2012 4th International Conference on Computer Engineering and Technology (ICCET 2012)
IPCSIT vol.40 (2012) © (2012) IACSIT Press, Singapore

Measuring the Insertion Attack Effect on Randomness Property of AES-based Pseudorandom Generator

Santi Indarjani [1] + and Belawati Widjaja [2]

[1] National Crypto Institute

[2] Faculty of Computer Science, University of Indonesia

Abstract. Random (pseudorandom) number generator (RNG/PRNG) as the heart of a cryptographic system could be a potential target for adversary to defect the security. The attack can be performed actively through insertion attack on the random outputs to reduce or even omit the randomness property. In this paper, we try to figure out the behavior of AES-based pseudorandom generator for all variants with OFB, CFB and CTR mode against the insertion attack. The insertion attack performed in five levels of insertion block (32-bit, 64-bit, 128-bit, 256-bit and 512-bit), by inserting 1bit-through 3-bits in a random manner. The bits inserted and the location of insertion are taken from a (difference) random sequence, where the location is determined by formulation 2Logn (n is the block of insertion). The tests are done by conducting the randomness test and the statistical distances test on the random sequences before and after the insertion attack. The randomness test used is NIST randomness tool with level significance of $\alpha = 0.01$. We use 1000 samples with length 10^6 bits for each variant. The results from the randomness tests showed that the insertion attack doesn't give the significant effects on the randomness property of AES-based PRNG which is shown that only about 21.48% of all samples have failed test at most 3 tests on a single experiment. The second tests are still in progress, but temporary results showed that the sequences before and after insertion attack are indistinguishable under ε=0.01, which come to conclusion that the AES-based PRNG is still random after the insertion attack.

Keywords: pseudorandom generator, randomness, insertion attack, encryption mode, statistical distance.

1. Introduction

Information as a critical asset need to be secured comprehensively and properly that would not be apart from cryptographic applications for assuring the confidentiality, integrity, authentication and non repudiation services [1]. Random number generator (RNG) or pseudorandom generator (PRNG) is the heart of a cryptographic system because it provides a secret key, nonce, IV, or other input parameter needed in cryptographic applications [2]. Any weakness on RNG/PRNG will automatically weaken the system. Therefore this critical component can be a potential target for adversary to reduce the security.

On the other hand, randomness is also needed in many other applications outside cryptography. For example, in a network system, every data packet will be transmitted almost at the same time that differs only a little bit of time, which is determined based on a random sequence. With no collision in transmitting time, all the packets will be delivered properly without crashed that will avoid the data corruption [3].

In some cases generate a truly random sequence using a (true) RNG is not practical, so that a PRNG is more preferred. Another problem is that there is possibility that the RNG/PRNG we use is being attacked. An adversary could find a way to compromise the input key (seed) or to manipulate the output in order to reduce the randomness property such as through insertion attack. The problem is, how does the effect of insertion attack on the randomness property of an RNG or PRNG? This paper proposes the effect of insertion attack on randomness property of AES-based PRNG of all variants with mode OFB, CFB or CTR by comparing the statistical property before and after the insertion attack.

+ Santi Indarjani. Tel.: +622186600645; fax: +62251-8541720.
E-mail address: santi_indarjani@yahoo.com

2. Definitions and preliminaries

2.1. Introduction of AES

AES is adopted as a substitution of Data Encryption Standard (DES). AES has 128-bit length of block with three variety of keys are 128-bits, 192-bits and 256-bits. AES parameter can be seen on Table 1[4].

Table 1: AES Parameters [4]

Parameters	AES-128	AES-192	AES-256
Key size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext block size (word/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of rounds	10	12	14
Round key size (words/bytes)	4/16/128	4/16/128	4/16/128
Expanded key size (words/bytes)	44/176	52/208	60/240

There are 4 different stages used in a single round of AES are:

- Substitute byte: Use an S-box to perform a byte-by-byte substitution of the block.
- Shift Row : a simple permutation
- Mix columns: a substitution that makes use of arithmetic over GF(2^8).
- Add round key: a simple bitwise XOR of the current block with a portion of the expanded key.

In a block cipher implementation there are some modes of encryption can be used to provide the suitable structure in certain application, i.e. Electronic Code Book (EBC), Cipher Block Chaining (CBC), Output Feedback (OFB) and Cipher Feedback (CFB) and Counter Mode (CTR) [4]. In our experiments we use mode OFB, CFB and CTR on AES algorithm to produce the pseudorandom sequences.

2.2. Concept of Randomness

All Random sequences generally divided into two classes i.e. truly random sequences and pseudorandom sequences, that produced by RNG and PRNG respectively [1]. RNG is defined as a system whose outputs consists of fully unpredictable (i.e., statistically independent and unbiased) bits. In security applications, the unpredictability of the output implies that the generator must be also not observable or even manipulated by any attacker. A true random bit generator usually based on some kind of non-deterministic phenomena [5]. PRNG is defined as a function that, once initialized with some random value (called the seed), outputs a sequences that appears random, in the sense that an observer who does not know the value of the seed cannot distinguish the output from that of a (true) random generator. PRNG is a deterministic process where put back in the same state it will reproduce the same sequence. [5]

Two general requirements that should be met by a pseudorandom bit sequence are: 1) the output sequence of pseudorandom generator is statistically difficult to distinguish from a truly random sequence, and 2) the output sequence of pseudorandom generator could not be predicted computationally by the enemy with limited resources. [6]. So that randomness property of a PRNG is important.

Maurer stated that randomness test is done to detect any statistical defect of a random generator (or pseudorandom generator). In test practice, the level of significant α should be relatively small in the range of 0.001 - 0.1 [7]. There are some randomness statistical tools that can be used to test the pseudorandom generator. The simple tool is five basic tests that contained with frequency test, serial test, poker test, run test and autocorrelation test [8]. Based on NIST SP 800-22 there is more complicated test tool that contains with 15 tests [9]. Another test tool is Diehard battery developed by Peter Marsaglia in 1996.

2.3. Statistical Distance

Let x and y be random variables taking on values in a finite set S. The statistical distance between x and y is defined as [10].

$$\Delta(x,y) = 1/2\sum_{\alpha\in S}\left|\Pr(x=\alpha) - \Pr(y=\alpha)\right| \tag{1}$$

An algorithm D distinguishes x and y with the advantage ε if and only if

$$\left|\Pr(D(x)=1) - \Pr(D(y)=1)\right| \geq \varepsilon \tag{2}$$

If the statistical distance between *x* and *y* is less than ε then no algorithm distinguishes *x* and *y* with advantage ε.

Goldreich also said that two ensembles *X* and *Y* are statistically close if their statistical difference is negligible, where the statistical difference (also known as variation distance) is defined as the function in (1). He said that if the ensembles *X* and *Y* are statistically close, then they are also polynomial-time-indistinguishable [11].

2.4. Attack on PRNG

Goldreich [11] said that modern cryptography is concerned with the construction of schemes that should be able to withstand any abuse, and the schemes are designed so as to maintain a desired functionality, even under malicious attempts aimed at making them deviate from their functionality. An adversary attacking a system will try to manipulate the environment into untypical states. Meanwhile, Sunar and Stinson divide active attack on an RNG in two categories are [12]:

- **Non-invasive attacks**. This attack is related with the external effect when attacker can bias (not accurately) the input/output bit (such as induces spike in power supply), apply electromagnetic shocks on the chip or push the temperature changes spontaneously.
- **Invasive attacks.** This type need more resources but more dangerous because can cause disturbance or permanent defects. The target is to destruct the random generator and make the output is bias.

Tilborg said that the defects on random generator can be done applied as electrical or timing offsets, intrinsic design behaviours, interferences with internal or external signals. But the most important problems can be caused by manipulation attempts such as the attacker trying to inject a signal to force the output stream bits which is undetected in statistical test. In other words, the random source still passes the randomness statistical tests even it was injected by certain signal. [5].

Other possible attack on RNG (PRNG) is manipulate a Trojan horse that living in the system to provide the attacker the access to get a critical entity produced such as the seed of the RNG or the output stream bits. A Trojan horse also could be manipulated to defect the statistic distribution of output of the RNG (PRNG) such that the output is very sensitive to the input entropy. An attacker also can put a Trojan horse in the RNG/PRNG (hidden) without detectable to perform some attacks such as insertion, deletion or even repetition of the output produced so that the randomness is not a guaranty [2].

Therefore, the insertion attack is possible to conduct by an adversary through software or hardware approaches, in order to force the PRNG dysfunctional or bias the output of a PRNG so that will reduce or even omit the randomness property.

2.5. Methodology

In this research, we perform simulation by implementing the insertion attack by injecting 1-bit through 3-bits into random output sequences produced by AES-based PRNG. The insertion attack is performed in five levels of insertion block (32-bit, 64-bit, 128-bit, 256-bit and 512-bit) in a random manner. The location of insertion in each block is determined based on the decimal value of ^{2}logb bits taken from a random sequence. The bit inserted is also taken from a random sequence. The two random sequences and the seed are generated by Random C. To see the effect of the insertion attack on the randomness property of AES-based PRNG, the sequence before and after insertion attack will be evaluated by performing randomness test and statistical distance test. The experiment model is described in Fig 1.

The statistical distance is performed using the surrounding function of the PRNG that evaluated under ε = 0.01 on the data sample to see whether the PRNG after insertion can be distinguished with advantage ε = 0.01 from the PRNG before attack. In this experiment, the statistical distance test is evaluated not only on the proportion of number bit 0 and 1 but are extended into some patterns limited to 1, 2, 3, 4 and 8 bit. The extension aims to see the effect of the insertion attack on the random pattern in the PRNG. The pattern is counted without overlapping. The formulation then can be defined as follow

$$\Delta(x, y) = 1/2^n \sum_{\alpha \in S} \left|\Pr(x = \alpha) - \Pr(y = \alpha)\right| \qquad (3)$$

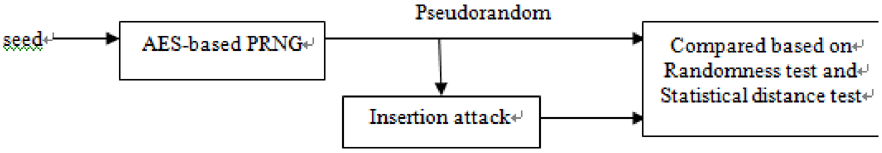

Fig.1: The Experiment Model

3. Experiment Results

3.1. Randomness test results before insertion

The randomness test results on 9 sequences produced by each AES variant with mode OFB, CFB and CTR only failed on two sequences as seen on Table 2, otherwise pass the test.

Table 2: Randomness Test results of AES-based PRNG before Insertion

AES variant	Mode	Test result	Summary
128	CFB	No	Fail on one FFT test because the Proportion 977/1000 < 982/1000
256	CTR	Yes	Failed on one test of Non Overlapping template because the p-value is less than 0.01.

3.2. Results after the insertion

After insertion attack in five levels of block insertion on the nine samples of random sequence using random bit inserted that gives total 135 experiments, we obtain the results as shown in Table 3.

Table 3: Summary of Test Results after Insertion Attacks

AES Variant	Number failed	Location failed	The tests failed	Total test failed	
128 OFB	3	128-1, 256-3, 512-1	4 NOT and 2 FFT	6	12
128 CFB	2	128-1, 256-3	1 serial and 2 NOT	3	
128 CTR	3	32-1, 64-2, 512-3	1 RX and 2 NOT	3	
192 OFB	3	128-1, 256-3, 512-3	3 NOT	3	12
192 CFB	1	32-2	1 NOT	1	
192 CTR	7	32-1, 32-2, 64-2, 64-3, 128-2, 256-1, 512-1	1 FFT, 1 universal, 1 serial, 5 NOT	8	
256 OFB	5	32-1, 128-1, 128-2, 256-2, 512-3	6 NOT, 1 FFT, and 1 RX	8	13
256 CFB	2	256-2, 256-3	1 FFT and 1 NOT	2	
256 CTR	3	128-1, 256-2, 256-3	1 RX, 1 RXV, 1 NOT	3	

Note: NOT = Non Overlapping Template, RX = Random Excursion, RXV = RX variant.

From data, there are 8 from 45 experiments (17.17%) are having failed test at most 3 tests on AES-128, 11 experiments from 45 (24.44%) on AES-192 are having failed one test where only one experiment has two failed tests, and 10 experiments from 45 (22.22%) on AES-256 are having failed test at most 3 tests, where 8 of them just have one test. So over all from 135 experiments, there are about 21,48 % experiments that failed the test for at most 3 tests. The most attack failed in the experiments is Non-Overlapping Template test for 25 times, followed by FFT test (5 times), random excursion (variance), serial and universal test that failed at most only two times. On average each experiments has failed at most 2 tests. Compare with the condition before insertion attack, which the test failed is at most one test (FFT test), then the effect of the insertion attack on AES-based PRNG does not significant.

The temporary results of statistical distance test on AES-128 CFB show that the value Δ(x,y), where x is a sequence produced by AES-based PRNG CFB and y notated the sequences after insertion 1-bit through 3-bits in level block of 32-bit, 64-bit and 128 bit, all are far away less than ε= 0.01. Table 4 shows the example of value Δ(x,y) for 1-bit through 3-bit insertion attacks on AES-128 bit CFB within block 32-bit, 64-bit and 128-bit. Due to the data, the insertion attack does not give the significant effect on the randomness property of the sequence x so that x and y still indistinguishable under ε= 0.01.

Table 4: Value of Δ(x,y) on AES-128 bit CFB against 1-bit through 3-bist insertion attack within block 32-bit, 64-bit and 128-bit.

	1-32-c128	2-32-c128	3-32-c128	1-64-c128	2-64-c128	3-64-c128	1-128-c128	2-128c128	3-128c128
1-bit	0.0000003120	1.147E-05	3.12E-07	6.91E-07	2.091E-06	4.354E-06	1.808E-06	6.92E-07	1.285E-06
-bit	0.0000033820	9.328E-06	2.2844E-05	2.61505E-05	1.462E-06	1.0798E-05	1.0618E-05	3.29E-06	1.1985E-05
3-bit	1.61025E-05	1.62188E-05	5.86425E-06	2.09138E-05	1.9605E-05	2.56178E-05	2.35688E-05	2.56178E-05	2.56178E-05
4-bit	1.2682E-05	1.46635E-05	1.6198E-05	1.2833E-05	1.2272E-05	1.0089E-05	1.17305E-05	1.0089E-05	1.28089E-05
8-bit	5.61587E-06	6.09544E-06	6.08344E-06	8.02063E-07	5.37362E-06	6.25869E-06	5.76094E-06	5.98469E-06	5.98469E-06

4. Conclusion

From the randomness test using NIST tool, the number of failed test are close for all variants. The total experiment with highest failed test (3 tests) is less than 5, where most of other experiments only failed in one test. Therefore, we could conclude that the effect of insertion attack on the randomness property of AES-based PRNG is not significant under level of confidence of 99.99%. The other hand, from the temporary results on statistical distance test, it showed that the sequence before and after the insertion attack still could not be distinguished under ε=0.01, which imply that the insertion attack does not give the effect significantly on AES-based PRNG.

For advanced works, the randomness behaviour of AES-based PRNG with mode OFB, CFB and CTR still need to be evaluated with the second scenario using extreme bits inserted and also overlapping patterns. And this experiment can be extended to other class of PRNG such as Stream Cipher Algorithm and Hash-based PRNG as a comparison.

5. Reference

[1] Schneier, Bruce, 1996, Applied Cryptography, 2nd ed., Wiley & Son, Inc., USA.

[2] Young, Adam and Yung, Moti, 2004, Malicious Cryptography, John Willey & Sons, USA.

[3] Uner, Eric, 2004, Embedded System Design : Generating Random Number, Embedded.Com: CMP United Bussiness Media, http://www.embedded.com/showArticle.jhtml.

[4] Stallings, William, 2003, Cryptography and Network Security: Principles and Practices, Pearson Education, Inc.

[5] van Tilborg, H.C.A., (2005), Encyclopedia of Cryptography and Security, Springer, USA.

[6] Stinson, D.R., (1995), Cryptography : Theory and Practice, CRC Press, Inc.

[7] Maurer,U. M., (1992), A Universal Statistical Test for Random Bit Generators, Journal of Cryptology, Volume 5 Nr. 2, Springer-Verlag.

[8] Menezes, A.J., van Oorschotet, P.C. and Vanstone, S.A., 1997, Handbook of Applied Cryptography, CRC Press LLC, USA.

[9] FIPS, 2010, SP800-22-1a : A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applicatons, NIST, AS.

[10] Farashahi, Schoemaker and Sidorenko,2007, Efficient of Pseudorandom Generators based on DDH Assumption , PKC'07 Proceedings of the 10th international conference on Practice and theory in public-key cryptography, Spriinger-verlag, Berlin.

[11] Goldreich, Oded, (2001), Foundation of Cryptography : Volume I Basic Tools, Cambridge University Press., England.

[12] Sunar, B., Martin, W.J., and Stinson, D.R., (2005), A provably True Random Generator with Built In Tolerance to Active Attacks, http://www.cacr.math.uwaterloo.ca/techreports/2005/cacr2025-20.pdf