# CS557: Cryptography

## Pseudo Random number Generator (PRNG)-II

S. Tripathy
IIT Patna

# Random number

- **Truly random** - is defined as exhibiting ``true'' randomness, such as
  - Noise in electrical circuits
  - Radio active decay.

- **Pseudorandom** - is defined as having the *appearance* of randomness, but nevertheless exhibiting a specific, repeatable pattern.
  - numbers calculated by a computer through a deterministic process, cannot, by definition, be random

# (Desirable) Properties of Pseudorandom Numbers

- **Uncorrelated Sequences -** The sequences of random numbers should be serially *uncorrelated*

- **Long Period** - The generator should be of *long period* (ideally, the generator should not repeat; practically, the repetition should occur only after the generation of a very large set of random numbers).

- **Uniformity** - The sequence of random numbers should be uniform, and unbiased. That is, equal fractions of random numbers should fall into equal ``areas'' in space.
  - Eg. if random numbers on [0,1) are to be generated, it would be poor practice were more than half to fall into [0, 0.1), presuming the sample size is sufficiently large.

- **Efficiency** - The generator should be efficient. Low overhead for massively parallel computations.

# Cryptographically Secure PRNG

- Indistinguishability
  - Uniform distribution
    - distribution $\mathcal{D}$ over strings of length-$\ell$ is pseudorandom if it is indistinguishable from a random distribution
    - Definition. We say an algorithm G, which on input of length n outputs a string of length $\ell(n)$, is a pseudorandom generator if
      1. For every n, $\ell(n) > n$
      2. For each PPT distinguisher D, there exists a negligible function negl such that $|Pr[D(r)=1 - Pr[D(G(s))=1| \leq negl(n)$

      Where r is chosen at uniformly random from $\{0,1\}^{\ell(n)}$ and s is chosen at uniform random from $\{0,1\}^s$

- Next-bit prediction

  Given N bits of the pseudo-random sequence, predict $(N+1)^{st}$ bit

  Probability of correct prediction should be very close to 1/2 for any efficient adversarial algorithm

# Classes of Attacks on PRNGs

- Direct Cryptanalytic Attack:
  - When the attacker can directly distinguish between PRNG numbers and random numbers (cryptanalyze the PRNG).
- Input Based Attack:
  - When the attacker is able to use knowledge and/ or control of PRNG inputs to  cryptanalyze the PRNG.
- State Compromise Extension Attacks:
  - When the attacker can guess some information due to an earlier breach of security. The advantage of a previous attack is extended.

# Direct Cryptanalytic Attacks

- When the attacker can directly cryptanalyze the PRNG.
  - Applicable to most PRNGs
  - Not applicable when the attacker is not able to directly see the output of the PRNG.
    - Eg A PRNG used to generate triple-DES keys. Here the output of the PRNG is never directly seen by an attacker.

# Input Based Attacks

- When an attacker used knowledge or control of the inputs to cyptanalyze the PRNG output.
- Types:
  - Known Input
    - If the inputs to the PRNG, that are designed to be difficult for a user to guess, turn out to be easily deducible.
      - Eg disk latency time. When the user is accessing a network disk, the attacker can observe the latency time.
  - Chosen input
    - Practical against smartcards, applications that feed incoming messages (username/password etc) to the PRNG as entropy samples.
  - Replayed Input
    - Similar to chosen input, except it requires less sophistication on the part of the attacker.

# State Compromise Extension Attacks

- These attacks are classified as:
  - Backtracking attacks
    - Uses the compromise of PRNG state S to learn about all previous PRNG outputs.
  - Permanent compromise attack
    - Once S has been compromised, all future and past outputs of the PRNG are vulnerable.
  - Iterative guessing attacks
    - Uses the knowledge of state S that was compromised at time t and the intervening PRNG outputs to guess the state S' at time $t+\Delta$.
  - Meet-in-the-middle attacks
    - Combination of iterative guessing and backtracking.

# Tests for Randomness in Random Numbers:

- Goal: To ensure that the random number generator produces a random stream. !
- Use of tests !
  - Passing a test is necessary but not sufficient
    - Pass ≠ Good
    - Fail ⇒ Bad
- Golomb's Postulates: Discussed
- Quantitative tests:
  - $X^2$ tests:
  - Lagged Correlation:
- Qualitative tests:
  - Scatter Plots
    - Plot pairs of random numbers.
      - Clumps of numbers, gaps and patterns are easily visible.

# $X^2$ (Chi$^2$)tests:

- Measure how well the presumed distribution (usually uniform) is represented.

- Algorithm for the test:
  - Divide the whole interval, within which the random number would be into finite number of bins (class intervals). Assume they have same size.

  - Count the number of random numbers within each interval and calculate the "expected" number of observations

  - Calculate: $D = X^2 = \Sigma(i=1,m)(\text{observed}_i - \text{expected}_i)^2 / (\text{expected}_i)$

  - The value of D determines if the numbers generated represent a chosen distribution, by looking up in a table, some critical values of Chi$^2$[1-α; k-1] Pass with confidence α if D is less.

# 5 Basic Tests for Random Numbers

1. ***Frequency test (Monobit)***. Uses the tests like chi-square test to compare the distribution of the set of numbers generated to a uniform distribution.

- Let $n_0$, $n_1$ denote the number of 0's and 1's in s, respectively. The statistic used is

$$X1 = (n_0 - n_1)^2 \ / \ n$$

which approximately follows a Chi$^2$ distribution with 1 degree of freedom

# 2.Serial test (2-BIT )

- Determine whether the number of occurrences of 00,01,10 and 11
  - as subsequences of s are approximately same, as would be expected for a random sequence.
- Let $n_0$, $n_1$ denote the number of 0 's and 1 's in s, respectively, and let $n_{00}$,…, $n_{11}$ etc., denote the number of occurrences of 00,01,10,11 in s respectively.
- Note that
  - $n_{00} + n_{01} + n_{10} + n_{11} = (n-1)$
  - since the subsequences are allowed to overlap

The statistic used is

$$X_2 = \frac{4}{n-1}\left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 + n_1^2\right) + 1$$

which approximately follows a $\chi^2$ distribution with 2 degrees of freedom

# 3. Poker test

- Let *m* be a positive integer such that . and let
  and let, $k = \lfloor n/m \rfloor$.
- Divide the sequence *s* into *k* non-overlapping parts each of length *m* and let $n_i$ be the no. of occurrences of the $i^{th}$ type of sequence of length *m*, $1 \leq i \leq 2^m$. $\lfloor \frac{n}{m} \rfloor \geq 5 \cdot (2^m)$

- The poker test determines whether the sequences of length *m*
  - each appear approximately the same number of times in *s*, as would be expected for a random sequence. The statistic used is

$$X_3 - \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k$$

- which approximately follows a $\chi^2$ distribution with $2^m - 1$ degrees of freedom.

# 4. Runs test

- Tests the runs up and down or the runs above and below the mean by comparing the actual values to expected values.
  - The statistic for comparison is the chi-square.
- The expected number of gaps (or blocks) of length *i* in a random sequence of length n is $e_i = (n - i + 3)/2^{i+2}$.
- Let *k* be equal to the largest integer *i* for which $e_i \geq 5$
- Let $B_i$, $G_i$ be the number of blocks and gaps, respectively, of length *i* in s for each *i*, $1 \leq i \leq k$.
- The statistic used is

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i}$$

which approximately follows a $\chi^2$ distribution with 2k-2 degrees of freedom.

# 5. Autocorrelation test

- Tests the correlation between numbers and compares the sample correlation to the expected correlation of zero.
- check for correlations between the sequence *s* and (non-cyclic) shifted versions of it.
- Let *d* be a fixed integer, $1 \leq d \leq \lfloor n/2 \rfloor$.

- The number of bits in *s* not equal to their *d*-shifts is

$$A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}$$

- The statistic used is

$$X_5 = 2\left(A(d) - \frac{n-d}{2}\right) / \sqrt{n-d}$$

- which approximately follows an $N(0,1)$ distribution if $n-d \geq 10$.

# Ex.:

S=

11100 01100 01000 10100 11101 11100 10010 01001
11100 01100 01000 10100 11101 11100 10010 01001
11100 01100 01000 10100 11101 11100 10010 01001
11100 01100 01000 10100 11101 11100 10010 01001

n=160

| | $\alpha$ | | | | | |
|---|---|---|---|---|---|---|
| $n$ | 0.100 | 0.050 | 0.025 | 0.010 | 0.005 | 0.001 |
| 1 | 2.7055 | 3.8415 | 5.0239 | 6.6349 | 7.8794 | 10.8276 |
| 2 | 4.6052 | 5.9915 | 7.3778 | 9.2103 | 10.5966 | 13.8155 |
| 3 | 6.2514 | 7.8147 | 9.3484 | 11.3449 | 12.8382 | 16.2662 |
| 4 | 7.7794 | 9.4877 | 11.1433 | 13.2767 | 14.8603 | 18.4668 |
| 5 | 9.2364 | 11.0705 | 12.8325 | 15.0863 | 16.7496 | 20.5150 |
| 6 | 10.6446 | 12.5916 | 14.4494 | 16.8119 | 18.5476 | 22.4577 |
| 7 | 12.0170 | 14.0671 | 16.0128 | 18.4753 | 20.2777 | 24.3219 |
| 8 | 13.3616 | 15.5073 | 17.5345 | 20.0902 | 21.9550 | 26.1245 |
| 9 | 14.6837 | 16.9190 | 19.0228 | 21.6660 | 23.5894 | 27.8772 |
| 10 | 15.9872 | 18.3070 | 20.4832 | 23.2093 | 25.1882 | 29.5883 |
| 11 | 17.2750 | 19.6751 | 21.9200 | 24.7250 | 26.7568 | 31.2641 |
| 12 | 18.5493 | 21.0261 | 23.3367 | 26.2170 | 28.2995 | 32.9095 |
| 13 | 19.8119 | 22.3620 | 24.7356 | 27.6882 | 29.8195 | 34.5282 |
| 14 | 21.0641 | 23.6848 | 26.1189 | 29.1412 | 31.3193 | 36.1233 |
| 15 | 22.3071 | 24.9958 | 27.4884 | 30.5779 | 32.8013 | 37.6973 |
| 16 | 23.5418 | 26.2962 | 28.8454 | 31.9999 | 34.2672 | 39.2524 |
| 17 | 24.7690 | 27.5871 | 30.1910 | 33.4087 | 35.7185 | 40.7902 |
| 18 | 25.9894 | 28.8693 | 31.5264 | 34.8053 | 37.1565 | 42.3124 |
| 19 | 27.2036 | 30.1435 | 32.8523 | 36.1909 | 38.5823 | 43.8202 |
| 20 | 28.4120 | 31.4104 | 34.1696 | 37.5662 | 39.9968 | 45.3147 |
| 21 | 29.6151 | 32.6706 | 35.4789 | 38.9322 | 41.4011 | 46.7970 |
| 22 | 30.8133 | 33.9244 | 36.7807 | 40.2894 | 42.7957 | 48.2679 |
| 23 | 32.0069 | 35.1725 | 38.0756 | 41.6384 | 44.1813 | 49.7282 |
| 24 | 33.1962 | 36.4150 | 39.3641 | 42.9798 | 45.5585 | 51.1786 |
| 25 | 34.3816 | 37.6525 | 40.6465 | 44.3141 | 46.9279 | 52.6197 |
| 26 | 35.5632 | 38.8851 | 41.9232 | 45.6417 | 48.2899 | 54.0520 |
| 27 | 36.7412 | 40.1133 | 43.1945 | 46.9629 | 49.6449 | 55.4760 |
| 28 | 37.9159 | 41.3371 | 44.4608 | 48.2782 | 50.9934 | 56.8923 |
| 29 | 39.0875 | 42.5570 | 45.7223 | 49.5879 | 52.3356 | 58.3012 |
| 30 | 40.2560 | 43.7730 | 46.9792 | 50.8922 | 53.6720 | 59.7031 |
| 31 | 41.4217 | 44.9853 | 48.2319 | 52.1914 | 55.0027 | 61.0983 |
| 63 | 77.7454 | 82.5287 | 86.8296 | 92.0100 | 95.6493 | 103.4424 |
| 127 | 147.8048 | 154.3015 | 160.0858 | 166.9874 | 171.7961 | 181.9930 |
| 255 | 284.3359 | 293.2478 | 301.1250 | 310.4574 | 316.9194 | 330.5197 |
| 511 | 552.3739 | 564.6961 | 575.5298 | 588.2978 | 597.0978 | 615.5149 |
| 1023 | 1081.3794 | 1098.5208 | 1113.5334 | 1131.1587 | 1143.2653 | 1168.4972 |

| $\alpha$ | 0.1 | 0.05 | 0.025 | 0.01 | 0.005 | 0.0025 | 0.001 | 0.0005 |
|---|---|---|---|---|---|---|---|---|
| $x$ | 1.2816 | 1.6449 | 1.9600 | 2.3263 | 2.5758 | 2.8070 | 3.0902 | 3.2905 |

- Thanks