International Journal of
Applied Mathematics,
Electronics and Computers

IJAMEC

ISSN: 2147-8228 www.atscience.org/IJAMEC

Advanced Technology and Science

Original Research Paper

Modification Attack Effects on PRNGs: Empirical Studies and Theoretical Proofs

Santi Indarjani*[1], Kiki A. Sugeng[2], and Belawati H. Widjaja[3]

Accepted 15th August 2014

Abstract: Random sequence as a critical part in a security system should be garranted as random that should be secure from any attacks. Modification attack is one of possible attacks on random generator in order to make the generator function mislead or the output random sequences bias. From previous research, it was shown that 1-bit modification attack has effects on the randomness property of AES-based PRNG outputs under advantage $\varepsilon = 0.00001$ based on statistical distance test and entropy difference test. In this paper, we propose the extended research on some other PRNGs i.e. Rabbit, Dragon, ANSI X9.17 and ANSI X9.31 under the same scenario with intensity of modification (1-bit to 3-bits) per block. From the experiment results we found that the modification attack already has effects on the four algorithms under advantage $\varepsilon = 0.001$ with intensity 3-bits per block. Even on PRNG X9.17, the attack effect is already significant for all intensity. The effect is getting more significant for all four algorithms under advantage $\varepsilon = 0.0001$ for all intensity. It is showed that PRNG ANSI X9.17 is weaker against the modification attack than the other three algorithms. From theoretical approach based on occurrance probability of an *m*-bit pattern in the sequence after the attack, we got two results. First, the modification attack will have no effect on the probability distribution of each *m*-bit pattern as long as the modified bits are balance. So it is possible that the randomness property of the target sequence still hold after the attack. Second, if the bits modified are not balanced then it caused the unbalanced of the probability distribution of the *m*-bit patterns after attack that could make the randomness of the target sequence bias. Based on the two results, we concluded that the modification attack is potential to reduce the randomness property of the output sequences of a random or pseudorandom generator.

Keywords: *pseudorandom generator, modification attack, statistical distance, entropy difference, bit pattern, occurance probability.*

1. Introduction

As mentioned in the abstract, a random sequence is very important in the security system that based on cryptographic application. The random (pseudo random) generator is mentioned as the heartbeat of a security system [2]. However a random sequence is also important for other applications such as in packets transmission on a network, online start-up of cable tv after crash down, online access on e-ticket, or other applications that are really depend on the randomness property of a random sequence. [3].

Due to that requirement, it is very useful to consider that the random or pseudorandom generator used in those applications is secure from any attack. Some literatures ([1],[2],[4],[5]), showed that there are some attacks can be mounted on random number generator (RNG) or pseudorandom number generator (PRNG) where the mechanisms and the attack goals are vary.

One possible attack to conduct on RNG/PRNG is modification attack. This attack can be mounted through environmental attack using software approach or hardware approach. From previous research [1], it is showed that the 1-bit modification attack has effect on randomness property of AES-based PRNG with mode CFB, OFB, CTR and CBC under advantage $\varepsilon = 0.00001$. But specifically on mode CBC, it already has effect under $\varepsilon = 0.0001$. This indicated that AES-based PRNG with mode CBC is weaker than other modes against this attack.

In this paper, we extend the research on some other PRNGs i.e. Rabbit stream cipher, Dragon stream cipher, PRNG ANSI X9.17 and PRNG ANSI X9.31 to acommodate all categories of crypto systems. In this research, we also did the statistical distance test and entropy difference test as we proposed in the first research [1]. To complete the knowledge of the modification attack effects on PRNG, we did the theoretical approaches by examining the occurance probability of an *m*-bit pattern in the target sequence after the attack, under the assumption that the target sequences are random before the attack.

From experimental results, we found that the modification attack are getting significant under $\varepsilon = 0.0001$ but limited only for 3-bits modification per block, except for the ANSI X9.17 that holds for all the three intensity levels. The effects are more significant under advantage $\varepsilon = 0.0001$ for all algorithms, especially for the ANSI X9.17 that perfectly affected by indication that 100% sequences can be distinguished under that value for all intensity level. This indicates that the PRNG ANSI X9.17 is weaker against the attack than other algorithms.

From theoretical proofs, we got two conditions. First, the probabilities distribution of m-bit pattern in the sequence after the attack is still balance whenever the bits modified are also

[1] *Crypto Engineering Department, National Crypto Institute, Jl. H. Usa, Bogor, Indonesia.*
** Corresponding Author: Email: santi.indaryani@ui.ac.id,*
[2] *Mathematical Department, Faculty of mathematical and basic science, University of Indomesia, Depok, Indonesia.*
[3] *Computer Science Faculty, University of Indonesia, Depok, Indonesia*
 # This paper has been presented at the International Conference on Advanced Technology & Sciences (ICAT'14) held in Antalya (Turkey), August 12-15, 2014.

balanced. This indicates that after the attack the occurance probability of each pattern is still the same. Second, the probability distribution of each pattern in the sequence after the attack is potentially damaged when the bits modified are not balanced. When the probability of bit '0' to be modified is > ½, it caused the bit '1' will occur more frequent than bit '0' after the attack (or vice versa), that could make the probability distribution of each pattern is no longer uniform.

The two results above lead us to conclude that the modification attack is potential to reduce the randomness property of the outputs of PRNGs.

The presentation of this paper is composed in 4 chapters. Chapter I is introduction including the basic idea of the attack, our contributions and the open problem left for future research. Chapter II is presenting the preliminaries, including the background theory, methodology, and related researches. Chapter III contains detailed results from the experiments and also theoretical proofs of the modification attack effects. And the last chapter is presenting the conclusion.

2. Background Theories

Modern cryptography is considered as a construction of robust systems against any malicious attempt that aim to make the systems malfunction. [6] In principle there are two kinds of attack on a cryptographic protocol, i.e. active and passive attack. [7]. Attack on RNG/PRNG can be done actively or passively depend on the goal of the attack.

According to [4], attack on RNG/PRNG can be divided into two classes : non-invasive attacks and invasive attacks. The first attack is related with external influences where the attacker can use it to disturb the RNG/PRNG such as make the input/output bits bias improperly by introducing spike in power supply, apply the electromagnetic shocks into the chip, push temperature changes, and so on. In this attack the time for the attacker is very limited. On the other hand, invasive attack need more resources from the attacker to mount the attack succesively. This attack is more powerful and the goal is to make a permanent damage on the target RNG/PRNG.

In [5], the attacks are divided into 3 classes : direct cryptanalysis attack, input based attack, and state compromise extension attack. These attacks comes from the idea that RNG/PRNG is designed to produce random numbers such that indistinguishable from truly random numbers. Therefore the attacks tried to find the possibility of distinguishing the RNG/PRNG outputs from trully random numbers.

Young and Yung [2] proposed another scheme of attack on RNG/PRNG by implementing the Trojan to manipulate the functions in order to get advantage of it. The Trojan can be designed to reveal the critical information such as the key (seed of PRNG example) to be sent to the attacker, make the output sequence bias, or even pretends as the right generator (masquarade). The Trojan can be made as a "bug" that will be planted into the system to apply the task that already set by the attacker, or designed based on mathematical function to influence the statistic distribution of the output bits so that the generator will be very sensitive against the entropy input.

Based on the literatures above, the attack can be mounted in traditional ways based on all possible cryptanalysis methods such as brute force, functional cryptanalysis or side channel (environmental) attack. Interestingly, it also can be peformed subversively by planting the trojan or spy chip during manufacturing. In this research, modification attack is part of environmental attack that in practice can be applied under software approach or hardware approach such as a Trojan.

As mentioned above to measure the attack effects on randomness property of the target sequence, we apply the statistical distance test and entropy difference test as indistinguishability parameters between the sequence after the attack and before the attack.

The statistical distance test are measured based on the maximum statistical distance that is proposed by Wang [8] that is defined in (1).

$$\Delta(x,y) = max_{\alpha \in S} \sum_{\alpha \in S} |P(x=\alpha) - P(y=\alpha)| \quad (1)$$

The idea to use the statistical distance are inspired by some previous researches such as [9] that using the test to distinguish the modified PRNG algorithm with the original algorithm.

Let p_i is a probability of an m-bit pattern to occur in the sequence. For the probability distribution of $D = p_1 p_2 \ldots p_n$, $n = 2^m$, the entropy of *D* is defined as : [10]

$$H_b(D) = -\sum_{i=1}^{n} p_i \log_b \; p_i = \sum_{i=1}^{2^m} log_b \frac{1}{p_i} \quad (2)$$

In this research, the entropy measurements are conducted by determining the entropy difference value between probability distribution of pattern in the sequence after the attack and before the attack. Suppose $X = p_1 p_2 \ldots p_{2^m}$ is probability distribution of each patern in the sequence before attack and $Y = q_1 q_2 \ldots q_{2^m}$ is probability distribution of each pattern in the sequence after the attack. Then the entropy difference between *X* and *Y* notated as $\Delta_{entropy}(X,Y)$ is defined as :

$$\begin{aligned} \Delta_{entropy}(X,Y) &= |H(X) - H(Y)| \\ &= \left| \left(\sum_{i=1}^{2^m} p_i log \frac{1}{p_i} \right) - \left(\sum_{i=1}^{2^m} q_i log \frac{1}{q_i} \right) \right| \end{aligned} \quad (3)$$

Because $p_i > 0$ such that $p_i log \frac{1}{p_i} > 0$ (it holds for q_i also) then it can be verified that :

$$H(x) - H(y) = \sum_{i=1}^{2^m} \left[\left(p_i log \frac{1}{p_i} \right) - \left(q_i log \frac{1}{q_i} \right) \right] \quad (4)$$

From (4) it can be proved tha$\left(p_i log \frac{1}{p_i} \right) - \left(q_i log \frac{1}{q_i} \right) \leq \left| \left(p_i log \frac{1}{p_i} \right) - \left(q_i log \frac{1}{q_i} \right) \right|$t. Thus, due to this condition we proposed to use $\Delta_{entropy}(X,Y) = max \left| \left(p_i log \frac{1}{p_i} \right) - \left(q_i log \frac{1}{q_i} \right) \right|$ as the parameter to conduct the entropy difference test in measuring the modification attack effects.

Here two definitions that are related with disjoint probability.

Definition 1 [11]:

Two events *E* and *F* are disjoint if there are no outcomes common to both E and F which is notated as $E \cap F = \emptyset$.

Definition 2 [11]:

$E \cup F$ is the collections of all outcomes in either *E* or *F* so that the probability of $E \cup F$ is the sum of each probability *E* and *F* that is written as

$$P(E \cup F) = P(E) + P(F) \quad (5)$$

3. Methodology

3.1. Simulation Process of Modification Attack.

The experiment are conducted under the same scenario as previous research [1], where the modification attack are simulated in five level of block modification with intensity of 1-bit to 3–bits per block. The location of modificated bit is determined randomly based on a random sequence using formulation dec[$\log_2(b)$],

where b is the length of the block and dec is decimal value. For example for b = 32 bits, then every position of each block will be determined by every 5-bits from a certain random sequence that will be transformed into decimal value. The bit that is pointed by this value will be modified into its complement. The illustration of the attack simulation is described in Figure 1.

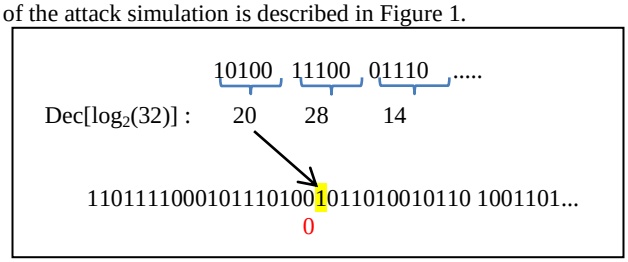

Figure 1. Example of 1-bit modification attack simulation

The attack effects measurements are performed using statistical distance test and entropy difference test under advantage value ε = 0.001. The samples used for each algorithm are 1000 random sequences with length 10^6 bits.The hypothesis for statitical distance test is defined :

H_0: $\Delta(x, y) > \varepsilon$ then x and y can be distinguihed
H_0: $\Delta(x, y) \leq \varepsilon$ then x and y can not be distinguished

and the hypothesis for entropy difference test is defined :
H_0: $\Delta_{entropy}(x, y) > \varepsilon$ then x and y can be distinguished
H_0: $\Delta_{entropy}(x, y) \geq \varepsilon$ then x and y can't be distinguished

To see more detail information, the analysis will be extended into measurement under ε = 0.0001 as a comparison.

3.2. Theoretical Proofs of Modification Attack Effects.

To get more information about the modification attack effects on the randomness property of the bit sequences produced by PRNGs, we also did some theoretical proofs using probability theory. In this case, it is assumed that the output sequence of PRNGs is random, so that each pattern occurs in the sequence has the same probability, or in other word the probability distribution of each pattern in the random sequence is uniform [12]. The measurement is done by determining the occurence probability of an m-bit pattern after the attack, under assumption that if the occurence probability of each pattern is changed, then the modification attack has effects on randomness property of the sequence. Otherwise the attack has no effects.
The m-bit pattern is all possible patterns that can be derived from a sequence with length m-bits. For example, for m =1 there are two patterns i.e. bit 0 and 1, for m = 2 we have 4 patterns i.e. 00, 01, 10, and 11, and so on. The proof is conducted in three cases: 1) the bits modified are balance that means each pattern has the same probability $P \approx \frac{1}{2^m}$ to be modified; 2) the bits modified are not balance that means the probability of one or more m-bit patterns are higher than the expected probability $\frac{1}{2^m}$ or $P > \frac{1}{2^m}$; and 3) the bits modified are not balance means the probability of one or more m-bit patterns are higher than the expected probability $\frac{1}{2^m}$ or $P < \frac{1}{2^m}$. Here some notations we used in this paper :

$U = u_1 u_2 \ldots u_n$ is the target sequence from a PRNG
$n_{i_1 i_2 \ldots i_m}$ is the m-bit pattern in a sequence before the attack
$s_{i_1 i_2 \ldots i_m}$ is the m-bit pattern that is modified
$n'_{i_1 i_2 \ldots i_m}$ is the m-bit pattern in a sequence after the attack

4. Related Works

Becker et al [13] proved that it is possible to implement hardware Trojan as a subversive attack into a crypto device and evaluate the impact on the security of the target device. They demonstrating the attack by inserting the Trojans into two designs: a digital post-processing derived from Intel's cryptographically secure RNG design used in the Ivy Bridge processors; and a side-channel resistant S-Box implementation.
The first attemp showed the Trojan can reduce the security of random key sequence produced by the RNG from 128-bit into n-bit where n is chosed by the attacker for n < 128 bits. The RNG device with Trojan inserted still passed the Built-in-Software-Test (BIST) and the output key sequence produced still passed the NIST randomness test tool, so that the user does not recognize the attack. Second attemp proved that the Trojan succed to reveal the right key with correlation goes up to 0.9971. They also proved that the resistancy of the design with Trojan and the design without Trojan are similar, so that user could not detect that the device was attacked.
Second related work is from Markettos and More [14] that implementing the injection attack on RNG in 2004 EMV Payment Card by injecting signals through prover suppy. The attack succed to make the output sequence bias that automatically reduce the security of the RNG output sequeces that used as PIN number for payment application from 2^{32} into 2^8 bits. The idea of the two attacks above is similar with the modification attack proposed by the writer.
The modification attack is also possible to be applied on RNG/PRNG as a Trojan based on software or hardware approach. This paper does not explained the modification attack on RNG/PRNG technically in practice but by simulation process. The writers show the possible impacts of the attack through empirical study by simulation process and theoretical proofs based on occurence probability of each pattern in the sequence after the attack.
From the experiment results and theoretical proofs, the modification attack could reduce the randomness property of the target sequence under certain circumstances that will be described in detail in the following chapter.

5. Results

5.1. Experimental Results

The experimental results of modification attack effect on the four algorithms Dragon, Rabbit, ANSI X9.17 and ANSI X9.31 based on statistical distance test under advantage value ε = 0.001 is presented in Table 1.

Table 1 Statistical distance test results under $\varepsilon = 0.001$

Modification intensity	Maximum statistical distance test			
	Rabbit	Dragon	ANSI X9.17	ANSI X9.31
1-bit	0(0%)	0(0%)	24(96%)	0(0%)
2-bit	0(0%)	0(0%)	24(96%)	0(0%)
3-bit	12(48%)	11(44%)	24(96%)	11(44%)

From the data in Table I, there are 2 (two) interesting results. First, the modification attack has affected the sequences only at level intensity of 3-bits per block for three algorithms Rabbit, Dragon and ANSI X9.31. This is indicated by some values that are already exceeded the advantage value ε = 0.001, which means that the sequences after the attack can be distinguished from the original sequence under this advantage value. The effects is not

significant, because not more than 12% sequences met that condition. Second, the attack is very significant for ANSI X9.17 at all level of intensity where about 96% sequences already exceeded the advantage value $\varepsilon = 0.001$.

As a comparison, the entropy difference test results on the four algorithms under advantage value $\varepsilon = 0.001$ is shown in Table 2. The test results as shown in Table II indicate that the modification attack also already affects the target sequences under $\varepsilon = 0.001$ but only at intensity level 3-bits per block for all algorithms except ANSI X9.17 that is already affected at all level of intensity. But the attack effects on ANSI X9.17 under this test is less significant compared with the statistical distance test results above, because the total sequences that exceeded the advantage value is more lower.

Table 2. Entropy difference test results under $\varepsilon = 0.001$

Modification intensity	Maximum entropy difference test			
	Rabbit	Dragon	ANSI X9.17	ANSI X9.31
1-bit	0(0%)	0(0%)	12(48%)	0(0%)
2-bit	0(0%)	0(0%)	12(48%)	0(0%)
3-bit	13(52%)	13(52%)	19(76%)	14(64%)

The two test results above indicates that the modification attack effects is less significant for three algorithm Dragon, Rabbit and ANSI X9.31 under advantage ε = 0.001, but very significant for ANSI X9.17 under the same advantage. This fact indicates that under advantage value $\varepsilon = 0.001$, ANSI X9.17 is relatively more weaker than other three algorithms against the modification attacks.

If the advantage value is reduced to a lower level $\varepsilon = 0.0001$ it is proved that the modification attack effects is more significant for all algorithms at all level of modification intensity as can be seen in Table 3 and Table 4.

Table 3. statistical distance test results under $\varepsilon = 0.0001$

Modification intensity	Maximum statistical distance test			
	Rabbit	Dragon	ANSI X9.17	ANSI X9.31
1-bit	1(4%)	0(0%)	25(100%)	0(0%)
2-bit	5(20%)	5(20%)	25(100%)	4(16%)
3-bit	22(88%)	22(88%)	25(100%)	22(88%)

Table 4. . entropy difference test results under $\varepsilon = 0.0001$

Modification intensity	Maximum entropy difference tests			
	Rabbit	Dragon	ANSI X9.17	ANSI X9.31
1-bit	0(0%)	0(0%)	25(100%)	0(0%)
2-bit	6(24%)	7(28%)	24(96%)	6(24%)
3-bit	21(84%)	22(88%)	25(100%)	23(92%)

From the two test results under advantage $\varepsilon = 0.0001$ it can be seen that the attack effects for all algorithms except ANSI X9.17 are quite similar at any level of intensity. And as many bits are modified is incresed at higher intensity, the number sequences after the attack that can be distinguished from the original sequence are also increased.

Compared with the test results on AES-based PRNG from previous research [1], the 1-bit modification attack still not affected the randomness property of AES-based PRNG under advantage ε = 0.001 even under ε = 0.0001 based on statistical distance test. This condition holds for all modes and all varians. The attack has just affected the randomness of the sequence under advantage value ε = 0.00001.

For entropy difference tests results on AES-based PRNG for all modes and all varians, the 1-bit modification attack still has no effect under advantage $\varepsilon = 0.001$. This condition also holds for the advantage value $\varepsilon = 0.0001$ except for mode CBC for all varian, where only a small number of maximum entropy difference values (less than 8%) has exceeded the advantage value $\varepsilon = 0.0001$ that can be ignored. This lead to a conclusion that the 1-bit modification attack has no effect on AES-based PRNG for all varians and all modes under ε = 0.001 even under advantage ε = 0.0001 that contradictive with the attack effects on other four algorithms under the same advantage value that presented in Table 4.

The comparison results showed that AES-based PRNG is more stronger against modification attack at level 1-bit intensity, meanwhile ANSI X9.17 is the weakest among the 5 algorithms against the attack under the same level.

Based on the overall experimental results, it could be concluded that modification attack has different effects on RNGs/PRNGs. One important point that under certain advantage value, the attack could be potentially damage the randomness property of the output of RNG/PRNG. To complete the results, the following chapter presented the theoretical proofs about the modification effect based on occurence probability of each pattern in the sequence.

5.2. Theoretical Approaches Results

Suppose there is a random sequence U_n with length n bits. Since the sequence U_n is asummed to be random then the bit 0_s and the bit 1_s will have the same probability to occur in the sequence. Let n_0 is all bit 0 in U_n and n_1 is all bit 1 in sequence U_1 such that $P(n_0) \approx P(n_1) \approx \frac{1}{2}$. Suppose s bits in sequence U_n will be modified into its complement and $s < n$. Suppose the complement bits are notated as s' such that $n - s + s' = n$. The proofs of modification attack effects are conducted in two schenarios i.e. when the bits modified are balanced and not balanced for each pattern.

First for 1-bit pattern, suppose s bits in U_n will be modified where s may contain of some bit 0 and some bit 1, all bit 0, or all bit 1. Let s_0 notated as all bit 0 in U_n that are modified into bit 1, which will be notated as s'_1 after modification. Vice versa, s_1 notated as all bit 1_s in U_n that are modified into bit 0 which will be notated as s'_0 after modification. Then we get $s'_1 = s_0$ and $s'_0 = s_1$ where $s'_0 + s'_1 = s_1 + s_0 = s$.

It can be proved that $n - s + s' = (n_0 + n_1) - (s_0 + s_1) + (s'_1 + s'_0) = (n_0 - s_0 + s'_0) + (n_1 - s_1 + s'_1) = n$. In other words, it is proved that the modification attack does not change the total number of bits in the sequence U_n.

Case 1, suppose that the probability of bit 0 will be modifed is the same as bit 1, then $s_0 = s_1$, so that the occurence probability of bit 0 in the sequence after the attack can be expressed as (Note that $P(s_0) \approx P(s_1) = \frac{1}{2}$):

$$P(n'_0) = \frac{P(n_0)n - P(s_0)s + P(s_1)s}{n} \approx \frac{\frac{1}{2}n - \frac{1}{2}s + \frac{1}{2}s}{n} = \frac{1}{2} \qquad (6)$$

The expression (6) also holds for bit 1 such that the probability of bit 1 to occur after the attack is ½.

Case 2, If probability of bit 0 which will be modified is not the same as bit 1. Suppose probability of bit 0 to be modified is bigger than bit 1, notated $P(s_0) \approx \frac{1}{2} + \delta$, so that $s_0 > s_1$, then the occurence probability of bit 0 after the modification attack can be expressed as (note that $P(s_1) = P(s'_0) = 1 - P(s_0)$) :

$$P(n'_0) = \frac{P(n_0)n - P(s_0)s + P(s'_0)s}{n} \approx \frac{\frac{1}{2}n - \left(\frac{1}{2}+\delta\right)s + \left(1-\left(\frac{1}{2}+\delta\right)\right)s}{n}$$
$$= \frac{\frac{n}{2} - \left(\frac{s}{2}+\delta s\right) + \left(\frac{s}{2}-\delta s\right)}{n} = \frac{n-s-2\delta s+s-2\delta s}{2n}$$
$$= \frac{n-4\delta s}{2n} = \frac{1}{2} - \frac{2\delta s}{n} < \frac{1}{2} \quad (7)$$

Thus from (7) we conclude that the probability of bit 0 to ocur after the attack is less than ½ that imply the probability of bit 1 to ocur is higher than ½. With the same way it can be prove that if the probability of bit 0 to be modified is less than bit 1 notated $P(s_0) \approx \frac{1}{2} - \delta$ such that $s_0 < s_1$, then the probability of bit 0 to ocur after the attack is :

$$P(n'_0) \approx \frac{n+4\delta s}{2n} = \frac{1}{2} + \frac{2\delta s}{n} > \frac{1}{2} \quad (8)$$

From (7) and (8) we have that the probability of each bit 0 and 1 to ocur after the attack will not balance if the probability of being modified for each bit is not the same. On the other hand it will still balance if the probability of each bit to be modified is the same.

If we extend the pattern bit into *m*-bit pattern with the same way, we could generalized the formulation of occurence probability of an *m*-bit pattern after the modification attack. Let $P(s_{i_1 i_2 \dots i_m})$ is the probability of an *m*-bit pattern $s_{i_1 i_2 \dots i_m}$ in the target sequence that will be modified into its complement $s^c{}_{i_1 i_2 \dots i_m}$, and let $P(s^c{}_{i_1 i_2 \dots i_m})$ is the probability of $s^c{}_{i_1 i_2 \dots i_m}$ to be modified. Then the generalized formulation of the occurence probability of an *m*-bit pattern $n'_{i_1 i_2 \dots i_m}$ after the attack are as follow :

Case 1:

If the probability of being modified of each *m*-bit pattern in sequence U_n is uniform such that $P(s_{i_1 i_2 \dots i_m}) \approx \frac{1}{4}$ than probability of each bit pattern to be ocurred after the attack is :

$$P(n'_{i_1 i_2 \dots i_m}) = \frac{P(n_{i_1 i_2 \dots i_m})n - P(s_{i_1 i_2 \dots i_m})s + P(s^c{}_{i_1 i_2 \dots i_m})s}{n}$$
$$\approx \frac{\frac{1}{2^m}n - \frac{1}{2^m}s + \frac{1}{2^m}s}{n} = \frac{1}{2^m} \quad (9)$$

Case 2 :

Suppose the probability of each *m*-bit pattern in sequence U_n of being modified is not uniform. Let the probaility of an m-bit pattern of being modified is bigger than $\frac{1}{2^m}$ such that $P(s_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} + \delta$. Then the probability of an *m*-bit pattern to be occurred after the attack can be defined in three cases depends on the probability of $s^c{}_{i_1 i_2 \dots i_m}$:

- $P(n'_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} - \frac{\delta s}{n}$, (10)

  where $P(s^c{}_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m}$

- $P(n'_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} - \frac{(\delta-\beta)s}{n}$, (11)

  where $P(s^c{}_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} - \beta$

- $P(n'_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} - \frac{(\delta+\beta)s}{n}$, (12)

  where $P(s^c{}_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} + \beta$

Case 3 :

In contrary of case 2, when the probability of an m-bit pattern of being modified is less than $\frac{1}{2^m}$ such that $P(s_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} - \delta$. Then the probability of an *m*-bit pattern to be occurred after the attack can be defined as folow:

- $P(n'_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} - \frac{\delta s}{n}$, (13)

  where $P(s^c{}_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m}$

- $P(n'_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} + \frac{(\delta+\beta)s}{n}$, (14)

  where $P(s^c{}_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} + \beta$

- $P(n'_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} - \frac{(\delta+\beta)s}{n}$, (15)

  where $P(s^c{}_{i_1 i_2 \dots i_m}) \approx \frac{1}{2^m} - \beta$

From the theoretical proofs above then we come to conclusion that the modification attack is potential to destroy the randomness property of a random sequence if the probability of each *m*-bit pattern that modified is not uniform. In contrary, the probability of each m-bit pattern to ocur after the attack are uniform if the probability of each *m*-bit pattern that are modified are also the same or uniform.

6. Conclusion

From empirical study based on statistical distance test and entropy difference test on some PRNGs i.e. AES-based PRNGs, Dragon, Rabbit, ANSI X9.17 and ANSI X9.3, we found that the modification attack is potential to affect the randomness property of the output sequences of PRNGs but the significancy of the effects are different for each algorithm.

From theoretical proofs based on occurance probability of each m-bit pattern after the attack, the modification attack may destroy the randomness property of a random sequence as long as the probability of each m-bit pattern modified bits is not uniform. In contrary, the probability of each *m*-bit pattern will still have the same probability to occur after the attack, if the probability of each pattern that is modified is the same.

Based on the two results above, the modification attack may have bad impacts on randomness property of the outputs from RNG or PRNG. And from related researches in [13] and [14], it showed that this kind of attack is possible to be implemented in practice, where as an adversary can conduct the modification attack as a Trojan in order to reduce the randomness property of the RNG/PRNG's outputs. Therefore, this modification attack cannot be disobeyed and should be anticipated.

Acknowledgements

Thanks to Laboratory of National Crypto Institute for facility supports in conducting the attack experiments. Thank you to my colleague Sari Agustin from National Crypto Agency for helps in understanding the basic theory of probability related to the case, and all friends that are involved in performing the experiments..

References

[1] Indarjani, S. and Widjaja, B., "Indistinguishable of AES-based PRNG against Modification Attack Based on Statistical Distance Tests and Entropy Measures," ser. Lecture Notes on Software Engineering vol. 1, no. 3, 2013, pp. 314-318.

[2] Young, A. and Yung, M., Malicious Cryptography : Exposing Crypto virology, John Willey & Sons, USA, 2004

[3] Uner, E., "Generating Random Numbers," Embedded: Cracking the code to system development, 2004, available: http://www.embedded.com/ design/configurable-systems/4024972/Generating-random-numbers. .
[4] Sunar, B., Martin,W.J., and Stinson, D.R., "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks," IEEE Transactions on Computers, vol. 56, no. 1, January, 2007, pp. 109-119.
[5] Schneier, B., Kelsey, J., Wagner,D., and Hall, C., "Cryptanalytic Attacks on Pseudorandom Number Generators", in Fast Software Encryption, Fifth International Workshop Proceedings, published by Springer-Verlag, March 1998, pp. 168–188.
[6] Goldreich, O., Foundation of Cryptography : Volume I Basic Tools, Cambridge University Press., England, 2001
[7] Schneier, B., Applied Cryptography, 2nd ed., John Wiley & Son, Inc., USA, 1996.
[8] Wang, Y., A comparison of two approaches to the randomness, Theor. Comput. Sci., Vol. 276, No. 1-2. ,2002, pp. 449-459
[9] Farashahi, R.R., Schoemaker, B., and Sidorenko, A., "Efficient of Pseudorandom Generators based on DDH Assumption", in Lecture Notes in Computer Science Volume 4450, published by Springer-Verlag, Berlin, 2007, pp 426-441.
[10] Bose,R., Information Theory, Coding and Cryptography, Tata McGraw Hill, New Delhi, 2002
[11] Hoffstein, J., Pipher, J, and Silverman, J.H., An Introduction to Mathematical Cryptography, Springer Science+Business Media, LLC, New York, 2008
[12] NIST, NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, 2004, available : http://csrc.nist.gov/groups/STM/cavp/documents/rng/931rngext.pdf
[13] G.T. Becker, F. Regazzoni, C. Paar, and W.P. Burlesson, "Stealthy Dupont-level Hardware Trojan", in CHES'13 Proceedings of the 15th international conference on Cryptographic Hardware and Embedded Systems, published by *Springer-Verlag, Berlin, 2013*, pp. 197-214.
[14] A.T. Markettos and S.W.. Moore, "The Frequency Injection Attacks on Ring-Oscillator-Based True Random Number Generator", in Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems, pp. 317 – 331, Springer-Verlag, 2009.