UNIFORM RESULTS IN POLYNOMIAL-TIME SECURITY

Paul Barbaroux

L.R.I. bât.490
Université Paris XI
91405 Orsay
France

Abstract

Most security results can be established both in the non-uniform and the uniform model of computation. Nonetheless, non-uniform results are often much easier to obtain than their uniform version. In this paper we initiate a general framework in which the classical sampling technique can be applied to obtain uniform results. Our main theorem gives sufficient conditions under which a non-uniform result can be extended to a uniform one. As a consequence, we derive the uniform version of Schrift and Shamir's generalization of Yao's theorem on the universality of the next-bit test.

1 Introduction

A (perfect) pseudorandom source of bits is a probability distribution which cannot be distinguished in probabilistic polynomial time from a truly random source.

This notion was defined by Yao [8], who also showed that it is equivalent to the following simpler one: no bit of the source can be predicted from the previous ones in probabilistic polynomial time with probability significantly greater than 1/2. In other words, if all the bits "resist" to be predicted by some probabilistic polynomial time adversary, then the source can be used by probabilistic polynomial time algorithms as a truly random source of coin tosses. In short, this result states the universality of the next-bit test.

More generally, two distributions are indistinguishable if they cannot be distinguished by a probabilistic polynomial-time algorithm.

Schrift and Shamir [7] generalized Yao's result by finding a suitable version of the next-bit test (which they called the " comparative next-bit test ") which is universal for two arbitrary distributions.

In fact, their proof is established in the non-uniform model of security, where adversary algorithms are non-uniform Turing machines or sequences of boolean circuits, without any assumption on the computability of the sequence.

R.A. Rueppel (Ed.): Advances in Cryptology - EUROCRYPT '92, LNCS 658, pp. 297-306, 1993.

In general, in computational security, most results have a non–uniform and a uniform version (depending on which model of security is chosen). Historically, non–uniform results were obtained first, and extending them to the uniform case is often a hard task.

The main technique for obtaining uniform results is now classical. It can be found in Levin [5]: it consists in replacing true probabilities by approximations computed from random samplings.

The purpose of this paper is twofold. We prove that Schrift and Shamir's next–bit test remains universal in the uniform model of security. More importantly, we initiate a general framework in which the sampling technique can always be applied to obtain uniform results. Our main theorem specifies some conditions which are sufficient to extend a non–uniform result into a uniform one. In fact, the universality of the next–bit test becomes a consequence of our main theorem.

The theory of one–way functions and pseudorandom generators involves different notions of resistance of some object against an adversary. Two distributions are indistinguishable if they resist to be distinguished by a probabilistic polynomial–time adversary. A one–way function is a function which resists to be inverted. A bit is hard–core on a function f if it resists to be predicted using the knowledge of $f(x)$.

Several well known results can be restated in terms of reduction of one notion of resistance to another. For example, the universality of the next–bit test can be expressed as follows: the resistance of two distributions against an arbitrary adversary reduces to their resistance against the particular one who tries to distinguish them by the next–bit prediction. Let us quote a few other:

— Goldreich–Levin [2]: "if there exists a one–way function, then there exists a function with a hard–core bit". This result has been proved in [2] in both models of security.

— Impagliazzo–Levin–Luby: "if there exists a one–way function, then there exists a pseudorandom generator". This theorem was first proved under different types of restrictive assumptions ([1],[5]), then in the general case in [4], in the non–uniform model of security. It was then proved in the uniform model by Hastad [3].

— Rompel [6]: "if there exists a one–way function, then there exists a secure signature scheme".

In section 2, we give a few notations and some basic definitions about uniform and non–uniform computation. In section 3, we formalize the very general notion of resistance by defining a *security scheme* and the *resistance* of a security scheme. Basically, a security scheme is a predicate saying that some algorithm significantly succeeds in attacking some object. Then we define the *reduction* between two security schemes (reduction of a scheme to another means that the problem of proving the resistance of the former reduces to the same problem for the latter). Finally we illustrate our definitions by restating several theorems in terms of reducibility between security schemes.

In section 4, we introduce the notion of a *virtual algorithm* which will be crucial for extending non–uniform results to uniform ones. Intuitively, a virtual algorithm

can compute in a single (virtual) step the accepting probability of a probabilistic polynomial–time algorithm. Then we define approximations of virtual algorithms, where the virtual steps are replaced by approximations with polynomially small error. Our main theorem states that under some technical assumptions, if a reduction is achievable using some approximations of a virtual algorithm, then it is in fact achievable in the uniform model. We show then that Schrift and Shamir's reduction [7] can be extended to satisfy the assumptions of our main theorem. This will prove our second theorem about the uniform universality of the next–bit test.

2 Uniform vs. non–uniform algorithms

Notations

We write Σ for the set $\{0,1\}$; Σ^* denotes the set of all finite strings of bits, and $\Sigma^{\mathbf{N}}$ the set of all infinite strings of bits. For a string $x \in \Sigma^*$, $|x|$ denotes the length of x, x_i the i–th bit of x, and x_i^j the substring of x from bit i to bit j. The concatenation of x and y is denoted by xy. Following the usual definition, a *distribution ensemble* is a sequence $(D_n)_{n\in\mathbf{N}}$, where D_n is a probability distribution on Σ^n. We write $x \in_D E$ when x is picked up at random in the set E according to distribution D. U_n denotes the uniform distribution on Σ^n. U denotes the usual distribution on $\Sigma^{\mathbf{N}}$ (infinite sequence of independent and unbiased coin tosses).

When not specified otherwise, all the algorithms we consider are supposed to *run in polynomial time*. As usual, a uniform algorithm is a Turing Machine (or any equivalent model of computation). A non uniform algorithm is a Turing machine provided with a sequence $(a_n)_{n\in\mathbf{N}}$ of advices in Σ^* (it can also be viewed as a sequence (C_n) of boolean circuits of polynomial size). When not specified, these algorithms are supposed to be *probabilistic*. For simplicity of the notations, we represent the sequence of random coin tosses by a random input which is an *infinite* string $\omega \in_U \Sigma^{\mathbf{N}}$. In the probabilistic case the algorithms run in time bounded by a polynomial function of the input length, independently from the random string ω.

Definition 1 *A distribution ensemble (D_n) is* ***uniformly samplable*** *if there exists a uniform algorithm $A(n,\omega)$ such that, for every n, D_n is the distribution generated by taking $\omega \in_U \Sigma^{\mathbf{N}}$ and applying $A(n,.)$.*

In other words, for every n and every event E we have:

$$\Pr[y \in E] = \Pr[A(n,\omega) \in E], \text{ where } y \in_{D_n} E \text{ and } \omega \in_U \Sigma^{\mathbf{N}}.$$

3 Security schemes

Definition 2 *A* ***security scheme*** *is a predicate $P(\phi, n, c)$, whose free variables are supposed to be some function ϕ, an integer n, and a positive constant c.*

In the following definition and the rest of the paper, we will often identify an algorithm with the function it computes.

Definition 3 *We say that an algorithm A **breaks** the security scheme P (or is a **breaker** of P) if there exists a constant c such that, for an infinite number of values n, $P(A, n, c)$ is true.*

Definition 4 *A security scheme P is **resistant in the non-uniform (resp. uniform) model of security** if no non-uniform (resp. uniform) algorithm can break P.*

Definition 5 *Given two security schemes P and Q, we say that P is **reducible** to Q if the resistance of Q implies the resistance of P.*

The notion of reducibility depends on the model of security (uniform or non-uniform). We will always take the same model for both P and Q (that is: the adversary algorithms A and B are of the same nature: both uniform or not).

Example 1

Let (D_n) and (D'_n) be two distribution ensembles. Let $P_1(A, n, c)$ be

" $|\Pr[A(x,\omega) = 1] - \Pr[A(y,\omega) = 1]| > \frac{1}{n^c}$
where $x \in_{D_n} \Sigma^n$, $y \in_{D'_n} \Sigma^n$, $\omega \in_U \Sigma^N$ "

We call P_1 the *distinguishing scheme* of (D_n) and (D'_n). Resistance of P_1 means that (D_n) and (D'_n) are indistinguishable.

Example 2

Let (D_n) and (D'_n) be two distribution ensembles. Let $P_2(A, n, c)$ be

" A on input n computes some integer i such that $1 \leq i \leq n-1$, then acts on Σ^i and satisfies $\Pr[A(x_1^i, \omega) = x_{i+1}] - \Pr[A(y_1^i, \omega) = y_{i+1}] > \frac{1}{n^c}$, where $x \in_{D_n} \Sigma^n$, $y \in_{D'_n} \Sigma^n$, $\omega \in_U \Sigma^N$ ".

P_2 is the *next-bit distinguishing scheme* of (D_n) and (D'_n). Reducibility of P_1 to P_2 means that the next-bit test is universal. Schrift and Shamir [7] proved that indeed, in the non-uniform model, P_1 reduces to P_2. Our notion of resistance of P_1 corresponds to their notion of "passing the comparative next-bit test". Note that they took the difference of the two probabilities in absolute value, but the absolute value can obviously be dropped in the non-uniform model by considering the complementary event if necessary.

The reason for the non-uniformity of their result is that the integer i is obtained by the pigeonhole principle, and therefore not *computed* from n. Here we derive the uniform version as a corollary of our main theorem. In the uniform version, we obtain a stronger result if we drop the absolute value, since the sign, a priori, cannot be computationally decided.

Example 3
Let (f_n) be a polynomial–time computable sequence of functions : $\Sigma^n \to \Sigma^n$. Let $P_3(A, n, c)$ be

" $\Pr[A(f_n(x), \omega) \in f_n^{-1}(f_n(x))] > \frac{1}{n^c}$ where $x \in_{U_n} \Sigma^n$, $\omega \in_U \Sigma^{N^n}$.

P_3 is the *inversion scheme* of (f_n). Resistance of P_3 means that (f_n) is a one–way function.

Example 4
Let $(f_n) : \Sigma^n \to \Sigma^n$, and $(b_n) : \Sigma^n \to \Sigma$, and let $P_4(A, n, c)$ be

"$\Pr[A(f_n(x), \omega) = b_n(x)] > \frac{1}{2}(1 + \frac{1}{n^c})$ where $x \in_{U_n} \Sigma^n$, $\omega \in_U \Sigma^{N^n}$.

P_4 is the *prediction scheme* of the bit family (b_n) from (f_n). Resistance of P_4 means that (b_n) is hard–core on (f_n).(cf. [2])

Example 5
Let $(f_n) : \Sigma^n \to \Sigma^n$, and $P_4(A, n, c)$ be

"$\Pr[A(r, f_n(x), \omega) = r \odot x] > \frac{1}{2}(1 + \frac{1}{n^c})$ where $x, r \in_{U_n} \Sigma^n$, $\omega \in_U \Sigma^{N^n}$.

where $r \odot x$ denotes the boolean inner product of r and x. P_5 is the *inner product prediction scheme* of (f_n). The theorem of Goldreich and Levin [2] asserts that P_5 is reducible to P_3, in both models of security.

4 Obtaining uniform results

Looking carefully to the proof of the universality of the next–bit test in [7], one can see that the problem of achieving the uniform universality of the next–bit test comes from the non–computability (in polynomial time) of $\Pr_\omega[A(x, \omega) = 1]$, when given an algorithm $A(x, \omega)$. In fact, this is a rather general phenomenon: many non–uniform results can be trivially extended to the uniform case under the condition that we can compute such probabilities. Unfortunately, this is in general not the case since the straightforward computation of this probability takes exponential time in n.

Levin [5] showed that the exact probability computation is not really necessary. In fact, it can be replaced by a suitable approximation, such as the median of average values of random samplings. In this section, we formalize this general phenomenon and specify when this sampling technique can be applied. Then we can derive easily the uniform version of Schrift and Shamir's result.

Definition 6 *A **virtual algorithm** is a deterministic uniform algorithm for which the evaluation of exact probabilities is allowed and regarded as an elementary operation.*

In other words, a virtual algorithm M can use a "black box" which receives as input some subroutine $A : \Sigma^N \to \Sigma$ of M, and outputs $\Pr_\omega[A(\omega)] = 1]$.

*Given a real number r, we call an ϵ-**approximation of** r any real number r' such that $|r - r'| \leq \epsilon$.*

Definition 7 *Given a virtual algorithm $M(x)$, an integer n, and some positive constant c, an (n, c)-**approximation of** M is a virtual algorithm which runs like M and for which, on every input $|x|$ of length n, the black box gives, instead of its normal output, an n^{-c}-approximation.*

For all other inputs, we do not require any specific behaviour from the algorithm. We do not require either that an (n, c)-approximation run in polynomial time. (In fact, it is easy to construct a virtual algorithm M such that a suitable (n, c)-approximation of M might run forever, even on inputs of length n).

Definition 8 *A security scheme P is **accessible** if for every algorithm $A(x, \omega_1, \omega_2)$ which gets two random inputs ω_1 and ω_2, and for any positive constants c_1, c_2 such that $c_1 > c_2$, we have for n large enough:*

$$\Pr[P(A(.,.,\omega_2), n, c_2) \text{ is false }] \leq 2^{-n} \Rightarrow P(a, n, c_1) \text{ is true.}$$

It is easy to see that schemes P_2, P_3, P_4, P_5 in the examples of section 3 are accessible. More generally, any security scheme which says that some "suitable" expectation (depending on A, and taken over inputs x of length n and $\omega \in \Sigma^N$) is greater than $\frac{1}{n^c}$, is accessible. Note that this is not the case for P_1, since this scheme corresponds to an expectation which is taken in absolute value.

Theorem 1 ***(Main)** Let P and Q be two security schemes, such that Q is accessible. Assume that for every positive constant c, there exist positive constants c_1, c_2, c_3 and a virtual algorithm $M(A)$ (receiving A as a subroutine) such that for n large enough, every (n, c_2)-approximation $\tilde{M}(A)$ of $M(A)$ satisfies the following two conditions:*

1. *$\tilde{M}(A)$ has running time bounded by n^{c_3} on inputs of length n*
2. *$P(A, n, c) \Rightarrow Q(\tilde{M}(A), n, c_1)$*

Then P is uniformly reducible to Q.

We begin the proof with some lemmas which will be our tools for the sampling technique.

Lemma 1 *For every positive constant c, there exists a polynomial-time probabilistic algorithm $M(A, x, \omega')$ which, when given some algorithm $A(\omega)$ as a subroutine and $x = 1^n$ as input, computes with probability at least $\frac{7}{8}$ (on ω') an n^{-c}-approximation $\tilde{p}$ of $p = \Pr[A(\omega) = 1]$.*

Proof
The algorithm M is defined as follows:

compute $N = \lceil 2n^{2c} \rceil$
pick up independently at random $\omega_1, \cdots, \omega_N$
(the collection $< \omega_1, \cdots, \omega_N >$ will form ω')
compute $\tilde{p} = \frac{1}{N}$(number of i s.t. $A(\omega_i) = 1$)

Let $X_i(\omega_1, \cdots, \omega_N) = A(\omega_i)$. Then the X_i's are independent Bernouilli random variables with expectation $E(X_i) = p$ and variance $V(X_i) = p(1-p) \leq 1/4$. Chebyshev's inequality gives

$$\Pr[|\tilde{p} - p| \geq n^{-c}] \leq \frac{V(X_i)n^{2c}}{N} \leq \frac{n^{2c}}{4N} \leq \frac{1}{8}$$

Lemma 2 *Let $(X_i)_{1 \leq i \leq 2S+1}$ be independent random variables and p, ϵ such that $\Pr[|X_i - p| \geq \epsilon] \leq 1/8$. Let Y denote the median of the X_i's. Then*

$$\Pr[|Y - p| \geq \epsilon] \leq \frac{1}{S}$$

Proof

We have $\Pr[|Y - p| \geq \epsilon] = \Pr[|X_i - p| \geq \epsilon$ for at least $S+1$ subscripts i]

$$\begin{aligned}
&= \sum_{i=S+1}^{2S+1} \binom{2S+1}{i} \left(\tfrac{1}{8}\right)^i \left(1 - \tfrac{1}{8}\right)^{2S+1-i} \\
&\leq \sum_{i=S+1}^{2S+1} \binom{2S+1}{i} \left(\tfrac{1}{8}\right)^{S+1} \\
&\leq 2^{2S+1} \left(\tfrac{1}{8}\right)^{S+1} \\
&= \tfrac{1}{2^{S+2}} \leq \tfrac{1}{2^S}
\end{aligned}$$

Corollary 1 *For every positive constant c, there exists a probabilistic polynomial-time algorithm $M(A, x, y, \omega')$ which, on inputs $x = 1^n$ and $y = 1^S$, computes a n^{-c}-approximation of $p = \Pr_\omega[A(\omega) = 1]$ with probability (on ω') at least $1 - 2^{-S}$*

Proof

Just repeat $2S + 1$ times the algorithm of lemma 1 with independent random samplings, then take the median, and apply lemma 2.□

Proof of theorem 1

Let A be a uniform breaker of P. Then there exists c such that, for an infinite number of values n, $P(A, n, c)$ is true. Let $c_1, c_2, c_3, M(A)$ be such that they satisfy the two conditions of the theorem.

We define an algorithm B which will be a uniform breaker of Q: B, on input x, computes $S = n + \lceil c_3 \log_2 n \rceil$, where $n = |x|$. Then it simulates $M(A)$ while the number of computational steps of $M(A)$ does not exceed n^{c_3}. Whenever this number of steps is not sufficient to finish the simulation, B stops (and fails). Moreover, each time $M(A)$ makes a call to the black box for computing some probability p, B replaces this call by computing itself an n^{-c_2}-approximation $\tilde{p}$ of p with probability

of failure 2^{-S} (cf. Corollary 1), using independent random samplings for all these computations.

The total number of calls of $M(A)$ to the black box is at most n^{c_3}. Moreover, for each call we have $\Pr[|\tilde{p}-p| > n^{-c_2}] \leq 2^{-S}$, so with probability $\geq 1-n^{c_3}2^{-S} \geq 1-2^{-n}$, B is a (n, c_2)-approximation of $M(A)$. Whenever this happens, the forced halt does not occur before the end of the execution.

Let n be such that $P(A, n, c)$ is true, and large enough for the conditions of the theorem 1 to be satisfied. Then with probability $\geq 1 - 2^{-n}$ $Q(B, n, c_1)$ is true (using cond. 2). Finally B is a probabilistic uniform algorithm using 2 kinds of random inputs (those of A, and the random samplings, denoted by ω'), such that, for an infinite number of values of n, with probability on ω' greater than $1 - 2^{-n}$, $P(B(.,\omega'), n, c_1)$ is true (cond. 2). Hence using the accessibility of Q, B is a uniform breaker of Q. □

From this theorem we derive the following:

Theorem 2 ***(Uniform universality of the next-bit test)***
Given two uniformly samplable distribution ensembles, their distinguishing scheme is uniformly reducible to their next-bit distinguishing scheme.

Proof

We have seen that the next-bit distinguishing scheme P_2 is accessible.

We will show that Schrift and Shamir's proof for the non-uniform universality yields in fact constants c_1, c_2, c_3 and a virtual algorithm $M(A)$ satisfying the assumptions of the main theorem.

The virtual algorithm M is defined as follows (there are two parts in the algorithm: first M chooses an integer i in $\{1, \cdots, n-1\}$ and then predicts x_{i+1} from x_1^i).

for $i = 1$ **to** n
  compute $p_i = \Pr[A(y_1^i z_{i+1}^n, \omega) = 1]$ and $p_i' = \Pr[A(\tilde{y}_1^i z_{i+1}^n, \omega) = 1]$,
    where $y \in_{D_n} \Sigma^n$, $\tilde{y} \in_{D_n'} \Sigma^n$, $\omega \in_U \Sigma^N$. The computation is possible
    using the "black box" operation allowed in virtual algorithms:
    since (D_n) and (D_n') are both uniformly samplable,
    one can generate in polynomial time D_n and D_n' from the integer n,
    using the usual distribution on Σ^N.
  compute $q_i = p_i - p_i'$
choose an i (for instance, the first) such that $q_{i+1} - q_i > \frac{2}{3}n^{-(c+1)}$.
  (if there does not exist such an i, then the algorithm fails)
    (Here ends the first part of the algorithm)
choose at random $z_{i+1}^n \in_{U_{n-i}} \Sigma^{n-i}$ and $\omega \in_U \Sigma^N$ **if** $q_{i+1} - q_i > 0$ **then**
  if $A(x_1^i z_{i+1}^n, \omega) = 1$ **then output** z_{i+1} **else output** $1 - z_{i+1}$
else
  if $A(x_1^i z_{i+1}^n, \omega) = 1$ **then output** $1 - z_{i+1}$ **else output** z_{i+1}

Let us verify that the assumptions of the theorem 1 are satisfied. The existence of c_3 is obvious, since every approximation of $M(A)$ has the same running time as $M(A)$. Let us take $c_1, c_2 > c+1$. Then for n large enough we have

$$n^{-c_2} \leq \frac{1}{12} n^{-(c+1)} \tag{1}$$

and

$$n^{-c_1} \leq \frac{1}{3} n^{-(c+1)} \tag{2}$$

Let $\tilde{M}$ be a (n, c_2)–approximation of $M(A)$, that is: $\tilde{M}$ replaces p_i by $\tilde{p}_i$ s.t. $|\tilde{p}_i - p_i| \leq n^{-c_2}$ (and the same for p'_i) and therefore q_i by $\tilde{q}_i$ s.t. $|\tilde{q}_i - q_i| \leq 2n^{-c_2}$ and $q_{i+1} - q_i$ by $\tilde{q}_{i+1} - \tilde{q}_i$ s.t.

$$|(\tilde{q}_{i+1} - \tilde{q}_i) - (q_{i+1} - q_i)| \leq 4n^{-c_2} \tag{3}$$

Suppose now $P_1(A, n, c)$ is true, that is:

$$|\Pr[A(y, \omega) = 1] - \Pr[A(\tilde{y}, \omega) = 1]| > n^{-c}$$

where $y \in_{D_n} \Sigma^n$, $\tilde{y} \in_{D'_n} \Sigma^n$, $\omega \in_U \Sigma^N$. Then by the pigeonhole principle (cf. [7]) there exist an i s.t. $|q_{i+1} - q_i| > n^{-(c+1)}$, that is, using (1),

$$|\tilde{q}_{i+1} - \tilde{q}_i| \geq |q_{i+1} - q_i| - 4n^{-c_2} > n^{-c+1} - 4n^{-c_2} \geq \frac{2}{3} n^{-(c+1)} \tag{4}$$

so $\tilde{M}$ does not fail and finds such an i. Then for this i we have

$$\frac{2}{3} n^{-(c+1)} < |\tilde{q}_{i+1} - \tilde{q}_i| < |q_{i+1} - q_i| + 4n^{-c_2} < |q_{i+1} - q_i| + \frac{1}{3} n^{-(c+1)}$$

so

$$\frac{1}{3} n^{-(c+1)} < |q_{i+1} - q_i| \tag{5}$$

Now we have to prove that $\tilde{M}$ can decide the sign of $q_{i+1} - q_i$ by looking at the sign of $\tilde{q}_{i+1} - \tilde{q}_i$. But if these two quantities were of opposite signs, this would imply, using (4) and (5):

$$|(\tilde{q}_{i+1} - \tilde{q}_i) - (q_{i+1} - q_i)| > n^{-(c+1)}$$

and therefore, using (3):

$$4n^{-c_2} > n^{-(c+1)}$$

which contradicts the choice of c_2.

Now the proof ends as in [7]: it is easy to see that

$$\Pr[\tilde{M}(y_1^i, \omega') = y_{i+1}] = \frac{1}{2} + \epsilon(p_{i+1} - p_i)$$

and

$$\Pr[\tilde{M}(\tilde{y}_1^i, \omega') = \tilde{y}_{i+1}] = \frac{1}{2} + \epsilon(p'_{i+1} - p'_i),$$

where ϵ denotes the sign of $q_{i+1} - q_i$. Therefore $\Pr[\tilde{M}(y_1^i, \omega') = y_{i+1}] - \Pr[\tilde{M}(\tilde{y}_1^i, \omega') = \tilde{y}_{i+1}]$ is positive, and equals $|q_{i+1} - q_i| > \frac{1}{3}n^{-(c+1)} \geq n^{-c_1}$. Hence $P_2(\tilde{M}, n, c_1)$ is true.

Bibliography

[1] O. Goldreich, H. Krawcyzk, M. Luby, "On the Existence of Pseudorandom Generators,", Proc. FOCS 1988, pp. 12–24.

[2] O. Goldreich, L.A Levin, "A Hard-Core Predicate For All One–Way Function", Proc. STOC 1989, pp. 25–32.

[3] J. Hastad, "Pseudorandom Generators Under Uniform Assumptions", Proc. STOC 1990, pp. 395–404.

[4] R. Impagliazzo, L.A. Levin, M. Luby, "Pseudo–Random Generation from One–Way Functions", Proc. STOC 1989, pp. 12–24.

[5] L.A. Levin, "One–Way Functions and Pseudorandom Generators", Combinatorica 7(4), pp. 357–363, 1987.

[6] J. Rompel, "One–Way Functions are Necessary and sufficient for Secure Signatures", Proc. STOC 1990, pp. 387–394.

[7] A.W. Schrift, A.Shamir, "On the Universality of the Next Bit Test", Proc. CRYPTO 1990, pp. 394–408.

[8] A.C. Yao, "Theory and Applications of Trapdoor Functions", Proc. FOCS 1982, pp. 80–91.