

# On testing for randomness of a bit-stream and applications to cryptography

Rajeeva L Karandikar

EVP

Cranes Software International Ltd.

Bangalore

[rajeeva.karandikar@cranessoftware.com](mailto:rajeeva.karandikar@cranessoftware.com)

## Brief Introduction to Symmetric key cryptography :

A **plaintext or message** is a string of 0's and 1's of finite length, a **key** (same as password) is also a string of 0's and 1's of a fixed length, say  $k$ . For every key, the **encryption algorithm**  $\mathcal{E}$  transforms the message into another string of 0's and 1's - called **ciphertext**.

The **decryption algorithm**  $\mathcal{D}$  gives the reverse mapping - when the same key is used.

Let  $\mathcal{S}_n$  denote the space of all sequences of 0's and 1's of length  $n$  and let

$$\mathcal{S} = \bigcup_{n=1}^{\infty} \mathcal{S}_n.$$

Let  $\mathcal{K}$  denote  $\mathcal{S}_k$  for a fixed  $k$ .  $\mathcal{E}$  and  $\mathcal{D}$  satisfy

$$\mathcal{E} : \mathcal{S} \times \mathcal{K} \rightarrow \mathcal{S}$$

$$\mathcal{D} : \mathcal{S} \times \mathcal{K} \rightarrow \mathcal{S}$$

and

$$\mathcal{E}(M, K) = C \text{ and } \mathcal{D}(C, K) = T \Rightarrow T = M.$$

Usually, one requires that encryption does not increase the length of the message by a big margin. So we often require that for some fixed  $b$

$$\mathcal{E}(\mathcal{S}_n) \subseteq \mathcal{S}_{n+b} \quad \forall n \geq 1.$$

When  $A$  wants to send a secret communication  $M^*$  to  $B$ , they share a common key  $K^*$  (in an earlier meeting or via another safe channel) and then  $A$  generates

$$C^* = \mathcal{E}(M^*, K^*)$$

and sends  $C^*$  to  $B$ .

Since  $B$  knows the key used  $K^*$ ,  $B$  can obtain

$$M^* = \mathcal{D}(C^*, K^*).$$

If anyone else has intercepted the ciphertext  $C^*$ , he/she **may not** be able to extract  $M$  as he/she does not know  $K^*$ . It is assumed that the algorithms  $\mathcal{E}$ ,  $\mathcal{D}$  are known to the interceptor.

We would like to design the algorithm  $\mathcal{E}$  such that without knowing the key used  $K^*$ , an interceptor **cannot** extract  $M$ .

Here, the word cannot has to be properly understood. If the interceptor tries all possible keys  $K_1, K_2, \dots, K_m$  with  $m = 2^k$  and generates

$$T_i = \mathcal{D}(C^*, K_i) \quad i = 1, 2 \dots m$$

then one of the  $T_i$  is the message  $M$ .

So the best we could do is to ensure that trying all possible keys is the only possible way for an adversary to recover the message  $M$ .

Ideally, we would like to require that without knowing  $K^*$ ,  $C^*$  does not contain any information about  $M^*$ .

**Ideal requirement:** given that  $M \in \mathcal{S}_n$ ,

$$P(M = M_0 \mid C^*) = \frac{1}{\#\mathcal{S}_n} = 2^{-n} \quad \forall M_0 \in \mathcal{S}_n$$

i.e. given the ciphertext, all possible messages are equally likely.

However, given  $C^*$ , we know for sure that  $M$  belongs to

$$\mathcal{Y} = \{\mathcal{D}(C^*, K) : K \in \mathcal{K}\}$$

so the best we can hope for is:

$$P(M = M_0 \mid C^*) = \frac{1}{\#\mathcal{S}_k} = 2^{-k} \quad \forall M_0 \in \mathcal{Y}.$$

So we should choose keysize  $k$  such that  $\mathcal{S}_k$  is large enough and then desire that

$$P(M = M_0 \mid C^*) = \frac{1}{\#\mathcal{S}_k} = 2^{-k} \quad \forall M_0 \in \mathcal{Y}$$

is achieved. This amounts to saying that the best attack is the brute force attack: trying all possible cases.

$k = 128$  suffices for this purpose

We assume that once an interceptor sees a decrypted text, he/she can say if this is the message or not. In other words, if  $M$  was a paragraph written in english language, then in the set  $\mathcal{Y}$  it is unlikely that it contains any other meaningful english text. Seems reasonable when message size  $n$  is much larger than  $k$ .

Let us assume that the interceptor has huge computing power: he/she has one million machines which are very fast - 1000 GHz and that they have extremely efficient code so that in one clock cycle, it can be decided if the decrypted bit-stream is the message or not.

What is the chance that in 1000 years the brute force attack can **discover** the message?

No of seconds in 1000 years:

$$1000 \times 60 * 60 * 24 * 266 \sim 2^{35}$$

No of cases scanned in 1000 years (with one million machines each 1000 GHz)

$$2^{35} \times 2^{20} \times 2^{10} \times 2^{30} \sim 2^{95}$$

$$\text{Probability of discovery} = \frac{2^{95}}{2^{128}} \sim 2^{-33}.$$

Probability of discovering a message is approximately one in 8 billion!

If we are able to construct an algorithm so that the cipher text is statistically indistinguishable from the output of a random bit-stream generator, then it can be taken as an indication that the ciphertext is not leaking any information about the message.

In this case, given an algorithm, we can generate ciphertext of any required length and still we should not be able to distinguish the ciphertext from random bit stream.

This leads to an interesting problem.

Given a bit-stream  $B = x_1x_2 \dots x_m$ , of length  $m$  where each  $x_i$  is 0 or 1, we need to **test the Null Hypothesis** that these are observations from an iid Bernoulli sequence with  $p = \frac{1}{2}$  (henceforth referred to as random bit-stream).

Most standard tests are based on Central Limit Theorem and have low power since CLT is robust. Thus minor deviations from iid nature will go undetected.

Another test called Maurer's universal test is popular in crypto literature: It has a parameter  $L$ . It looks at the bit-stream in non-overlapping  $L$  bit blocks:

$$B = y_1 y_2 \dots y_p$$

where each  $y_i$  is  $L$  bits. If  $L = 8$ , then  $y_i$ 's are the Bytes.

Let  $G(y_i)$  be the gap since the last occurrence of the pattern  $y_i$  in the stream  $y_1y_2 \dots y_{i-1}$ , thus  $G(y_i) = t$  if  $y_j \neq y_i$  for  $i - t < j < i$  and  $y_{i-t} = y_i$ . (If the pattern has not occurred before, we define  $G(y_i)$  to be  $i$ .) The test statistics is

$$T = \frac{1}{p - q} \sum_{i=q}^p \log_2(G(y_i)).$$

Here we ignore an initial block of length  $q$  so that all patterns are likely to occur in the first  $q$  positions if the stream is a random bit-stream.

The distribution of  $T$  is asymptotically Normal (under the null-hypothesis) and the asymptotic mean and variance have been computed.

Recommended values  $q = 10 \times 2^L$  and  $p = 1010 \times 2^L$ .  
For  $L = 8$ , it means 256KB or 2048 Megabits size.

For  $L = 8$ ,  $q = 10 \times 256$  and  $p = 1010 \times 256$ , the mean is  $\mu = 7.1836656$  and standard deviation (for the chosen size of  $p - q$ ) is  $\sigma = 0.00217401$ .

$$Z = \frac{(T - \mu)}{\sigma}$$

(approximately) has standard normal distribution, if the bit-stream is random.

While my own testing of the Maurer's test (called Maurer's universal test in cryptography literature) suggests that the test has capability to identify streams that have some patterns, I am not aware of any study of power of this test against a specified class of alternatives.

We should look at stationary process taking values in  $\{0, 1\}$  with 8-dimensional marginals being uniform but not iid- perhaps some Markovian structure.

One kind of popular symmetric ciphers is Block Cipher. Here blocks of a fixed size -  $t$  bits- are transformed into a block of same size and the whole plaintext is broken into these blocks, the last block, if unfinished is padded with, say 0.

In this case, apart from the requirement that the ciphertext stream should be indistinguishable from random bit-stream, we make the same requirement on the following bit-streams:

$$B_1(M, K) = \mathcal{E}(M, K)$$

$$B_2(M, K) = M \oplus \mathcal{E}(M, K)$$

$$B_3(M, K) = \mathcal{E}(M, K) \oplus \mathcal{E}(M, K_1)$$

$$B_4(M, K) = \mathcal{D}(\mathcal{E}(M, K), K_1)$$

$$B_5(M, K) = \mathcal{E}(M, K) \oplus \mathcal{E}(M_1, K)$$

where  $K_1$  differs from  $K$  at exactly one bit and  $M_1$  differs from  $M$  at exactly one fixed location in each block.

$$B_6(M, K) = \mathcal{D}(M, K)$$

$$B_7(M, K) = M \oplus \mathcal{D}(M, K)$$

$$B_8(M, K) = \mathcal{D}(M, K) \oplus \mathcal{D}(M, K_1)$$

$$B_9(M, K) = \mathcal{E}(\mathcal{D}(M, K), K_1)$$

$$B_{10}(M, K) = \mathcal{D}(M, K) \oplus \mathcal{D}(M_1, K)$$

where  $K_1$  differs from  $K$  at exactly one bit and  $M_1$  differs from  $M$  at exactly one fixed location in each block.

**If any of these bit-streams is not random, has some patterns, the same can be exploited to launch an attack.**

Thus for every input stream  $M$  and key  $K$  we have 10-bit-streams  $B_i(M, K)$  and if the Blockcipher is to pass the randomness test, these bit-streams should be indistinguishable from a random bit-stream. Thus

$$X_i(M, K) = Z(B_i(M, K))$$

should be observations from  $N(0, 1)$  distributions.

Our primary aim is to give a verdict on the algorithms  $\mathcal{E}$  and  $\mathcal{D}$ . How should the input streams  $M$  and key streams  $K$  be chosen?

One view is to specify some specific files and demand that the algorithms performs well on these. In our view, we should choose the files randomly, subject to some probabilistic structure- some files should be pure random bit-streams, some could be of only two or three characters randomly placed, some could be sparse (very few 1's) and some could be dense (very few 0's)

In the proposed battery of tests 50 files  $M_1, M_2, \dots, M_{50}$  of specified structure are randomly generated and likewise, 50 key files  $K_1, K_2, \dots, K_{50}$  are generated. For the 2500 pairs, the Maurer's statistic is computed for each of the 10 streams given above.

$$X_{ijk} = Z(B_i(M_j, K_j)) \quad 1 \leq i \leq 10; 1 \leq j, k \leq 50$$

How does one combine these 25000 values and come out with a verdict for the cipher algorithm?

Extensive tests with various known algorithms and their variations have shown that when the algorithm is weak, it fails one type of test and so we propose that

$$U_i = \sum_{j,k} X_{ijk}^2$$

is computed, this under Null Hypothesis is approximately Chi-Square (2500) (since we really do not know if  $X_{ijk}$ 's are independent).

Let

$$V_i = \frac{1}{50\sqrt{2}}(U_i - 2500)$$

and

$$W = \sum_i V_i^2.$$

$W$  would have Chi-square(10) distribution, under Null Hypothesis and assuming independence. We have observed that when algorithm is weak,  $W$  values tend to be rather large. We can use a 0.999 percentile point, 29.59 as cut-off.

We have tested this consolidated test of randomness on several blockciphers and their variants created to be weaker. We next present the results for AES (Advanced Encryption Standard) algorithm and two of its variants (one with fewer number of iterations and one where one of the mixing operations has been deleted) For each of this we have made six runs and we will give values of  $V_1, \dots, V_{10}$  along with  $W$ . Recall, the critical region for  $W$  is

$$\{W \geq 29.59\}.$$

## Results for 6 runs of tests on AES

$V_1$	1.0	-2.3	0.0	0.7	1.0	1.3
$V_2$	-1.0	-1.4	1.3	0.3	-0.4	-1.2
$V_3$	1.3	-0.3	-0.3	-1.2	-1.7	0.6
$V_4$	-0.1	-1.1	-0.6	-0.1	0.3	-1.4
$V_5$	1.0	2.2	0.7	-0.5	-0.1	-1.1
$V_6$	1.0	-0.1	0.9	-0.2	0.2	0.2
$V_7$	1.1	0.3	1.0	-0.1	-0.6	-1.5
$V_8$	-0.9	0.2	0.3	0.7	0.0	0.6
$V_9$	0.4	0.8	0.3	0.9	-1.5	-0.5
$V_{10}$	0.3	-1.4	-1.4	1.4	-0.6	-1.4
$W$	7.9	16.1	6.5	5.7	7.3	11.6

### Results for 6 runs of tests on AES modified-1

$V_1$	1.0E-02	4.0E-02	-1.6E-01	-7.3E-01	-9.1E-01	2.8E-01
$V_2$	1.9E+00	5.8E-01	7.9E-01	1.3E+00	-1.1E+00	-9.3E-01
$V_3$	1.8E+02	1.8E+02	2.3E+02	6.3E+02	4.7E+02	3.2E+02
$V_4$	1.1E+00	-3.4E-01	-1.4E-01	-1.2E+00	2.7E-01	-3.0E-02
$V_5$	9.5E+02	9.3E+02	9.3E+02	9.3E+02	9.3E+02	9.4E+02
$V_6$	1.5E+00	-9.4E-01	1.6E+00	4.7E-01	1.1E+00	4.0E-02
$V_7$	5.3E-01	-1.8E-01	-3.3E-01	2.0E+00	6.0E-01	6.8E-01
$V_8$	1.6E+00	-7.8E-01	8.4E-01	1.7E+00	-1.2E+00	-7.2E-01
$V_9$	1.7E+02	1.2E+02	2.5E+02	1.3E+02	4.3E+01	8.6E+01
$V_{10}$	3.0E+02	2.9E+02	3.0E+02	2.9E+02	2.9E+02	2.9E+02
$W$	1.1E+06	1.0E+06	1.1E+06	1.4E+06	1.2E+06	1.1E+06

## Results for 6 runs of tests on AES modified-2

$V_1$	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05
$V_2$	2.0E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05
$V_3$	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05
$V_4$	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05
$V_5$	1.7E+08	1.7E+08	1.7E+08	1.7E+08	1.7E+08	1.7E+08
$V_6$	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05
$V_7$	1.9E+05	1.9E+05	1.9E+05	1.9E+05	2.0E+05	1.9E+05
$V_8$	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05
$V_9$	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05	1.9E+05
$V_{10}$	1.7E+08	1.7E+08	1.7E+08	1.7E+08	1.7E+08	1.7E+08
$W$	5.9E+16	5.9E+16	5.9E+16	5.9E+16	5.9E+16	5.9E+16