Pseudorandom Sequence Generation

1 Distinguishability and Bit Prediction

Let \( D \) be a probability distribution on a finite set \( \Omega \). Then \( D \) associates a probability \( P_D(\omega) \) with each each element \( \omega \in \Omega \). We will also regard \( D \) as a random variable that ranges over \( \Omega \) and assumes value \( \omega \in \Omega \) with probability \( P_D(\omega) \).

Definition: An \((S, \ell)\)-pseudorandom sequence generator (PRSG) is a function \( f: S \to \{0, 1\}^\ell \). (We generally assume \( 2^\ell \gg |S| \).) More properly speaking, a PRSG is a randomness amplifier. Given a random, uniformly distributed seed \( s \in S \), the PRSG yields the pseudorandom sequence \( z = f(s) \). We use \( S \) also to denote the uniform distribution on seeds, and we denote the induced probability distribution on pseudorandom sequences by \( f(S) \).

The goal of an \((S, \ell)\)-PRSG is to generate sequences that “look random”, that is, are computationally indistinguishable from sequences drawn from the uniform distribution \( U \) on length-\( \ell \) sequences. Informally, a probabilistic algorithm \( A \) that always halts “distinguishes” \( X \) from \( Y \) if its output distribution is “noticeably different” depending whether its input is drawn at random from \( X \) or from \( Y \). Formally, there are many different kinds of distinguishably. In the following definition, the only aspect of \( A \)’s behavior that matters is whether or not it outputs “1”.

Definition: Let \( \epsilon > 0 \), let \( X, Y \) be distributions on \( \{0, 1\}^\ell \), and let \( A \) be a probabilistic algorithm. Algorithm \( A \) naturally induces probability distributions \( A(X) \) and \( A(Y) \) on the set of possible outcomes of \( A \). We say that \( A \) \( \epsilon \)-distinguishes \( X \) and \( Y \) if

\[
|P[A(X) = 1] - P[A(Y) = 1]| \geq \epsilon,
\]

and we say \( X \) and \( Y \) are \( \epsilon \)-indistinguishable by \( A \) if \( A \) does not distinguish them.

A natural notion of randomness for PRSG’s is that the next bit should be unpredictable given all of the bits that have been generated so far.

Definition: Let \( \epsilon > 0 \) and \( 1 \leq i \leq \ell \). A probabilistic algorithm \( N_i \) is an \( \epsilon \)-next bit predictor for bit \( i \) of \( f \) if

\[
P[N_i(Z_1, \ldots, Z_{i-1}) = Z_i] \geq \frac{1}{2} + \epsilon
\]

where \((Z_1, \ldots, Z_\ell)\) is distributed according to \( f(S) \).

A still stronger notion of randomness for PRSG’s is that each bit \( i \) should be unpredictable, even if one is given all of the bits in the sequence except for bit \( i \).

Definition: Let \( \epsilon > 0 \) and \( 1 \leq i \leq \ell \). A probabilistic algorithm \( B_i \) is an \( \epsilon \)-strong bit predictor for bit \( i \) of \( f \) if

\[
P[B_i(Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_\ell) = Z_i] \geq \frac{1}{2} + \epsilon
\]

where \((Z_1, \ldots, Z_\ell)\) is distributed according to \( f(S) \).
The close relationship between distinguishability and the two kinds of bit prediction is established in the following theorems.

**Theorem 1** Suppose \( \epsilon > 0 \) and \( N_i \) is an \( \epsilon \)-next bit predictor for bit \( i \) of \( f \). Then algorithm \( B_i \) is an \( \epsilon \)-strong bit predictor for bit \( i \) of \( f \), where algorithm \( B_i(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_\ell) \) simply ignores its last \( \ell - i \) inputs and computes \( N_i(z_1, \ldots, z_{i-1}) \).

**Proof:** Obvious from the definitions.

Let \( \mathbf{x} = (x_1, \ldots, x_\ell) \) be a vector. We define \( \mathbf{x}^i \) to be the result of deleting the \( i \)-th element of \( \mathbf{x} \), that is, \( \mathbf{x}^i = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_\ell) \).

**Theorem 2** Suppose \( \epsilon > 0 \) and \( B_i \) is an \( \epsilon \)-strong bit predictor for bit \( i \) of \( f \). Then algorithm \( A \) \( \epsilon \)-distinguishes \( f(S) \) and \( U \), where algorithm \( A \) on input \( \mathbf{x} \) outputs 1 if \( B_i(\mathbf{x}^i) = x_i \) and outputs 0 otherwise.

**Proof:** By definition of \( A \), \( A(\mathbf{x}) = 1 \) precisely when \( B_i(\mathbf{x}^i) = x_i \). Hence, \( P[A(f(S)) = 1] \geq 1/2 + \epsilon \). On the other hand, for \( r = U \), \( P[B_i(r^i) = r_i] = 1/2 \) since \( r_i \) is a uniformly distributed bivalued random variable that is independent of \( r^i \). Thus, \( P[A(U) = 1] = 1/2 \), so \( A \) \( \epsilon \)-distinguishes \( f(S) \) and \( U \).

For the final step in the 3-way equivalence, we have to weaken the error bound.

**Theorem 3** Suppose \( \epsilon > 0 \) and algorithm \( A \) \( \epsilon \)-distinguishes \( f(S) \) and \( U \). For each \( 1 \leq i \leq \ell \) and \( c \in \{0, 1\} \), define algorithm \( N^c_i(z_1, \ldots, z_{i-1}) \) as follows:

1. Flip coins to generate \( \ell - i + 1 \) random bits \( r_i, \ldots, r_\ell \).
2. Let \( v = \begin{cases} 1 & \text{if } A(z_1, \ldots, z_{i-1}, r_i, \ldots, r_\ell) = 1; \\ 0 & \text{otherwise}. \end{cases} \)
3. Output \( v \oplus r_i \oplus c \).

Then there exist \( m \) and \( c \) for which algorithm \( N^c_m \) is an \( \epsilon/\ell \)-next bit predictor for bit \( m \) of \( f \).

**Proof:** Let \( (Z_1, \ldots, Z_\ell) = f(S) \) and \( (R_1, \ldots, R_\ell) = U \) be random variables, and let \( D_i = (Z_1, \ldots, Z_i, R_{i+1}, \ldots, R_\ell) \). \( D_i \) is the distribution on \( \ell \)-bit sequences that results from choosing the first \( i \) bits according to \( f(S) \) and choosing the last \( \ell - i \) bits uniformly. Clearly \( D_0 = U \) and \( D_\ell = f(S) \).

Let \( p_i = P[A(D_i) = 1], 0 \leq i \leq \ell \). Since \( A \) \( \epsilon \)-distinguishes \( D_\ell \) and \( D_0 \), we have \( |p_\ell - p_0| \geq \epsilon \).

Hence, there exists \( m, 1 \leq m \leq \ell \), such that \( |p_m - p_{m-1}| \geq \epsilon/\ell \). We show that the probability that \( N^c_m \) correctly predicts bit \( m \) for \( f \) is \( 1/2 + (p_m - p_{m-1}) \) if \( c = 1 \) and \( 1/2 + (p_{m-1} - p_m) \) if \( c = 0 \). It will follow that either \( N^0_m \) or \( N^1_m \) correctly predicts bit \( m \) with probability \( 1/2 + |p_m - p_{m-1}| \geq \epsilon/\ell \).

Consider the following experiments. In each, we choose an \( \ell \)-tuple \( (z_1, \ldots, z_\ell) \) according to \( f(S) \) and an \( \ell \)-tuple \( (r_1, \ldots, r_\ell) \) according to \( U \).

**Experiment E_0:** Succeed if \( A(z_1, \ldots, z_{m-1}, z_m, r_{m+1}, \ldots, r_\ell) = 1 \).

**Experiment E_1:** Succeed if \( A(z_1, \ldots, z_{m-1}, z_m, r_{m+1}, \ldots, r_\ell) = 1 \).

**Experiment E_2:** Succeed if \( A(z_1, \ldots, z_{m-1}, z_m, r_{m+1}, \ldots, r_\ell) = 1 \).
Let \( q \) be the probability that experiment \( E \) succeeds, where \( j = 0, 1, 2 \). Clearly \( q = (q_0 + q_1)/2 \) since \( r_m = z_m \) is equally likely as \( r_m = \neg z_m \).

Now, the inputs to \( A \) in experiment \( E_0 \) are distributed according to \( D_m \), so \( p_m = q_0 \). Also, the inputs to \( A \) in experiment \( E_2 \) are distributed according to \( D_m - 1 \), so \( p_m - 1 = q_2 \). Differentiating, we get \( p_m - p_m - 1 = q_0 - q_2 = (q_0 - q_1)/2 \).

We now analyze the probability that \( N_m \) correctly predicts \( m \) of \( f(S) \). Assume without loss of generality that \( A \)'s output is in \( \{0, 1\} \). A particular run of \( N_m(z_1, \ldots, z_m) \) correctly predicts \( z_m \) if

\[
A(z_1, \ldots, z_{m-1}, z_m, r_{m-1}, \ldots, r_{\ell}) + r_m \oplus c = z_m
\]

If \( r_m = z_m \), \((1)\) simplifies to

\[
A(z_1, \ldots, z_{m-1}, z_m, \ldots, r_{\ell}) = c,
\]

and if \( r_m = \neg z_m \), \((1)\) simplifies to

\[
A(z_1, \ldots, z_{m-1}, \neg z_m, \ldots, r_{\ell}) = \neg c.
\]

Let \( \text{OK}_m^c \) be the event that \( N_m(Z_1, \ldots, Z_m) = Z_m \), i.e., that \( N_m \) correctly predicts \( m \) for \( f \). From \((2)\), it follows that

\[
P[\text{OK}_m^c \mid R_m = Z_m] = \begin{cases} q_0 & \text{if } c = 1 \\ (1 - q_0) & \text{if } c = 0 \end{cases}
\]

for in that case the inputs to \( A \) are distributed according to experiment \( E_0 \). Similarly, from \((3)\), it follows that

\[
P[\text{OK}_m^c \mid R_m = \neg Z_m] = \begin{cases} q_1 & \text{if } \neg c = 1 \\ (1 - q_1) & \text{if } \neg c = 0 \end{cases}
\]

for in that case the inputs to \( A \) are distributed according to experiment \( E_1 \). Since \( P[R_m = Z_m] = P[R_m = \neg Z_m] = 1/2 \), we have

\[
P[\text{OK}_m^c] = \frac{1}{2} \cdot P[\text{OK}_m^c \mid R_m = Z_m] + \frac{1}{2} \cdot P[\text{OK}_m^c \mid R_m = \neg Z_m]
\]

\[
= \begin{cases} q_0/2 + (1 - q_1)/2 = 1/2 + p_m - p_m - 1 & \text{if } c = 1 \\ q_1/2 + (1 - q_0)/2 = 1/2 + p_m - 1 - p_m & \text{if } c = 0 \end{cases}
\]

Thus, \( P[\text{OK}_m^c] = 1/2 + |p_m - p_m - 1| \geq \epsilon/\ell \) for some \( c \in \{0, 1\} \), as desired.

\[\square\]

2 BBS Generator

We now give a PRSG due to Blum, Blum, and Shub for which the problem distinguishing its outputs from the uniform distribution is closely related to the difficulty of determining whether a number with Jacobi symbol 1 is a quadratic residue modulo a certain kind of composite number called a Blum integer. The latter problem is believed to be computationally hard. First some background.

A Blum prime is a prime number \( p \) such that \( p \equiv 3 \pmod{4} \). A Blum integer is a number \( n = pq \), where \( p \) and \( q \) are Blum primes. Blum primes and Blum integers have the important property that every quadratic residue \( a \) has a square root \( y \) which is itself a quadratic residue. We call such a \( y \) a principal square root of \( a \) and denote it by \( \sqrt{a} \).
Lemma 4 Let \( p \) be a Blum prime, and let \( a \) be a quadratic residue modulo \( p \). Then \( y = a^{(p+1)/4} \mod p \) is a principal square root of \( a \) modulo \( p \).

Proof: We must show that, modulo \( p \), \( y \) is a square root of \( a \) and \( y \) is a quadratic residue. By the Euler criterion [Theorem 2, handout 15], since \( a \) is a quadratic residue modulo \( p \), we have \( a^{(p-1)/2} \equiv 1 \mod p \). Hence, \( y^2 = (a^{(p+1)/4})^2 \equiv a a^{(p-1)/2} \equiv a \mod p \), so \( y \) is a square root of \( a \) modulo \( p \). Applying the Euler criterion now to \( y \), we have

\[
y^{(p-1)/2} \equiv (a^{(p+1)/4})^{(p-1)/2} \equiv (a^{(p-1)/2})^{(p+1)/4} \equiv 1^{(p+1)/4} \equiv 1 \mod p.
\]

Hence, \( y \) is a quadratic residue modulo \( p \). \qed

Theorem 5 Let \( n = pq \) be a Blum integer, and let \( a \) be a quadratic residue modulo \( n \). Then \( a \) has four square roots modulo \( n \), exactly one of which is a principal square root.

Proof: By Lemma 4, \( a \) has a principal square root \( u \) modulo \( p \) and a principal square root \( v \) modulo \( q \). Using the Chinese remainder theorem, we can find \( x \) that solves the equations

\[
x \equiv \pm u \pmod p \\
x \equiv \pm v \pmod q
\]

for each of the four choices of signs in the two equations, yielding 4 square roots of \( a \) modulo \( n \). It is easily shown that the \( x \) that results from the +, + choice is a quadratic residue modulo \( n \), and the others are not. \qed

From Theorem 4 it follows that the mapping \( b \mapsto b^2 \pmod n \) is a bijection from the set of quadratic residues modulo \( n \) onto itself. (A bijection is a function that is 1–1 and onto.)

Definition: The Blum-Blum-Shub generator BBS is defined by a Blum integer \( n = pq \) and an integer \( \ell \). It is a \( (\mathbb{Z}_n^*, \ell) \)-PRSG defined as follows: Given a seed \( s_0 \in \mathbb{Z}_n^* \), we define a sequence \( s_1, s_2, s_3, \ldots, s_\ell \), where \( s_i = s_{i-1}^2 \pmod n \) for \( i = 1, \ldots, \ell \). The \( \ell \)-bit output sequence is \( b_1, b_2, b_3, \ldots, b_\ell \), where \( b_i = s_i \pmod 2 \).

Note that any \( s_m \) uniquely determines the entire sequence \( s_1, \ldots, s_\ell \) and corresponding output bits. Clearly, \( s_m \) determines \( s_{m+1} \) since \( s_{m+1} = s_m^2 \pmod n \). But likewise, \( s_m \) determines \( s_{m-1} \) since \( s_{m-1} = \sqrt{s_m} \), the principal square root of \( s_m \) modulo \( n \), which is unique by Theorem 5.

3 Security of BBS

Theorem 6 Suppose there is a probabilistic algorithm \( A \) that \( \epsilon \)-distinguishes \( \text{BBS}(\mathbb{Z}_n^*) \) from \( U \). Then there is a probabilistic algorithm \( Q(x) \) that correctly determines with probability at least \( \epsilon' = \epsilon/\ell \) whether or not an input \( x \in \mathbb{Z}_n^* \) with Jacobi symbol \( \left( \frac{x}{n} \right) = 1 \) is a quadratic residue modulo \( n \).

Proof: From \( A \), one easily constructs an algorithm \( \hat{A} \) that reverses its input and then applies \( A \). \( \hat{A} \) \( \epsilon \)-distinguishes the reverse of \( \text{BBS}(\mathbb{Z}_n^*) \) from \( U \). By Theorem 3 there is an \( \epsilon' \)-next bit predictor \( N_m \) for bit \( \ell - m + 1 \) of \( \text{BBS} \) reversed. Thus, \( N_m(b_\ell, b_{\ell-1}, \ldots, b_{m+1}) \) correctly outputs \( b_m \) with probability at least \( 1/2 + \epsilon' \), where \( (b_1, \ldots, b_\ell) \) is the (unreversed) output from \( \text{BBS}(\mathbb{Z}_n^*) \).
We now describe algorithm $Q(x)$, assuming $x \in \mathbb{Z}_n^*$ and $(\frac{x}{n}) = 1$. Using $x$ as a seed, compute $(b_1, \ldots, b_\ell) = BBS(x)$ and let $b = N_m(b_{\ell-m}, b_{\ell-m-1}, \ldots, b_1)$. Output “quadratic residue” if $b = x \mod 2$ and “non-residue” otherwise.

To see that this works, observe first that $N_m(b_{\ell-m}, b_{\ell-m-1}, \ldots, b_1)$ correctly predicts $b_0$ with probability at least $\frac{1}{2} + \epsilon'$, where $b_0 = (\sqrt{x^2} \mod n) \mod 2$. This is because we could in principle let $s_{m+1} = x^2 \mod n$ and then work backwards defining $s_m = \sqrt{s_{m+1}} \mod n$, $s_{m-1} = s_m \mod n$, ..., $s_0 = \sqrt{s_1} \mod n$. It follows that $b_0, \ldots, b_{\ell-m}$ are the last $\ell - m + 1$ bits of $BBS(s_0)$, and $b_0$ is the bit predicted by $N_m$.

Now, $x$ and $-x$ are clearly square roots of $s_{m+1}$. We show that they both have Jacobi symbol 1. Since $(\frac{x}{n}) = (\frac{\sqrt{x}}{p}) \cdot (\frac{\sqrt{x}}{q}) = 1$, then either $(\frac{x}{p}) = (\frac{\sqrt{x}}{q}) = 1$ or $(\frac{x}{p}) = (\frac{\sqrt{x}}{q}) = -1$. But because $p$ and $q$ are Blum primes, $-1$ is a quadratic non-residue modulo both $p$ and $q$, so $(\frac{-1}{p}) = (\frac{-1}{q}) = -1$. It follows that $(\frac{x}{n}) = 1$. Hence, $x = \pm \sqrt{s_{m+1}}$, so exactly one of $x$ and $-x$ is a quadratic residue.

Since $n$ is odd, $x \mod n$ and $-x \mod n$ have opposite parity. Hence, $x$ is a quadratic residue iff $x$ and $\sqrt{s_{m+1}}$ have the same parity. But $N_m$ outputs $\sqrt{s_{m+1}} \mod 2$ with probability $1/2 + \epsilon'$, so it follows that $Q$ correctly determines the quadratic residuosity of its argument with probability $1/2 + \epsilon'$.