Statistical Analysis of the 3WAY Block Cipher

By

Himanshu Kale

Project Report Submitted
In
Partial Fulfilment of the Requirements for the Degree of Master of Science In Computer Science

Supervised by

Professor Alan Kaminsky

Department of Computer Science
B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

December, 2013
Abstract

A block cipher is supposed to produce outputs that look random. There are statistical test suites like the NIST test suite that have been used for the purpose of evaluating the randomness of a block cipher’s mapping. But the NIST test suite is suited for evaluating binary sequences and it essentially treats the block cipher as a Pseudo Random Number Generator (PRNG). The Input-Output Independence Test uses a Bayesian approach which takes into account the randomness of the block cipher’s mapping. 3WAY was the block cipher chosen for performing statistical analysis using the Input-Output Independence Test. It uses a 96-bit block size, 96-bit key size and consists of a total of 11 rounds. The test is applied to all rounds of 3WAY along with a suite of 5 other tests which also use a Bayesian approach. In the end, an analysis of the results is presented which determines the randomness margin of 3WAY.
# Table of Contents

1. Introduction........................................................................................................................ 1
2. Background ........................................................................................................................ 2
   2.1 Bayes Factor and Posterior Odds Ratio................................................................. 2
   2.2 Coincidence Test...................................................................................................... 3
3. System ................................................................................................................................ 3
   3.1 3WAY Block Cipher .............................................................................................. 4
      3.1.1 Mu Function ($\mu$) .......................................................................................... 4
      3.1.2 Gamma Function ($\gamma$) ................................................................................ 4
      3.1.3 Theta Function ($\theta$) .................................................................................... 6
      3.1.4 Pi1 and Pi2 Functions ($\pi_1$ and $\pi_2$).......................................................... 6
      3.1.5 Rho Function ($\rho$) ....................................................................................... 6
      3.1.6 Key Schedule .................................................................................................. 8
      3.1.7 Encryption ...................................................................................................... 8
   3.2 Input-Output Independence Test ............................................................................. 9
4. Results .............................................................................................................................. 10
5. Conclusions ..................................................................................................................... 12
6. Future Work .................................................................................................................... 12
7. References....................................................................................................................... 12
1. Introduction

A block cipher when supplied with a plaintext and key as inputs is supposed to produce a ciphertext (output) that looks completely random and bears no correlation with the input supplied. Many statistical test suites have been used for evaluating the randomness of a block cipher’s mapping from (plaintext, key) to ciphertext. The NIST test suite is one such suite which has been used for evaluating cryptographic functions.

The NIST test suite however is more suited to assessing the randomness of binary sequences and it does not take into account whether the sequence was produced by a block cipher or some other source. It essentially treats the block cipher as a Pseudo Random Number Generator (PRNG). The cipher may be used in a particular mode of operation such as Cipher Block Chaining (CBC) to make it behave as a PRNG.

A statistical test which takes into account the randomness of the block cipher’s mapping would be more suitable as it would not need a block cipher to behave as a PRNG. The Input-Output Independence test uses a Bayesian statistical testing methodology for making a decision whether a block cipher’s mapping is random or non-random. For a given set of plaintext, keys and the corresponding ciphertext, this test compares the input (plaintext, key) and output (ciphertext) to check whether they behave independent of each other.

As part of the project, the 3WAY block cipher was chosen and implemented in Java for performing statistical analysis. The Input-Output Independence test will be tested on 3WAY along with a collection of 5 other tests to determine the randomness of the cipher’s mapping. All of these tests use a Bayesian approach for evaluating the randomness of the cipher’s mapping. The results of all tests will be analysed to determine the randomness margin of 3WAY. The randomness margin requires finding the number of rounds for which the cipher produces random outputs. This is then compared with the total number of rounds used by the cipher to yield the randomness margin.

The rest of the report is structured as follows. Section 2 covers the background work done by Prof. Kaminsky on developing the Coincidence test using a Bayesian approach. The Input-Output independence test has been designed similar to this test. Section 3 describes the structure of the 3WAY cipher and also the structure of the Input-Output Independence test. Section 4 discusses the results produced by running the entire suite of tests including the Input-Output independence test on 3WAY. The conclusions and outcomes of this project are discussed in Section 5. Section 6 lists the possible future work that could be done.
2. Background

Prof. Kaminsky has described a statistical test called the Coincidence test in his paper [1]. In this paper he has proposed using the Bayesian approach for designing a statistical test which would yield a clear random/non-random decision regarding a cipher’s mapping. It also describes the Bayes Factor and the posterior odds ratio which is the value that is used for making the random/non-random decision.

2.1 Bayes Factor and Posterior Odds Ratio

If \( H \) is a hypothesis for an experiment and \( D \) is the data sample that is collected after performing the experiment then according to the Bayes Theorem the conditional probability of the hypothesis given the data sample is

\[
pr(H|D) = \frac{pr(D|H) \cdot pr(H)}{pr(D)}
\]

If we consider two alternative hypotheses \( H_1 \) and \( H_2 \) and \( D \) is still the data sample collected from the experiment, the posterior odds ratio can be calculated as

\[
\frac{pr(H_1|D)}{pr(H_2|D)} = \frac{pr(D|H_1)}{pr(D|H_2)} \cdot \frac{pr(H_1)}{pr(H_2)}
\]

Here the term \( pr(H_1)/pr(H_2) \) is the prior odds ratio and the term \( pr(D|H_1)/pr(D|H_2) \) is the Bayes factor. It can also be stated as posterior odds ratio = Bayes Factor x prior odds ratio. The prior odds ratio is the preliminary assumption that we have made and the Bayes factor is then used to update the assumption we have made after getting a data sample \( D \) from the experiment. If two experiments are performed then we will obtain two data samples \( D_1 \) and \( D_2 \). The posterior odds ratio taking into account both the samples can be calculated as

\[
\frac{pr(H_1|D_2, D_1)}{pr(H_2|D_2, D_1)} = \frac{pr(D_2|H_1)}{pr(D_2|H_2)} \cdot \frac{pr(H_1|D_1)}{pr(H_2|D_1)}
\]

Here the second term is the posterior odds ratio from the first experiment’s data sample \( D_1 \). This is used as the prior odds ratio of the second experiment. This equation can be simplified using the second equation as follows

\[
\frac{pr(H_1|D_2, D_1)}{pr(H_2|D_2, D_1)} = \frac{pr(D_2|H_1)}{pr(D_2|H_2)} \cdot \frac{pr(H_1|D_1)}{pr(D_1|H_2)} \cdot \frac{pr(H_1)}{pr(H_2)}
\]
This equation is used to calculate the posterior odds ratio for N number of samples. The final posterior odds ratio is obtained by multiplying the prior odds ratio of the initial assumption with the N number of Bayes factors obtained by performing N experiments.

### 2.2 Coincidence Test

The coincidence test was tested on two block ciphers (PRESENT and IDEA) and two Message Authentication Codes (SipHash and SQUASH). It was run on a cluster with multiple trials being performed in parallel to speed up the process. The diagram below illustrates how the test is performed.

![Coincidence Test Diagram](image)

**Figure 1**: Coincidence Test [1]

The ciphertext C is computed by encrypting the plaintext P with key K. V is an arbitrarily chosen output which is then compared with C. Bit groups of a certain size are selected from both C and V which are then checked for equality. A coincidence is said to have occurred when each bit from C is equal to the corresponding bit from V.

The data samples are obtained by choosing V, P and K at random. Multiple trials of the test are run to get an accurate picture of the cipher’s randomness. For each of the trials the test selects multiple adjacent bit groups in sizes of 1, 2, 4, 8 and so on in powers of 2.

### 3. System

The system includes an implementation of the 3WAY cipher and the Input-Output independence test in Java. Both the classes are integrated into Prof. Kaminsky’s CryptoStat Library. The library consists of 5 other tests which are also performed on 3WAY. They are
the Uniformity test, Ciphertext independence test, Complement test, Strong Avalanche test and Non-Linearity test. All of them use a Bayesian approach for performing statistical analysis.

Multiple trials of each of the tests can be computed in parallel to enable a large number of trials to be executed. The final program was run on the Tardis cluster which has ten quad-core nodes. CryptoStat makes use of the Parallel Java library also developed by Prof. Kaminsky to enable trials of each test to be executed in parallel.

Each test produces a log odds ratio for each round of the 3WAY cipher.

3.1 3WAY Block Cipher

3WAY is a block cipher which has a 96-bit block size and a 96-bit key size. It consists of a total of 11 rounds where each round involves computing a round function $\rho$. The cipher was designed specifically with 32-bit computation in mind and each of the cipher’s functions operates on 32-bit data blocks. The multiple functions used in this cipher are described below

3.1.1 Mu Function ($\mu$)

Mu is a function that reverses the order of the input bits. The operation can be specified as: $A \circ \mu \circ \mu = A$. Mu is required for inverting the gamma and theta functions when decrypting: $\theta^{-1} = \mu \circ \theta \circ \mu$ and $\gamma^{-1} = \mu \circ \gamma \circ \mu$. The inverted key for decryption is also obtained using this function: $K^{-1} = K \circ \theta \circ \mu$.

3.1.2 Gamma Function ($\gamma$)

Gamma is a non-linear 3-bit S-box function. The function used in the S-box is:

$$b_i = \overline{a}_i \text{ (XOR)} \overline{a}_{i+k}a_{i+2k}$$

Here $a$ is the input vector, $b$ is the output vector, $i$ is the position in the vector, $n$ is the length of vector $a$ and $k = n/3$. All of the vector positions are modulo $n$. The 96-bit bock is processed to produce 32 bits of output in parallel at one time.
Figure 2: Overview of Gamma Function

Figure 3: S-box of the Gamma Function

The S-box has been designed to run efficiently on a 32-bit machine. The table of the S-box is as follows

Table 1: Gamma Function’s S-box Output

```
<table>
<thead>
<tr>
<th>Input Bits</th>
<th>Output Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1</td>
</tr>
<tr>
<td>001</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
</tr>
</tbody>
</table>
```
3.1.3 **Theta Function (θ)**

Theta is a linear substitution operation that operates on inputs that have a multiple of 12 bits. There are two ways to describe it: either a finite field polynomial multiplication problem or an XOR matrix multiplication. For simplicity, the matrix multiplication will be used to describe Theta.

The rows of the matrix represent 12 inputs and the columns of the matrix represent 12 outputs. Each of the inputs and outputs are 8 bits, since the block size is 96-bit. For producing the outputs, XOR the inputs with a 1 within each output column.

![Theta Function’s Multiplication Matrix](image)

**Figure 4: Theta Function’s Multiplication Matrix**

3.1.4 **Pi1 and Pi2 Functions (π1 and π2)**

These two functions diffuse the bits between the functions. Pi1 performs a right rotation of 10 bits on the most significant 32-bit block and a left rotation of 1 bit on the least significant 32-bit block (no operation on the middle 32-bit block). Pi2 performs a left rotation of 1 bit on the most significant 32-bit block and a right rotation of 10 bits on the least significant 32-bit block (no operation on the middle 32-bit block). These are designed such that \( \pi_1 \circ \mu \circ \pi_2 = \mu \)

3.1.5 **Rho Function (ρ)**

Rho (ρ) is the round function for 3-Way. It executes the following functions in order on the input data bits: Theta, Pi1, Gamma and Pi2. This function is called once per round.
Figure 5: Round Function [3]
3.1.6 Key Schedule

3-Way’s key schedule is very simple. Every round’s subkey is the global key XORed with a round constant with a small hamming weight. The round constant is only 16 bits and is only XORed with the 7 first and last 16 bits of the global key. With 11 round 3-Way, the encryption round constants are the following, with one extra round constant for the subkey used after the last round:

0B0B  
1616  
2C2C  
5858  
B0B0  
7171  
E2E2  
D5D5  
BBBB  
6767  
CECE  
8D8D

If more than 11 rounds are used, the round constants loop back to the top. If less than 11 rounds are used, the remaining round constants are ignored. The decryption constants are as follows:

B1B1  
7373  
E6E6  
DDDD  
ABAB  
4747  
8E8E  
0D0D  
1A1A  
3434  
6868  
D0D0

3.1.7 Encryption

Encryption works by XORing the data block with the current round’s subkey and running the data block through Rho. This process is repeated for the number of rounds desired. After all of the rounds, one final subkey is XORed with the data block and the data is passed through Theta again.
3.2 Input-Output Independence Test

The Input-Output independence test consists of a number of subtests. Each of these subtests works on an input (plaintext, key) and output (ciphertext) from a series of randomly chosen plaintexts, keys and the corresponding ciphertexts produced by the 3WAY cipher.

The subtests examine bit groups of size 1, 2, 4, 8 and so on from the input and output. The bits for each bit group are chosen at random. Each subtest checks whether a particular output bit group is independent of particular like-sized input bit group. The test then XORs these like-sized bit groups. This XORed value is then checked for uniform distribution. If the value is uniformly distributed then it can be said to be random otherwise it is declared nonrandom.

The subtests compute the logarithm of the Bayes Factor for all samples of the input and output. The sum of all these individual Log Bayes Factors then yields the final odds ratio which determines whether the cipher is random or nonrandom.

The diagram below illustrates how the test works for a bit group of size 4:

![Diagram](image)

**Figure 6:** Input-Output Independence Test.

The number of bit groups that can be chosen for a given bit group size is limited by an external parameter. The Input-Output Independence test class implements the Test interface from the CryptoStat library. The Test interface is an interface which specifies how a statistical test is to be implemented.
4. Results

The entire test suite was run on the Tardis Cluster and took 44 hours to complete for the 3WAY cipher. 8 worker tasks were specified which allowed 8 trials to be computed in parallel. A total of 200 trials were performed with a 1000 random input and output samples in each trial. The number of random bit groups that can be chosen for each subtest was 50. The table below summarizes the log odds ratio that were obtained.

<table>
<thead>
<tr>
<th>Rounds</th>
<th>Log Odds Ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Uniformity Test</td>
</tr>
<tr>
<td>1</td>
<td>1.44769e+02</td>
</tr>
<tr>
<td>2</td>
<td>3.70829e+02</td>
</tr>
<tr>
<td>3</td>
<td>3.71164e+02</td>
</tr>
<tr>
<td>4</td>
<td>3.77755e+02</td>
</tr>
<tr>
<td>5</td>
<td>3.67992e+02</td>
</tr>
<tr>
<td>6</td>
<td>3.48532e+02</td>
</tr>
<tr>
<td>7</td>
<td>3.60773e+02</td>
</tr>
<tr>
<td>8</td>
<td>3.72978e+02</td>
</tr>
<tr>
<td>9</td>
<td>3.64698e+02</td>
</tr>
<tr>
<td>10</td>
<td>3.73738e+02</td>
</tr>
<tr>
<td>11</td>
<td>3.71146e+02</td>
</tr>
</tbody>
</table>
The graphs below show the maximum number of non-random rounds and the randomness margin for 3WAY according to each of the 6 tests.

**Nonrandom rounds for 3WAY block cipher**

<table>
<thead>
<tr>
<th>Statistical Test</th>
<th>Number of Rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniformity</td>
<td>0</td>
</tr>
<tr>
<td>Ciphertext...</td>
<td>0</td>
</tr>
<tr>
<td>Complement</td>
<td>2</td>
</tr>
<tr>
<td>Input-Output...</td>
<td>0</td>
</tr>
<tr>
<td>Strong Avalanche</td>
<td>2</td>
</tr>
<tr>
<td>Non-Linearity</td>
<td>1</td>
</tr>
</tbody>
</table>

**Randomness margin for 3WAY block cipher**

<table>
<thead>
<tr>
<th>Statistical Test</th>
<th>Randomness Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniformity</td>
<td>1</td>
</tr>
<tr>
<td>Ciphertext Independence</td>
<td>1</td>
</tr>
<tr>
<td>Complement</td>
<td>0.82</td>
</tr>
<tr>
<td>Input-Output Independence</td>
<td>1</td>
</tr>
<tr>
<td>Strong Avalanche</td>
<td>0.82</td>
</tr>
<tr>
<td>Non-Linearity</td>
<td>0.91</td>
</tr>
</tbody>
</table>
5. Conclusions

- As seen from the first Chart in the results, from round 3 onwards the 3WAY cipher passed in all 6 tests.
- This gives it a randomness margin of 0.82 considering the worst case. This is calculated as $1 - \frac{r}{R}$. $r$ is the largest number of nonrandom rounds detected and $R$ is the total number of rounds in the cipher.

6. Future Work

- The input-output independence test could be modified to run on graphics processing units (GPU) which would allow a larger number of trials to be evaluated.
- Additional tests could be designed and added to the suite for even more comprehensive analysis of the 3WAY cipher.

7. References

