International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Sensor-Seeded Cryptographically Secure Random Number Generation

K.Sathya, J.Premalatha, Vani Rajasekar

Abstract: ***Random numbers are essential to generate secret keys, initialization vector, one-time pads, sequence number for packets in network and many other applications. Though there are many Pseudo Random Number Generators available they are not suitable for highly secure applications that require high quality randomness. This paper proposes a cryptographically secure pseudorandom number generator with its entropy source from sensor housed on mobile devices. The sensor data are processed in 3-step approach to generate random sequence which in turn fed to Advanced Encryption Standard algorithm as random key to generate cryptographically secure random numbers.***

Keywords: Sensor-based random number, cryptographically secure, AES, 3-step approach, random key

I. INTRODUCTION

Information security algorithms aim to ensure the confidentiality, integrity and availability of data that is transmitted along the path from sender to receiver. All the security mechanisms like encryption, hashing, signature rely on random numbers in generating secret keys, one time pads, initialization vector, and nonce respectively. The random numbers are generated either from True Random Number Generator (TRNG) or Pseudo Random Number Generator (PRNG). TRNG uses entropy source that have natural randomness to produce random bit sequence. PRNG uses deterministic algorithm that takes a seed value to produce random sequence that are easily predictable. Hence PRNG require their seed to be more secure and unknown to attacker. Seed values can be obtained from any randomness rich sources like clock pulse, user activity, interrupts, time, etc.

To overcome the issue of predictability one can use sensor data as seeds for PRNG. Sensors are used to measure various physical parameters like pressure, temperature, motion, etc. These physical parameters exhibit randomness. Hence the values recorded by these sensors are a great source of random seeds for PRNG. The lack of security within the random number generation process would compromise the entire system as long as an attacker could easily infer the key. Hence the sensor source data must be processed before they are fed to PRNG that prevents attacker to gain adversarial control over the source of seed.

Manuscript published on 30 September 2019
* Correspondence Author
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.

Advanced Encryption Standard uses a symmetric key algorithm meaning same key is used for encryption and decryption. It uses 128-bit fixed length message block and 128-bit key with 10 cycles of repetition. AES is the widely used security algorithm that can be implemented easily on both hardware and software.

AES can be executed in various modes like Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR).

II. PRNG IMPLEMENTATIONS

PRNG requires the seed value to be fed to a deterministic algorithm. Many deterministic algorithms are proposed, some of widely used algorithms are

Blum-Blum-Shub algorithm uses $X_{n+1} = {X_n}^2 mod\ M$ where X_0 is seed value and M =pq, p and q are prime.

Linear congruential Generator uses $X_{n+1} = (aX_n + c) mod\ m$ where X_0 is the seed value.

Mersenne Prime Twister uses range of $2^{19937} - 1$, where $2^{19937} - 1$ that outputs sequence of 32-bit random numbers.

III. SENSOR-SEEDED PRNG

Some of PRNGs uses clock pulse of computer system as seeds. However they are obtainable by hacker making them vulnerable. The idea proposed is to use sensor data from mobile devices that has high source of entropy. The sensor data are collected from mobile device sensors and they are used as seeds in any one of the deterministic algorithm mentioned above. In our paper the accelerometer of Smartphone is used to collect data. An accelerometer is a sensor which measures the tilting motion and orientation of a mobile phone. The dataset collected is recorded value of accelerometer of a single day usage. Though they produce good quality randomness they are not enough secure in their implementation. When an attacker gains control of sensor or knows the sensor data the output of PRNG is compromised. Attacker may move the mobile in same orientation to generate the same seed value. Also the accelerometer data are available to the mobile's operating system which can be obtained by any malicious software.

IV. SECURING SENSOR SEEDS

To make the sensor data suitable as seeds for generating random sequences we must first remove the predictable patterns present in them and secondly resolving the collinearity problem that arise from multiple sensors housed on same device.

V. REMOVING PREDICTABLE PATTERNS

The predictable patterns in sensor data provide valuable information to attacker to control the movement patterns of mobile phone. These patterns are removed by 3 step approach namely

- Wash
- Rinse
- Spin

The above three steps can be represented in the Fig 1. below.

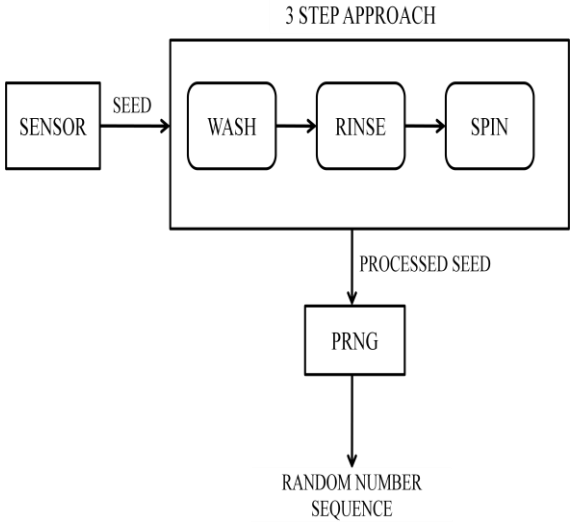

Fig.1 Schematic Diagram of Wash-Rinse-Spin

A. Wash

The sensor data used for this research was collected from Smartphone's accelerometer that measures the coordinates of phone in X-, Y- and Z-axis. These data are dependent upon the position of the mobile phone held by user and are easily predictable. These patterns are easily predictable and pose a risk. The recorded accelerometer data along X-axis is shown in Fig.2. Such collected sensor data are very weak source to produce random numbers. To remove those predictable patterns that are seen as slow drifts, the data are preprocessed. Preprocessing is done in such a way that those slow drifts are washed away. Such processed sensor data are less likely to be useful to hacker in determining the generated random numbers.

The Smartphone may be held idle for sometime or held at same position for sometime or periodically used in same position. These slow drifts causes slight variations in mean and variance of the sequence. Such variations lead to nonstationarity. The first step in processing is to remove those slow drifts by differentiating the data until all the nonstationarity are removed.

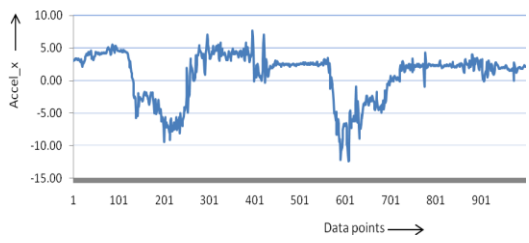

Fig.2 Raw Sensor Data

Differentiating the sensor data produces First order derivative that sufficiently removes nonstationarity. If nonstationarity still remains, differentiation can be repeated until threshold stationarity is obtained. The result is shown in Fig.3. As can be seen from the figure Fig.3, the sensor data shows absence of any minor drifts.

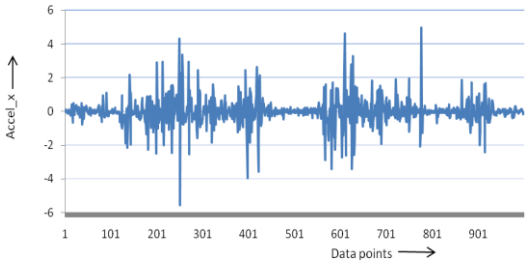

Fig.3 Washed Sensor Data

B. Rinse

Slow drifts are removed in the previous step by differentiation. Next step aims to dust away the soap traces in the washed data. The washed data are free from drifts however still contain predictable patterns and bare spots. They need to be removed completely by further inducing randomness in the data. The randomness is not introduced into the data directly. They are converted into complex numbers using Fast Fourier Transform with complex exponentials given by

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}}$$

The complex number has two parts namely real and imaginary. The real number denotes the sine wave magnitude. To add further randomness to data, the imaginary number is replaced by new random number. This modified complex numbers are converted back to sensor data i.e. from frequency domain to time domain using Inverse Fast Fourier Transform. This results in rinsed senor data represented in Fig.4. After rinse step, 8-bit integer is taken as seed value to generate the Pseudo random numbers.

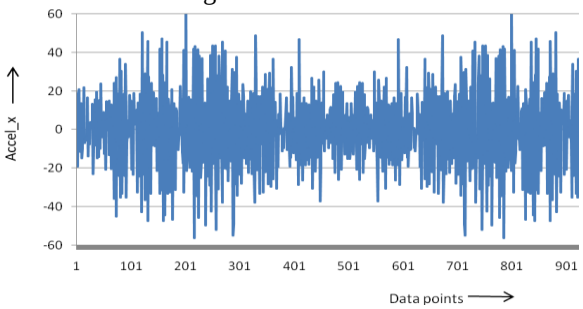

Fig.4 Rinsed Sensor Data

C. Spin

The data sequence with added randomness is produced in the rinse step. This is now used as seed for a RNG to produce required quantity of random numbers that find its application in password creation, key exchange, encryption, connection establishment, etc. The data sequence is broken down into sizes depending on the PRNG used. For our testing Linear congruential generator was used.

VI. CRYPTOGRAPHICALLY SECURE PRNG (CSPRNG)

The random bit sequence generated from above process may exhibit randomness but does not have good quality randomness to be used for highly secure applications. For example some applications require nonces to be unique. However encryption keys for secure application require it to be of good quality with rich entropy.

Hence there comes another class of generators called Cryptographically Secure Pseudo Random Number Generator (CSPRNG). A CSPRNG lengthens the available entropy over more number of bits.

A CSPRNG must satisfy the additional two requirements along with requirements of PRNG.

- Next bit test
- Withstand state compromise extensions

Next Bit test requires that given first k bits of random sequence it is infeasible for any polynomial time algorithm to determine the k+1th bit of random sequence.

CSPRNG must withstand state compromise extension where the current or previous state of generator is exposed to attacker, it is impossible to recreate sequence prior to revelation or future sequence to be generated.

Many PRNGs that pass statistical tests fail to pass these requirements making them unfit for cryptographic algorithms. One way of designing CSPRNG is to use a block cipher with counter mode.

VII. AES IN CTR MODE

AES can be implemented in any of modes- Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). AES uses 128-bit plaintext block and 128-bit, 192-bit, 256-bit key to generate highly secure cipher text.

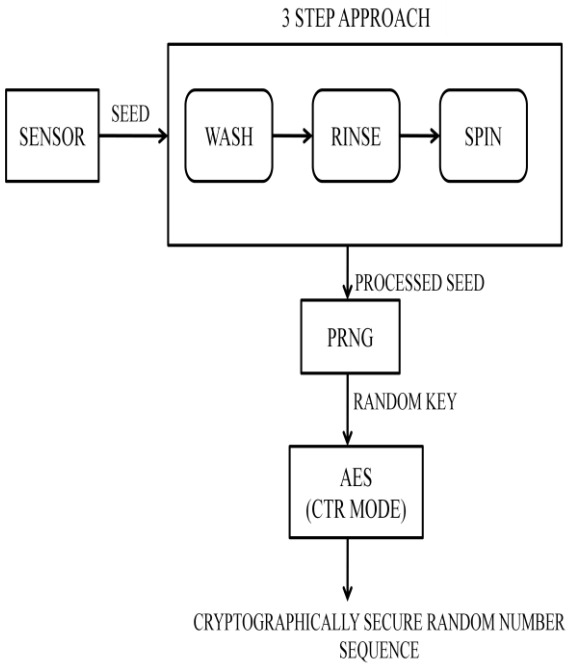

Fig.5 Schematic Diagram of Wash-Rinse-Spin with AES

Number of rounds of repetition depends on key size chosen. For this paper 128-bit key size is used with 10 cycle repetitions. To make our PRNG as CSPRNG that satisfy the two requirements, the random sequence generated from sensor data is used as random key in AES implementation with Counter mode. The counter can be initially set to any arbitrary value. After 10 cycles of repetition the cipher text produced can be used as random number that exhibits high-quality randomness. The random numbers generated have very high period length. AES without CTR mode can result in generating random numbers that may repeat after $2^{n/2}$ rounds. Running AES with Counter mode prevents this problem of producing identical random numbers since the counter value keeps on increasing or decreasing

VIII. NEED FOR TESTING RANDOMNESS

Various statistical tests are available that attempt to compare and evaluate the given sequence with a truly random sequence. It is necessary to test the random sequence before they are deployed in any application to ensure their randomness property is met. Randomness is a probabilistic property; that is, the properties of a random sequence can be characterized and described in terms of probability. The likely outcome of statistical tests, when applied to a truly random sequence, is known a priori and can be described in probabilistic terms. There are an infinite number of possible statistical tests, each assessing the presence or absence of a "pattern" which, if detected, would indicate that the sequence is nonrandom. Because there are so many tests for judging whether a sequence is random or not, no specific finite set of tests is deemed "complete." In addition, the results of statistical testing must be interpreted with some care and caution to avoid incorrect conclusions about a specific generator.

To measure the randomness guaranteed by wash-rinse-spin approach the seeds are tested by NIST (National Institute of Standards and Technology) statistical test suite for randomness. This test suite has fifteen different tests that measure various properties of randomness. For our analysis raw sensor data generated sequence, processed sensor data generated sequence, AES generated sequence are compared. These sequences are separately tested under NIST Test suite and results are compared. Though wash-rinse-spin approach and AES approach provide good results the former fails to pass the next bit test and state compromise extension withstanding.

IX. PERFORMANCE ANALYSIS

Table.1 Without Sensor Data

P-Value	Statistical Test
0.000000	Frequency
0.000000	Block Frequency
0.000000	Rank
0.017912	FFT
0.066882	Linear Complexity

Table.2 With Processed Sensor Data

P-Value	Statistical Test
0.742111	Frequency
0.742111	Block Frequency
0.000000	Rank
0.641233	FFT
0.000210	Linear Complexity

Table.3 With AES Generated Random Sequence

P-Value	Statistical Test
0.891224	Frequency
0.842311	Block Frequency
0.000000	Rank
0.645875	FFT
0.035425	Linear Complexity

X. CONCLUSION AND FUTURE WORK

The NIST test results for the random numbers produced by our processed sensor data shows better probability over the random numbers generated by raw sensor data. This ensures that the wash-rinse-spin processing of sensor data are reliable and less vulnerable to be predicted. Raw sensor data have very less entropy and are inadequate to produce secure random numbers. More cyberattacks are launched against mobile devices with the growth of Internet. This technique leverages the weakness by processing and enriching the entropy of sensor data. Such processed sensor data are more suitable for cryptographically secure random number generation. Using AES in counter mode in this approach produces much stringer random numbers as AES still remains unbreakable.

The approach uses only measurement data from single sensor. But practically there are many sensors housed in the environments. The future works may consider collecting and processing multiple sensor data for better improvement. The rinse step involving Fast Fourier Transform can be replaced by any other better technique in inducing the randomness to the sequence. Replacing AES with any other block cipher that guarantees hardness to break can improve the results further.

In summary this approach generates very high secure good quality random sequence that is suitable for applications requiring robustness. Also AES can be implemented in hardware to enhance efficiency of generating random numbers.

CONFLICTS OF INTEREST

The authors declare here that they have no conflict of interest.

FUNDING STATEMENT

The authors declare here that they got no funding for this research work.

REFERENCES

1. S.L. Hong, C. Liu(2015), Sensor-Based Random Number Generator Seeding, IEEE Access.
2. M.Matsumoto and T.Nishimura(1998),Mersenne twister:A623-dimensionally equidistributed uniform pseudo-random number generator, ACM Transactions on Modeling and Computer Simulation.
3. B.Schneier(1996), Applied Cryptography, Addison-Wesley, Reading.
4. L.Blum, M.Blum, and M.Shub,(1986), A simple unpredictable pseudo random number generator, SIAM Journal of Computing.
5. A.Francillon and C.Castellccia(2007), TinyRNG: A Cryptographic Random Number Generator for Wireless Sensors Network Nodes, 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops.
6. A.Rukhin et al.(2010), A statistical test suite for random and pseudorandom number generators for cryptographic applications, National Institute of Standards and Technology, Gaithersburg, MD, USA.
7. S.K.Park and K.W.Miller(1988), Random number generators: Good ones are hard to find, Commun. ACM.
8. A.L.Rukhin(2001), Testing randomness: A suite of statistical procedures, Theory of Probability and Applications.
9. B.Sunar, W.Martin, and D.Stinson(2007), A provably secure true random number generator to active attack, IEEE transactions on computers.
10. C.H.Vincent(1970), The Generation of Truly Random Binary Numbers, Journal of Physics
11. C.Bosley and Y.Dodis(2007), Does privacy require true randomness?, Proceedings of 4th International Conference on Theory Cryptography

AUTHORS PROFILE

K.Sathya completed her B.Tech(IT), M.Tech (Information and cyberwarefare) in dept of IT Kongu engineering college. She is pursuing her PhD (Information and Communication Engineering) in the area of Network security. Presently she is working as assistant professor in the department of CT/UG Kongu engineering college for past 3 years. Her area of interest includes Network security, Computer networks.

Dr.J.Premalatha completed BE(ECE), ME(CSE), PhD(Information and Communication Engineering). She is working as professor in the department of IT Kongu engineering college. Her teaching experience is 28 years. Her area of interest includes Network security, Cryptography, Computer networks and Database Management system.

Vani Rajasekar completed her B.Tech(IT), M.Tech (Information and cyberwarefare) in department of IT Kongu engineering college. She is pursuing her PhD (Information and Communication Engineering) in the area of Network security. Presently she is working as assistant professor in the department of CSE Kongu engineering college for past 3 years. Her area of interest includes Network security, Cryptography and Wireless networks.