

Pseudorandom Generators, a Survey

Claus P. SCHNORR

Department of Computer Science and Mathematics
Johann Wolfgang Goethe-Universität
Frankfurt am Main

Logic, Computability and Randomness,
January 10-13, 2007 – Buenos Aires, Argentina

Contents

History : Von MISES, KOLMOGOROV
individual random sequences

drawback for applications:

no efficient construction of individual random sequences.

HIGHLIGHTS

- Pseudorandom generators, Statistical Tests
- Indistinguishability, Universality of Predictors
- Quadratic Residuosity Assumption
- The $x^2 \bmod N$ generator and Factoring
- Other one-way Functions, Complexity Assumptions
- Weaning Polynomial, Exponential PRG
- Perfect PRG's via Exponentiation if Factoring is Hard
- Conclusions, Prospects

Pseudorandom Generators

YAO and BLUM-MICALI (1982), **goal**:

stretch random sequences $(x_1, \dots, x_n) \in \{0, 1\}^n$ into longer pseudorandom sequences $G(x_1, \dots, x_n) \in \{0, 1\}^{\ell(n)}$,
 $n < \ell(n) = n^{O(1)}$.

Pseudorandom generator, (PRG), informally:

a poly-time computable function $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$,
 $n < \ell(n) = n^{O(1)}$, such that the sequence of prob. distributions

$$U_{\ell(n)}: (y_1, \dots, y_{\ell(n)}) \in_R \{0, 1\}^{\ell(n)}$$

$$G(U_n): G(x_1, \dots, x_n) \in \{0, 1\}^{\ell(n)} \text{ for } (x_1, \dots, x_n) \in_R \{0, 1\}^n$$

cannot be distinguished in poly-time.

G is "*uniform*" poly-time, if it is computed by a deterministic (prob.) Turing machine in time $O(n^t)$ for all n and some $t > 0$.

"*non-uniform*" poly-time: $G|_{\{0,1\}^n}$ is computed for each n by some boolean circuit of size $O(n^t)$.

Statistical Tests, Indistinguishability

Def. The ensembles $(U_{\ell(n)})_{n \in \mathbb{N}}$ and $(G(U_n))_{n \in \mathbb{N}}$ are **poly-time indistinguishable** if for all poly-time $T : \{0, 1\}^* \rightarrow \{0, 1\}$

$$|\Pr[T(y_1, \dots, y_{\ell(n)}) = 1] - \Pr[T(G(x_1, \dots, x_n)) = 1]| = O(n^{-t})$$

for all $t > 0$, $(y_1, \dots, y_{\ell(n)}) \in_R \{0, 1\}^{\ell(n)}$, $(x_1, \dots, x_n) \in_R \{0, 1\}^n$.

We call T a poly-time **statistical test** and define

$$\text{dist}_T(U_{\ell(n)}, G(U_n)) =_{\text{def}}$$

$$|\Pr[T(y_1, \dots, y_{\ell(n)}) = 1] - \Pr[T(G(x_1, \dots, x_n)) = 1]|.$$

The sequences $(U_{\ell(n)})_{n \in \mathbb{N}}$ and $(G(U_n))_{n \in \mathbb{N}}$ are poly-time ensembles.

Def. The ensemble $(G(U_n))_{n \in \mathbb{N}}$ is **pseudorandom** if the ensembles $(U_{\ell(n)})_{n \in \mathbb{N}}$ and $(G(U_n))_{n \in \mathbb{N}}$ are poly-time indistinguishable. G is a **perfect PRG** if the ensemble $(G(U_n))_{n \in \mathbb{N}}$ is pseudorandom.

Predictors, Unpredictability

How to prove perfectness ?

1. Express arbitrary stat. tests by *predictors*.
2. Use *complexity assumptions* to prove unpredictability.

Let $(X_n)_{n \in \mathbb{N}}$ be an ensemble, X_n a prob. distr. on $\{0, 1\}^{\ell(n)}$.

Advantage of $F : \{0, 1\}^* \rightarrow \{0, 1\}$ in predicting the next bit:

$$\text{adv}_F(i, X_n) := \left| \Pr[F(x_1, \dots, x_i) = x_{i+1}] - \frac{1}{2} \right|$$

for $(x_1, \dots, x_{\ell(n)}) \in X_n \{0, 1\}^{\ell(n)}$.

Def. The ensemble $(X_n)_{n \in \mathbb{N}}$ is **unpredictable** to the right if for all poly-time predictors F

$$\max_{i < \ell(n)} |\text{adv}_F(i, X_n)| = O(n^{-t}) \quad \text{for all } t > 0.$$

informally: Given the first i bits x_1, \dots, x_i the next bit x_{i+1} can only be guessed **negligibly** better than with prob. $\frac{1}{2}$. The prob. refers to $(x_1, \dots, x_{\ell(n)}) \in X_n \{0, 1\}^{\ell(n)}$.

Universality of Predictors

Thm. [Yao, 1982]

The ensemble $(X_n)_{n \in \mathbb{N}}$ is *pseudorandom* iff it is *unpredictable*.
It is unpredictable to the right iff it is unpredictable to the left.

Proof.

" \Rightarrow " A successful predictor yields a rejecting stat. test.

" \Leftarrow " Given a stat. test T such that $\text{dist}_T(U_{\ell(n)}, X_n) \geq \varepsilon_n > 0$,
we construct a predictor F such that

$$\max_{i < \ell(n)} \text{adv}_F(i, X_n) \geq \varepsilon_n / \ell(n).$$

Intertwine X_n and $U_{\ell(n)}$:

$$X_n = X_{0,n}, X_{1,n}, \dots, X_{i,n}, \dots, X_{\ell(n),n} = U_{\ell(n)},$$

where $X_{i,n}$: $(y_1, \dots, y_{\ell(n)-i}, x'_1, \dots, x'_i) \in \{0, 1\}^{\ell(n)}$ for

$$(y_1, \dots, y_{\ell(n)}) \in_{X_n} \{0, 1\}^{\ell(n)} \text{ and } (x'_1, \dots, x'_i) \in_R \{0, 1\}^i.$$

Obviously $\text{dist}_T(X_n, U_{\ell(n)}) \leq \sum_{i=0}^{\ell(n)-1} \text{dist}_T(X_{i,n}, X_{i+1,n})$.

Hence $\exists i < \ell(n) : \text{dist}_T(X_{i,n}, X_{i+1,n}) \geq \varepsilon_n / \ell(n)$.

Proof end, BLUM-MICALI Paradigm

Therefore the stat. test T distinguishes for some i

given $y_1, \dots, y_{\ell(n)-i}$ the next bit $y_{\ell(n)-i+1}$

from a random bit x'_i with advantage $\geq \varepsilon_n/\ell(n)$.

Define the predictor F accordingly, such that it predicts $y_{\ell(n)-i+1}$ with advantage $\geq \varepsilon_n/\ell(n)$. \square

The BLUM-MICALI PRG construction (1982):

Iterate a one-way permutation $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ outputting some hardcore bits $H(x)$ per iteration:

INPUT $x_0 \in_R \{0, 1\}^n$

FOR $i = 0, \dots, \ell(n)$ DO $x_{i+1} := f(x_i)$, OUTPUT $H(x_i)$.

Thm. [BLUM, MICALI, 1984]

If $H(x)$ is **hard** (i.e. perfect pseudorandom) for given $f(x)$ then $G(x_0) := (H(x_0), \dots, H(x_{\ell(n)}))$ is a perfect PRG.

How to get one-way permutations ?

The Quadratic Residuosity Assumption

Notation. Let $p, q = 3 \pmod 4$ be primes, $N = p \cdot q$ is a Blum integer, let $Blum_n$ denote the set of all n -bit Blum integers.

$$\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z} \cong \{0, \dots, N-1\}, \mathbb{Z}_N^* = \{x \in \mathbb{Z}_N \mid \gcd(x, N) = 1\},$$

$$QR_N = \{x^2 \mid x \in \mathbb{Z}_N^*\}, J_N(x) \text{ is the JACOBI symbol,}$$

$$\mathbb{Z}_N^{+1} = \{x \in \mathbb{Z}_N^* \mid J_N(x) = 1\}, \text{ groups: } QR_N \subset \mathbb{Z}_N^{+1} \subset \mathbb{Z}_N^*,$$

Squaring, $Sq : QR_N \rightarrow QR_N, x \mapsto x^2 \pmod N$ is a bijection for Blum integers N , $|QR_N| = |\mathbb{Z}_N^{+1}|/2 = |\mathbb{Z}_N^*|/4$, $-1 \in \mathbb{Z}_N^{+1} \setminus QR_N$.

Thm. $Sq : QR_N \rightarrow QR_N$ is a one-way permutation if N is hard to factor. One Sq -inversion factors $N \in_R Blum_n$ with prob. $\frac{1}{2}$.

QR-problem. Given $x \in_R \mathbb{Z}_N^{+1}$, $N \in_R Blum_n$ decide $x \in? QR_N$.

The advantage of a QR-decision algorithm $AL : \mathbb{Z}_N^{+1} \rightarrow \{0, 1\}$:
 $adv_{AL} := |\Pr_x[AL(x) = \chi_{QR_N}(x)] - \frac{1}{2}|$ for $x \in_R \mathbb{Z}_N^{+1}$, $N \in_R Blum_n$.

QR-assumption. All poly-time QR-decision algorithms AL have negligible advantage: $adv_{AL} = O(n^{-t})$ for all $t > 0$.

The $x^2 \bmod N$ Generator

The QR-assumption implies that factoring N is not in poly-time.

INPUT $N \in_R \text{Blum}_n$, $x_1 \in_R \text{QR}_N$,

FOR $i = 1, \dots, \ell(n) = n^{O(1)}$ DO

$x_{i+1} := x_i^2 \bmod N$ such that $0 < x_{i+1} < N$,

OUTPUT $x_{i+1} \bmod 2 \in \{0, 1\}$.

$G(x_1) = (x_1 \bmod 2, x_2 \bmod 2, \dots, x_{\ell(n)} \bmod 2) \in \{0, 1\}^{\ell(n)}$.

Thm. [BLUM, BLUM, SHUB, 1986] Under the QR-assumption the $x^2 \bmod N$ generator is unpredictable to the left.

Proof. Exactly one of $\pm x_i$ is in QR_N . The least significant bit of $x_i \bmod 2$ carries the QR-information, which of $\pm x_i$ is in QR_N :

$$(-x_i \bmod 2) = (N - x_i \bmod 2) = -(x_i \bmod 2).$$

Therefore, given x_{i+1} and $\pm x_i$ computing the $x_i \bmod 2$ requires to decide quadratic residuosity of $\pm x_i$.

Random Selfreducibility of QR_N

Thm. If $QR_N \subset \mathbb{Z}_N^*$ can be decided in poly-time with advantage $\geq n^{-t}$ then it can be decided in poly-time with prob. ≈ 1 .

Proof. $x \in QR_N \Leftrightarrow xr_1^2, \dots, xr_m^2 \in QR_N$ for $r_1, \dots, r_m \in \mathbb{Z}_N^*$.
Let AL have advantage ε in deciding QR_N .

Deciding with high probability whether $x \in QR_N$:

Pick random $r_i \in_R \mathbb{Z}_N^*$ for $i = 1, \dots, m \geq n^t$

Decide that $x \in QR_N$ if $AL(xr_i^2) = 1$ for the majority of the i .

Hence, a non-negligible advantage $\geq n^{-t}$ can be raised by majority decision in poly-time to near certainty. \square

Deciding QR_N might be easier than factoring N .

We don't know whether deciding QR_N can help factoring N .

Perfect PRG if Factoring is Hard

Thm. [ALEXI, CHOR, GOLDREICH, SCHNORR, 1988] The $x^2 \bmod N$ generator is perfect if factoring $N \in_R \text{Blum}_n$ is hard.

N can be factored in poly-time given any poly-time stat. test that breaks the $x^2 \bmod N$ generator G . FISCHLIN, SCHNORR (2000) have greatly improved the efficiency of this factoring method.

The $x^2 \bmod N$ generator **with m output bits** per iteration.

INPUT $N \in_R \text{Blum}_n$, $x_1 \in_R \text{QR}_N$,

FOR $i = 1, \dots, \ell(n)$ DO

$x_{i+1} := x_i^2 \bmod N$ such that $0 < x_{i+1} < N$,

OUTPUT $x_{i+1} \bmod 2^m \in [0, 2^m[\cong \{0, 1\}^m$.

$G(x_1) = (x_1 \bmod 2^m, x_2 \bmod 2^m, \dots, x_{\ell(n)} \bmod 2^m) \in \{0, 1\}^{\ell(n)}$.

Thm. G is perfect for $m = O(\log_2 n)$ if factoring N is hard.

Better Complexity Assumptions

Factoring N is not known to break the $x^2 \bmod N$ generator. The hardness of factoring N may not be required.

A "better" complexity assumption for the $x^2 \bmod N$ generator:

Indistinguishability Assumption. [MICALI, SCHNORR, 1991]

The following ensembles are poly-time indistinguishable for $N \in_R \text{Blum}_n$, $e \geq 2$, $\gcd(e, \phi(N)) = 1$, $k \leq n/4 - (\log n)^2/2$:

- $(N, x^e \bmod N)$ for $x \in_R [1, N2^{-k}]$.
- (N, y) for $y \in_R [1, N]$.

Thm. [MS, 1991] Let $k \leq n/4 - (\log n)^2/2$, $e \geq 3$ and N an n -bit prime. Then $(x^e \bmod N) \bmod 2^k \in [0, 2^k[\cong \{0, 1\}^k$ is for $x \in_R [1, N2^{-k}]$ statistically random within $\varepsilon = O(e n^2 n^{-\log n})$.

The proof uses exponential sums [NIEDERREITER, 1988].
The Indistinguishability Assumption can be proved "locally".

Weaning Polynomial PRG

The **weaning** polynomial generator, $n = \lceil \log_2 N \rceil$.

INPUT $N \in_R \text{Blum}_n$, $x_1 \in_R [1, N - 1]$, $e \geq 2$, $k = k(e, n)$,

FOR $i = 1, \dots, \ell(n) = n^{O(1)}$ DO

$$x_{i+1} := \lfloor (x_i^e \bmod N) / 2^k \rfloor,$$

$$\text{OUTPUT } \text{out}(x_i) := (x_i^e \bmod N) \bmod 2^k \in [0, 2^k[\cong \{0, 1\}^k.$$

$$G(x_1) = (\text{out}(x_1), \text{out}(x_2), \dots, \text{out}(x_{\ell(n)})) \in \{0, 1\}^{\ell(n)}.$$

Weaning property. $x_i^e \bmod N$ splits into $\text{out}(x_i)$ and x_{i+1} , the k least and $n - k$ most significant bits.

Thm. [MS 1991] The weaning polynomial PRG is perfect if the Indistinguishability Assumption holds for the chosen e, k .

Efficiency. For $e = 17$ the PRG outputs k bits per 5 multiplication/squarings modulo N . For $k = n/4 - (\log n)^2/2$ and a n -bit prime N this speeds up the $x^2 \bmod N$ generator by a factor $\Theta(n / \log_2 n)$.

One-way Functions, Complexity Assumptions

If multiplication of primes $p, q \mapsto p \cdot q$ is one-way then the $x^2 \bmod N$ generator is perfect. By definition one-way functions cannot be inverted in poly-time. Other one-way functions ?

Thm. [Hastad, Impagliazzo, Levin, Luby, 1992]

A perfect PRG can be constructed from any one-way function.

This result is theoretically important. Every perfect PRG is one-way. Hence, perfect PRG exist iff poly-time one-way functions exist. This stops short of proving that PRG's exist if $\mathbf{NP} \neq \mathbf{P}$. The general construction is impractical.

Thm. [RAZBOROV, RUDICH, 1994] (*informal*) "Natural" proofs of exponential circuit-size lower bounds yield a poly-time stat. test that breaks exponential hardness of all poly-time RNG's.

Unproven complexity assumptions are unavoidable.

Preferable assumptions ? Hardness of discrete logarithm / factoring, see the **generic group model** for elliptic curves.

Exponential PRG

Exponentiation mod N yields a perfect PRG outputting $O(\log n)$ bits per exponentiation if exponentiation is one-way, BLUM, MICALI. The discrete log problem is random selfreducible. The $O(\log n)$ bound can be raised if certain N are hard to factor.

Let $N = p \cdot q \in_R \text{Blum}_n$. Denote $f_{g,N}(x) = g^x \bmod N$.
Let $g \in_R \text{QR}_N$ generate a large subgroup of \mathbb{Z}_N^* .

Thm. [HASTAD, SCHRIFT, SHAMIR, 1993] If factoring N is hard the $n/2$ least/most significant bits of x are simultaneously hard (i.e. pseudorandom) for given $f_{g,N}(x)$. All bits of x are individually hard except for the $O(\log n)$ most significant bits.

Cor. By iterating $f_{g,N}(x)$ via hashing one can output $n/2 - O(\log n)^2$ pseudorandom bits per exponentiation $x \mapsto g^x \bmod N$.

Straight forward exponentiation costs $\frac{3}{2}n$ multiplications mod N . This yields a PRG that is similarly efficient as the $x^2 \bmod N$ generator with $\log_2 n$ output bits per iteration.

Exponential Weaning PRG

DEDIC, REYZIN, VADHAN (2003) improve the [HSS]-generator. They present various simpler, exponential weaning PRG's with $n/2 - O(\log n)$ output bits per exponentiation without hashing.

- Let $N = p \cdot q \in_R \text{Blum}_n$ with *equally sized, safe* primes p, q .
- Let $s, \bar{s} \in \mathbb{Z}_N^* \setminus QR_N$ such that $J_N(s) = 1$ and $J_N(\bar{s}) = -1$.
- $g \in_R QR_N$ a generator of QR_N , $m = \lceil n/2 \rceil - O(\log n)$

INPUT $N, y = \sum_{i=1}^n y_i 2^{i-1} \in_R [1, N-1], y_i \in \{0, 1\}$.

FOR $i = 1, \dots, \ell(n) = n^{O(n)}$ DO

$$y := g^{\lfloor y/2^{m+3} \rfloor} s^{y_{m+2}} \bar{s}^{y_{m+1}} \bmod N$$

$$\text{OUTPUT out}(i) := y \bmod 2^m,$$

Thm. $G(y) := (\text{out}(1), \text{out}(2), \dots, \text{out}(\ell(n))) \in \{0, 1\}^{m\ell(n)}$ is a perfect PRG if factoring those $N \in_R \text{Blum}_n$ is not in poly-time.

DEDIC, REYZIN, VADHAN speed up exponentiation by halving the exponent $\lfloor y/2^{m+3} \rfloor$ to $\leq n/2$ bits.

Conclusions, Prospects

The [DEDIC, REYZIN, VADHAN]-PRG outputs $n/2 - O(\log n)$ bits **per exponentiation**.

The polynomial weaning generator [MS 91] should be perfect when outputting $n/4 - O(\log n)^2$ bits **per polynomial evaluation** within $\Theta(1)$ multiplications modulo N .

Factoring $\notin \mathbf{P}$ does not exhaust the potential of weaning PRG.

Is the link between factoring and perfect PRG's natural ?

Primes $\in \mathbf{P}$, randomness no more needed [Agrawal, alii, 2002].
Why should perfect PRG's require primes and BLUM integers ?

Conj. The [MS 91] Indist. Ass. holds for random n -bit $N \in \mathbb{N}$.

The conj. can be proved "locally", namely that blocks of substrings distribute statistically close to the uniform distribution. This gives positive evidence that $\mathbf{P} \neq \mathbf{NP}$.