Twining Technique to Minimize Collinearity in Cryptographically Secure Pseudorandom Number Generator

K.Sathya, J.Premalatha, Vani Rajasekar

Abstract: - Random numbers are the strength of many cryptographic algorithms. They can be generated by either True Random Number Generator (TRNG) or Pseudo Random Number Generator (PRNG). TRNG generates true random numbers that is completely unpredictable by nature. PRNG generates sequence of random numbers by mathematical function defined over an initial value called seed. Recently PRNG is designed to use the sensor data as seed. The problem faced is that sensor data are likely to be exposed to an attacker, thus exploiting the randomness generated by PRNG. Wash-Rinse-Spin approach is implemented to process the sensor data before being fed into a PRNG. This approach prevents attacker from taking control of sensor data. The random numbers generated through wash-rinse-spin approach are not suitable for more secure algorithms that require uniqueness like nonce, one time pads. The time consumption is also more which is not suitable in mobile devices with limited resources. Installing multiple sensors in same environment result in collinearity among their data. Feeding these collinear data to PRNG generates collinear random numbers. To solve the problem of collinearity, twining technique is used to combine the collinear sequences. AES is used in counter mode to generate cryptographically secure random numbers in less time

Index Terms-: Twining technique, cryptographically secure random number generator, AES with counter mode, Testing randomness

1 INTRODUCTION

Random Numbers are numbers are sequence of numbers with two important properties of uniform distribution over a defined interval and unpredictability with no relation to the previous number. Random numbers are widely used in many applications like encryption, hashing, and digital signature, connection establishment to create secret key, initialization vector, one time pads, and nonce. Thus security of these applications rely on randomness of the random number used which when compromised lead to security breach. The random numbers satisfying above two properties are hard to be predicted by an attacker. There are two basic types of generators used to produce random sequences: True Random Number Generators (TRNG) and Pseudo-Random Number Generators (PRNG). Cryptographic applications use both TRNG and PRNG to generate stream of bits with uniform distribution. The stream can further be divided into blocks and used as keys in stream ciphers.

2 PSEUDORANDOM NUMBER GENERATOR (PRNG)

A Pseudo Random Number Generator generates random numbers using a deterministic mathematical formula [1]. PRNG needs an initial value called seed upon which the mathematical processing is done.

- *K.Sathya*,Dept of CT/UG, Kongu engineering college, Perundurai, Erode, India. Email: pearlhoods@gmail.com.*
- *Dr.J.Premalatha, Dept of IT, Kongu engineering college, Perundurai, Erode, India. Email: jprem@kongu.ac.in.*
- *Vani Rajasekar*, Dept of CSE, Kongu engineering college, Perundurai, Erode, India. Email: vanikecit@gmail.com.*

The generated number is again fed back to the generator as seed [2] to produce future numbers. Thus the seed determines the output of the generator. To make the generated sequence unpredictable the seed must be random and unpredictable [3]. It makes sufficient to store the seed value only to regenerate the entire random number sequence. The pseudorandom arise due to the fact that randomness produced is deterministic by an algorithm and not truly random by nature.

3 LITERATURE REVIEW

Morris Dworkin (2001) created a Recommendation for Block Cipher Modes of Operation Methods and Techniques like AES, DES. Chung-Chi et al (2005) dealt with Linear Congruential Generators (LCG) for Cryptographic Purposes. The only way to circumvent the weakness of the LCG is to hide the generated numbers from the attacker. Rukhin et al (2010) developed a Statistical Test Suite to test the statistical properties of RNG namely National Institute of Standards and Technology (NIST) Statistical Test Suite. Majid Babaei et al (2011) specified the properties of good quality random numbers. Requirements to generate random numbers are specified. Voris et al (2011) discussed how accelerometers and randomness are perfect together. Random numbers were generated from accelerometer data. Von Herrewege et al (2013) proposed Secure PRNG seeding on commercial off-the-shelf microcontrollers. SRAM was identified as entropy source for PRNG seeding. Renault et al (2014) proposed Mutual Authentication Method for WSNs Based on the Three-Card Trick Ancient Card Game. It is a novel mechanism to perform a lightweight mutual authentication between a node and sink based on simple processor operations and two messages only. Hong et al (2015) proposed Sensor-Based Random Number Generator Seeding that processed sensor data and used them as seeds for generating pseudorandom numbers.

4 SELECTION OF SEED FOR PRNG

An important outgrowth of Internet-connected devices is the embedding of sensors in mobile devices. Tremendous growth of Internet opens possibilities for insecurities that need to be

protected. In the existing system, PRNGs are implemented with entropy source from sensors. Sensors measure the physical phenomenon like temperature, pressure, movement, noise, etc. These sensors provide source of true randomness that are used as seeds in PRNGs to generate long sequence of random numbers. The sensor data are processed through wash-rinse-spin approach to prevent attacker from gaining control over the random number generation process. The wash-rinse-spin approach further increases the randomness of the seeds.

4 SEED PRE-PROCESSING

Using wash-rinse-spin approach to process sensor data imposes three problems 1.Collinearity among multiple sensor data, 2. Sequence is not cryptographically secured that are required for highly secure applications and 3. Wash-rinse-spin approach consumes more time in rinsing the seeds. When multiple sensors are housed in same environment, they measure same physical phenomena. Their data are said to be collinear since they are highly correlated. Collinear data are not suitable seeds in generating secure random numbers. Some of the applications require that random numbers be unique like nonce, some require high entropy like one-time pads [4]. The random numbers generated by wash-rinse-spin approach are not suitable for such applications as they lack to provide both features at a time. Using mobile devices or sensor devices poses time constraints in generating random numbers from their data. Wash-rinse-spin approach requires more time to rinse the seeds and spin the seeds into random sequence that may drain the power of mobile and sensor devices.

5 MULTI-SENSOR SEEDS

The PRNGs require a seed value to generate the sequence of random numbers. The seed need to have high entropy so as to be random and unpredictable. Since the seed determines the sequence to be generated, it is sufficient to store only the seed value. It has been proposed [5] to use sensor data as seed for PRNGs. The sensor seeds can't be used directly as sensors are prone to be controlled by an attacker. Hong and Liu [6] proposed to process the sensor data to prevent adversarial control of an attacker. The sensors data are washed rinsed and then spin into random sequence. The wash process eliminates the predictable patterns, the rinse process further increases the randomness of seed, and the spin process generates the random numbers. The use of multiple sensor data leads to collinearity that affects the randomness. The rinse process takes more time to compute Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT). The spin process takes much more time to produce random numbers that are not suitable for secure applications. The proposed scheme is illustrated in Fig. 1. The collinear data from multiple sensors are washed and using Linear Congruential Generator (LCG) the random sequences are generated. The generated sequences are combined using twining technique to minimize collinearity. The combined sequence is taken as 128-bit random key to encrypt the counter value using AES. The resulting 128-bit cipher text is the cryptographically secure random number.

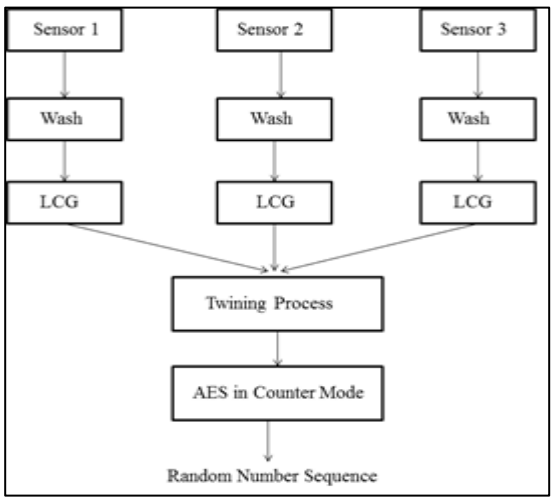

Fig. 1.*Systematic Diagram to Minimize Collinearity in CSPRNG*

6 MINIMIZING COLLINEARITY

Generating cryptographically secure random numbers have two phases.

1. Twining Technique
2. AES with Counter Mode

To test this approach accelerometer data are collected that is embedded in smart phone. The accelerometer sensor on a mobile phone records the coordination of mobile phone based on orientation of its screen. The acceleration magnitude in X-axis, Y-axis and Z-axis are considered as collinear data.

7.1 Twining Technique

The twining technique involve threes steps of operation.

1.Wash the sensor seeds
2.Generate pseudo random sequences
3.Combine the sequences to minimize collinearity

Wash

The sensor data collected represent each tap on screen, movement of phone, usage of screen, rotation. These data represent the user behavior and are predictable. These predictable patterns so called 'true data' that are still vulnerable. These predictable patterns are weak source of seeds for PRNG. This step aims to remove the slow drifts as they are predictable since they are the obvious data the sensor tracks. Removing the slow drifts prevents the adversarial control of attacker to affect the random sequences. Fig. 2 shows the collected accelerometer data with slow drifts.

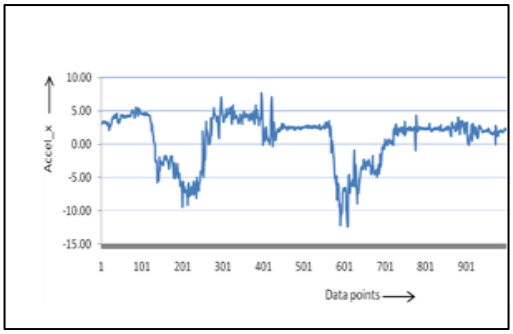

Fig. 2.*Raw Sensor Data*

Sensors produce data that are mostly predictable with little randomness, our first step is to remove these major predictable to completely eliminate the user behavior from sensor data. Since the data keep on drifting and moving within range the mean and variance sequence keep on changing over time. This is called nonstationarity. Wash step involves removing those predictable patterns by contaminating them. True data are removed by differentiating the raw data until the nonstationarity is removed. First order derivative of sensor data will be sufficient to remove nonstationarity. If still nonstationarity remains differentiation can be repeated until threshold stationarity is obtained. Stationarity represents the data are not drifting and moving anymore. Fig. 3 represents the washed data after removing the slow drifts.

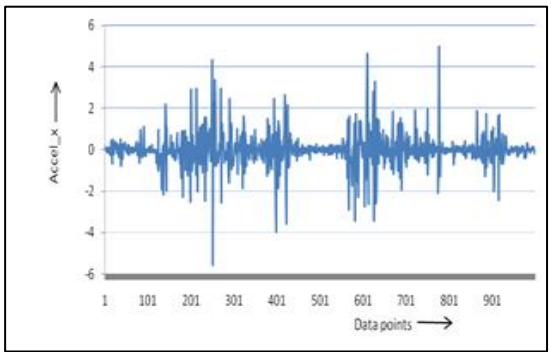

Fig. 3.*Washed Sensor Data*

PRNG Implementation

The washed data are used as seed for a PRNG. In the proposed scheme LCG was used. A LCG is an algorithm that yields a sequence of pseudo-randomized numbers calculated with a discontinuous piecewise linear equation

The generator is defined by the recurrence relation as in (1)

$$X_{n+1} = (aX_n + c) \bmod m \quad (1)$$

In the (1), 'm' is the maximum range and the generator outputs the sequence in the range of 0 to m-1. The LCG can produce uniformly distributed random numbers only when it satisfies the Hull-Dobell Theorem. Hull-Dobell Theorem states that:

m and the offset c are relatively prime i.e GCD(m, c)=1

a-1 should be divisible by all prime factors of m,

a-1 should be divisible by 4 if m is divisible by 4.

While LCGs are capable of producing pseudorandom numbers which can pass formal tests for randomness, this is extremely sensitive to the choice of the parameters c, m, and a. Hence the values of c, m, and a are chosen carefully

a = 1103515245
c = 12345
m = 2147483648

The output is the sequence of random numbers in the range 0 to 2147483647.

Combining the sequences

Collinear sensor data generates collinear random data sequences which are highly correlated. To minimize the collinearity, a novel technique called Twining technique is used. The twining process swaps and combines the random sequences into a single random data sequence through a set of swap operations [7]. The two basic left swap and right swap are carried out. The remaining basic operations like left shift, right shift and inverse are not carried out as they can be deduced from swap operations.

Left Shift, L`(C) = R(L(C))
Right Shift, R`(C) = L(R(C))
Inverse, Inv(C) = R(R`(C)) or L(L`(C))

The swap operations are performed five times over random length of three sequences. The three sequences are considered X, Y and Z. Either left swap or right swap is chosen randomly each time and segment length are chosen random. Fig. 4 pictorially shows the left swap operation.

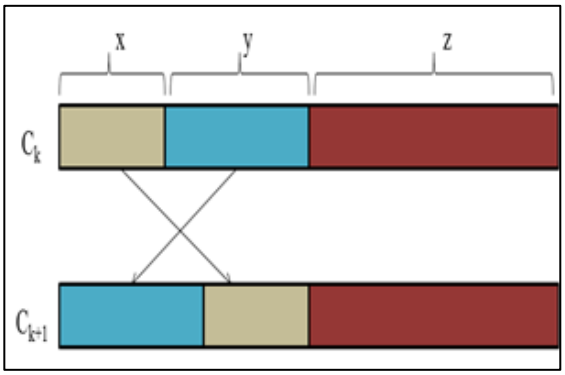

Fig. 4.*Left Swap operation*

The left swapped sequence Ck+1 is given by the equation as in (2).

$$C_{k+1} = L_{x,y}(C_k) \quad (2)$$

Fig. 5 pictorially shows the right swap operation. The right swapped sequence Ck+1 is given by the Equation.3

$$C_{k+1} = R_{x,y}(C_k) \quad (3)$$

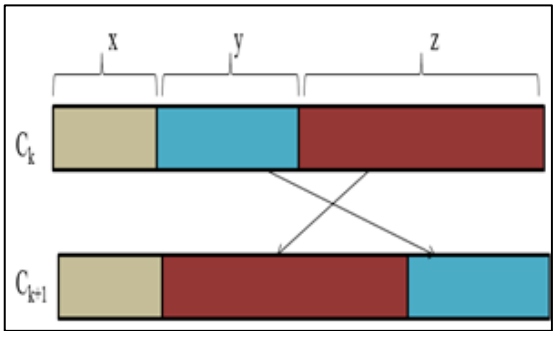

Fig. 5.*Right Swap operation*

7.2 AES with Counter Mode

The Advanced Encryption Standard (AES) is a symmetric key cryptographic algorithm developed for the encryption of electronic data stream. AES specification was established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES uses both substitution and permutation technique in each round of processing. It is efficient and faster to implement in both hardware and software. AES can be

implemented in variants with fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. The number of rounds in encryption is determined by the key size used. The numbers of cycles of repetition are as follows:

10 cycles of repetition for 128-bit keys.
12 cycles of repetition for 192-bit keys.
14 cycles of repetition for 256-bit keys.

Each round of encryption in AES has four sub processes namely SubBytes, ShiftRows, MixColumns and AddRoundKey to create cipher text from plaintext. To make available the single key for all rounds, the key is processed and divided into sub blocks for each round. The decryption process has the reverse of those four sub processes to regenerate plaintext from cipher text. AES can be converted into a Cryptographically Secure Pseudo Random Number Generator (CSPRNG) by running it in Counter mode (CTR). Here a counter is initialized to 0 and keeps on incrementing. The counter value is considered as plaintext and encrypted using a random key to generate the cipher text. The cipher text is now used as random stream of bits. This way AES in CTR mode ensures no identical cipher blocks are produced. The counter can also be initialized to any random value instead of 0. The counter is initialized to any arbitrary value. The random sequence generated by twining technique is broken down into 128-bit random keys. These random keys are used to encrypt the counter value plain text. The resulting 128-bit cipher text is used as cryptographically secure random numbers.

7 RESULTS OF THE PROPOSED METHOD

8.1Testing Randomness

The random numbers sequence generated must be tested to check for presence of any predictable repetitive patterns before being deployed in any application. Various randomness testing tools are available each focusing on variety of characteristics of random numbers. In this paper NIST statistical test suite [8] was used to measure the randomness.

8.2Test Results

Table. I show the NIST test suite results for random sequence generated using sensor data that are processed in wash-rinse-spin approach. Table. II show the NIST test suite results for random sequence generated using twining technique and AES with CTR mode.

TABLE 1
NIST TEST RESULTS FOR WASH-RINSE-SPIN APPROACH

STATISTICAL TEST	*P-VALUE*
Frequency	*0.734146*
Block Frequency	*0.734146*
Rank	*0.132004*
FFT	*0.613309*
Linear Complexity	*0.004532*

TABLE 2
NIST TEST RESULTS FOR TWINING TECHNIQUE AND AES WITH CTR MODE

STATISTICAL TEST	*P-VALUE*
Frequency	*0.739918*
Block Frequency	*0.911413*
Rank	*0.164678*
FFT	*0.687855*
Linear Complexity	*0.006756*

8.3Time Comparison

The times taken to process the sensor seeds and generate random numbers in wash-rinse-spin approach are measured. The times taken to perform twining technique and run AES with CTR mode to generate cryptographically secure random numbers are measured. The time measurements are compared in Table. III.

TABLE 3
TIME COMPARISON

S.NO	PROCESS	TIME TAKEN (in seconds)
	Wash-Rinse-Spin	
1.	Wash	3.090827
2.	Rinse	9.323631
3.	Spin	48.869937
	Total	61.284395
	Twining Technique	
1.	Wash	2.579656
2.	PRNG implementation	1.256333
3.	Twining Technique	0.281249
4.	AES with CTR Mode	6.964132
	Total	11.08137

The wash-rinse-spin approach consumes more time in rinse step and spin step. The twining technique uses simple swap operations to minimize collinearity and AES is time efficient algorithm to generate secure random numbers.

8 CONCLUSION AND FUTURE ENHANCEMENT

Results from the NIST Test Suite indicate that the use of novel twining technique minimizes collinearity among sensor seeds. The twined sequence has better randomness for usage as random key in AES with counter mode. The AES outputs cryptographically secure random numbers. These random numbers have higher P-value in NIST tests when compared to random numbers generated by wash-rinse-spin approach. The time consumption to generate random numbers has been greatly reduced by the use of simple swap operations and time efficient AES. In future research collinearity among sensor data housed on same environment can be removed by any randomization techniques that process more than three random sequences. There is also a scope to generate cryptographically secure random numbers by using stream ciphers in CTR mode replacing AES with CTR mode.

9 REFERENCES

[1] David Johnston, 'Random Number Generators – Principles and Practices', Walter De Gruyter GmbH & Co, 2018.

[2] Shah T, Upadhyay D, 'Design analysis of an n-Bit LFSR-based generic stream cipher and its implementation discussion on hardware and software platforms', Proceedings of the International Congress on Information and Communication Technology, 2016.

[3] Majid Babaei and Mohsen Farhadi, 'Introduction to Secure PRNGs', IET journal on Communications Network and System Sciences, 2011.
[4] Michele Feltz and Cas Cremers, 'Strengthening the security of authenticated key exchange against bad randomness', Designs, Codes and Cryptography, 2017.
[5] Jonathan Voris, Nitesh Saxena and Tzipora Halevi, 'Accelerometer and randomness: perfect together', Proceedings of the 4th ACM conference on Wireless Network Security, 2011.
[6] Hong S L and Liu C, 'Sensor-Based Random Number Generator Seeding', IEEE Access, 2015.
[7] Renault E, Boumerdassi S, 'Mutual Authentication Method for WSNs Based on the Three-Card Trick Ancient Card Game', Proceedings of IEEE 80th Vehicular Technology Conference, 2014.
[8] Rukhin A et al, 'A statistical test suite for random and pseudorandom number generators for cryptographic applications', National Institute of Standards and Technology, Gaithersburg, MD, USA, Technical Report, 2010.