Outline

- What is a block cipher
- Test the randomness of a block cipher
  - Popular test suites: Nist, Dieharder
  - Problems: multiple outputs, modes-of-operation dependent
- Alternative approach: Crytostat
  - Modes-of-operation independent
  - Bayes test aggregates multiple outputs
  - Empirical results
    1. When the null is true
    2. When the alternative is true
    3. Dependence on sample size
    4. Sensitivity to prior distribution
- Conclusions and discussions

Tests for Randomness with NIST Test Suites

- Random number generators (RNGs) or pseudorandom number generators (PRNGs) generate arbitrarily-long binary sequences.
- NIST test suite consists of 15 tests to detect deviation of these sequences from randomness, e.g., Frequency test, the Runs test.
- Each test produces a set of P-values corresponding to the set of sequences.
  Two ways to interpret the P-values:
  - whether the proportion of P-values $\geq \alpha$ (the significance level) is within the range of acceptable proportions.
  - Whether the P-values $\sim$Uniform (0,1).

Bayesian Approach: Crytostat

NIST Test Suite does not work well for testing the randomness of a block cipher:
1. A block cipher has to be turned into a PRNG and bias could be introduced in this prodcess.
2. Difficulty with interpreting a multitude of P-values.

Crypotat uses Bayes test which
1. can test the input-to-output mapping directly and
2. combine multiple test results easily.

Bayesian Hypothesis Test

Let H_0, H_1 denote the null and alternative hypothesis and D denote sample data.
Then $P(H_i|D) = \frac{P(D|H_i)P(H_i)}{P(D)}, i = 0, 1.$
Assuming $P(H_0) = P(H_1)$, H_0 and H_1 can be compared via the Bayes factor
$K = \frac{P(D|H_0)}{P(D|H_1)}.$

Note for independent data samples $D_1, D_2, \cdots D_m$,
$K = \frac{P(D_1|H_0)P(D_2|H_0)\cdots P(D_m|H_0)}{P(D_1|H_1)P(D_2|H_1)\cdots P(D_m|H_1)} = K_1 K_2 \cdots K_m$
or $\log K = \sum_{i=1}^{m} \log K_i$ where K_i is the Bayes facor based on sample D_i.
This allows aggregation of multiple test results.

Cryptostat Output Data

Cryptostat derives test data series from output data series, partitions the test data bit positions into disjoint bit groups and test randomness on each bit group. The bit group values are integers in $(0, 2^b - 1)$ where b is the bit group size.

H_0 : the big group values are uniformlly distributed.

Crytostat uses run tests and noncolliding block tests to test uniformity which can be turned into testing if a discrete random variable X follows a certain distribution $p(x)$.

Bayes test

Through a Kolmogorov-Smirnov type operation, the test can be turned into a binomial test
$H_0 : p = p_0$
$H_1 : p \neq p_0$
with k successes out of n trials.
Cryptostat uses $P \sim U(0, 1)$ to specify H_1 and derives the Bayes factor as
$K = \frac{\Gamma(n+2)}{\Gamma(k+1)\Gamma(n-k+1)} p_0^k (1 - p_0)^{n-k}$.