \frac{1}{2} \right|.$$

Definition 4.5. *Let Γ be a family of pseudorandom generators (having the same size). Let T and ϵ be positive real numbers and s be an integer such that $1 \leq s < l$. A (T, s, ϵ)-prediction algorithm $\mathcal{B}$ is a probabilistic algorithm such that:*

(1) the maximal running time of $\mathcal{B}$ is $\leq T$,
(2) the inputs of $\mathcal{B}$ are an element $f \in \Gamma$ and an element $Z \in \{0,1\}^s$,
(3) the output of $\mathcal{B}$ is a bit,
(4) the algorithm $\mathcal{B}$ can predict the next bit, namely

$$Adv^{pred}_{\Gamma,s}(\mathcal{B}) > \epsilon.$$

Definition 4.6. *Let Γ be a family of pseudorandom generators (having the same size). Let s an integer such that $1 \leq s < l$. The family Γ is (T, s, ϵ)-impredictable, if there does not exist any (T, s, ϵ)-prediction algorithm.*

4.2. **Yao's theorem.**

Theorem 4.7. *Let Γ be a family of pseudorandom generators having the same size where each $f \in \Gamma$ is a function*

$$f : \mathbf{U}_f \subset \{0,1\}^k \to \{0,1\}^l.$$

If we have a

$$(T, s, \epsilon)\text{-prediction algorithm}$$

for f, we can build a

$$(T + c, \epsilon)\text{-distinguisher}$$

where c is the constant time needed to compare two bits.

Proof. The proof is similar to the proof of Theorem 3.8. Let $\mathcal{B}$ be a (T, s, ϵ)-prediction algorithm. We build a (T, ϵ)-distinguisher $\mathcal{A}$ in the following way:

$\mathcal{A}(f, x_1, x_2, \cdots x_l)$
  $b \leftarrow \mathcal{B}(f, x_1, x_2, \cdots x_s)$
  if $b = x_{s+1}$

the return 1
else return 0
fi
End.

The probability to have $\mathcal{A}(f, f(X_0)) = 1$ is then $> 1/2 + \epsilon$ since $\mathcal{B}$ is a (T, s, ϵ)-prediction algorithm. But for a random

$$(y_1, y_2, \cdots, y_l) \in \{0,1\}^l,$$

the probability to have

$$\mathcal{A}(f, y_1, y_2, \cdots, y_l) = 1$$

is $1/2$. Moreover, to obtain the running time of the built distinguisher we just add to the running time of $\mathcal{B}$, the constant time c needed to compare b to x_{s+1} (to compare two bits). □

Theorem 4.8. *Let Γ be a family of pseudorandom generators having the same size where each $f \in \Gamma$ is a function*

$$f : \mathbf{U}_f \subset \{0,1\}^k \to \{0,1\}^l.$$

Let us suppose that for all s such that $1 \leq s < l$, it does not exist any (T, s, ϵ)-prediction algorithm. Then f is $(T - (c_1 l + c_2), l\,\epsilon)$-secure where c_1 is the constant time needed to draw one bit at random, and c_2 is the constant time needed to test the value of a bit and then depending upon the value of this bit to return a bit or its complementary.

Proof. Let us suppose that Γ is not (T_1, η)-secure. Then there is a distinguisher algorithm $\mathcal{A}$ which has an running time $\leq T_1$ and an advantage $> \eta$, namely

$$q_l - q_0 = \sum_{f \in \Gamma} \delta(f)(q_{f,l} - q_{f,0}) > \eta.$$

But

$$q_{f,l} - q_{f,0} = \sum_{s=0}^{l-1} (q_{f,s+1} - q_{f,s}),$$

hence

$$q_l - q_0 = \sum_{f \in \Gamma} \delta(f) \sum_{s=0}^{l-1} (q_{f,s+1} - q_{f,s}) = \sum_{s=0}^{l-1} \sum_{f \in \Gamma} \delta(f)(q_{f,s+1} - q_{f,s}) > \eta.$$

Then, there is an integer s such that

$$\left| \sum_{f \in \Gamma} \delta(f)(q_{f,s+1} - q_{f,s}) \right| > \eta / l.$$

But we have shown in the proof of Theorem 3.9 that

$$q_{f,s+1} - q_{f,s} = r_{f,s} - \frac{1}{2},$$

hence

$$\left|\sum_{f\in\Gamma} \delta(f)(q_{f,s+1} - q_{f,s})\right| = \left|\sum_{f\in\Gamma} \delta(f)\left(r_{f,s} - \frac{1}{2}\right)\right| = \left|r_s - \frac{1}{2}\right| > \eta/l.$$

We can conclude as in the proof of Theorem 3.9. □

5. Asymptotic behaviour

As a consequence of the previous results for fixed k and l, we can deduce results on the asymptotical theory of the pseudorandom generators, namely k growing to infinity and $l = l(k) > k$ a polynomial function of k (cf. [2, Chapter 3]).

Let k be a positive integer (the security parameter) and $l(k)$ a polynomial function of k such that $l(k) > k$. For any k we have a set Γ_k of deterministic functions such that

(1) if $f \in \Gamma_k$ then f is a function from a subset $\mathbf{U}_f$ of $\{0,1\}^k$ into $\{0,1\}^{l(k)}$;
(2) there exist a polynomial function $t(k)$ such that for any k, any $f \in \Gamma_k$ and any $X \in U_f$ the computation time of $f(X)$ is upper-bounded by $t(k)$;
(3) for any k we provide a probability δ_k on the set Γ_k.

The asymptotic notions of indistinguishability and impredictability are derived respectively from the definitions 4.3 and 4.6. We define now a distinguisher $\mathcal{A}$ to be a probabilistic polynomial algorithm having for inputs the security parameter k, a function $f \in \Gamma_k$ and a vector $Y \in \{0,1\}^{l(k)}$, and which outputs a bit. Let k be an integer, we will denote by $\mathcal{A}_k$ the probabilistic algorithm obtained from $\mathcal{A}$ by fixing the first entry to the value k.

Definition 5.1. *The family $\Gamma = (\Gamma_k)_{k>0}$ of sets of pseudorandom generators is said asymptotically secure if for any polynomial $S(k)$, any integer u and any distinghuisher $\mathcal{A}$ with running time $\leq S(k)$, the advantage of the algorithm $\mathcal{A}_k$ (cf. Definition 4.1) is a negligible function of $\frac{1}{k^u}$, namely*

$$\lim_{k\to+\infty} k^u \, Adv^{dist}_{\Gamma_k}(\mathcal{A}_k) = 0.$$

Let $s = (s_k)_{k\geq 1}$ a sequence of integers such that $1 \leq s_k < l(k)$. We define now a s-prediction algorithm to be a probabilistic polynomial algorithm $\mathcal{B}$ having for inputs the security parameter k, a function

$f \in \Gamma_k$ and a vector $Z \in \{0,1\}^{s_k}$, and which outputs a bit. Let k be an integer, we will denote by $\mathcal{B}_k$ the probabilistic algorithm obtained from $\mathcal{B}$ by fixing the first entry to the value k.

Definition 5.2. *The family $\Gamma = (\Gamma_k)_{k>0}$ of sets of pseudorandom generators is said asymptotically impredictable if for any polynomial $S(k)$, any sequence s and any s-prediction algorithm $\mathcal{B}$ with running time $\leq S(k)$, the advantage of the s_k-prediction algorithm $\mathcal{B}_k$ (cf. Definition 4.4) is a negligible function of $\frac{1}{k^u}$, namely*

$$\lim_{k \to +\infty} k^u \, Adv^{pred}_{\Gamma_k, s_k}(\mathcal{A}_k) = 0.$$

The two notions are related by the following theorem:

Theorem 5.3. *Let $l(k)$ be a polynomial function of one integer variable k such that $l(l) > k$. Let $\Gamma = (\Gamma_k)_{k>0}$ a family of sets, where any set Γ_k is a probabilized set of random generators mapping a subset of $\{0,1\}^k$ into $\{0,1\}^{l(k)}$ (more precisely, each $f \in \Gamma_k$ has its own definition subset $\mathbf{U}_f \subseteq \{0,1\}^k$). The family Γ is asymptotically secure if and only if it is asymptotically impredictable.*

Proof. Let Γ be an asymptotically secure family. Suppose that Γ is not asymptotically impredictable, then there exist a polynomial function $S(k)$, an integer u and a s-prediction algorithm $\mathcal{B}$ such that $k^u \, Adv^{pred}_{\Gamma_k, s_k}(\mathcal{B}_k)$ doest not tend to 0. Then one can find $\epsilon > 0$, a sequence $(k_n)_n$ of integers and a sequence $(s_{k_n})_n$ of integers such that

$$k_n^u \, Adv^{pred}_{\Gamma_{k_n}, s_{k_n}}(\mathcal{B}_{k_n}) > \epsilon.$$

Let $\mathcal{A}_{k_n}$ be the distinguisher algorithm built in the proof of Theorem 4.7. The running time of $\mathcal{A}_{k_n}$ is $\leq S(k) + c$ (where c is a constant) and

$$k_n^u \, Adv^{dist}_{\Gamma_{k_n}}(\mathcal{A}_{k_n}) > \epsilon.$$

So we obtain a contradiction.

Now suppose that Γ is an asymptotically impredictable family. Suppose that Γ is not asymptotically secure, then there exist a polynomial $S(k)$, an integer u and a distinguisher algorithm $\mathcal{A}$ such that $k^u \, Adv^{dist}_{\Gamma_k}(\mathcal{A}_k)$ doest not tend to 0. Then one can find $\epsilon > 0$ and a sequence $(k_n)_n$ of integer such that

$$k_n^u \, Adv^{dist}_{\Gamma_{k_n}}(\mathcal{A}_{k_n}) > \epsilon.$$

Let $\mathcal{B}_{k_n}$ be the s_{k_n}-prediction algorithm built in the proof of Theorem 4.8. The running time of $\mathcal{B}_{k_n}$ is $\leq S(k) + c_1 l(k) + c_2$ (where c_1 and c_2

are two constants) and

$$k_n^u \, Adv^{pred}_{\Gamma_{k_n}, s_{k_n}}(\mathcal{B}_{k_n}) > \frac{\epsilon}{l(k)},$$

and as $l(k)$ is a polynomial function, there is an integer v such that

$$k_n^v \, Adv^{pred}_{\Gamma_{k_n}, s_{k_n}}(\mathcal{B}_{k_n}) > \epsilon.$$

So we obtain a contradiction. □

References

[1] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Number 17 in Algorithms and Combinatorics. Springer, 1999.

[2] Oded Goldreich. *The Foundations of Cryptography, Volume I*. Cambridge University Press, 2001.

[3] Douglas Stinson. *Cryptography: Theory and Practice, Third Edition*. CRC Press, 2005.

[4] Andrew C. Yao. Theory and Applications of Trapdoor Functions. In *Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science*, pages 80–91. IEEE Computer Society, 1982.

Institut de Mathématiques de Luminy, case 930, F13288 Marseille cedex 9, France

E-mail address: ballet@iml.univ-mrs.fr

Institut de Mathématiques de Luminy, Campus de Luminy, Case 907, 13288 MARSEILLE Cedex 9

E-mail address: robert.rolland@acrypta.fr