Indonesian Journal of Electrical Engineering and Computer Science
Vol. 12, No. 1, October 2018, pp. 254~260
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v12.i1.pp254-260

Randomize IPv6 Stateless Address Autoconfiguration in None-stable Storage Arduino Devices

Zolidah Kasiran, Rizaki Napi
Faculty of Computer & Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Malaysia

Article Info

Article history:

Received Jun 1, 2018
Revised Jul 10, 2018
Accepted Jul 25, 2018

Keywords:

Internet of Thing
Stateless Address Configuration
IPv6

ABSTRACT

The pervasiveness of IoT device requires each device to have a unique address number in order to communicate. Internet Standard specified in RFC4941-Privacy Extension for SLAAC had question raised on the randomness of the IPv6 address generated due to the shortcoming of device random generator algorithm. Improvements to the RFC's have been proposed and Arduino Uno had been selected as IoT platform since it currently supports IPv6 implementation. An enhancement algorithm was developed. The generated IPv6 address is then tested against ENT Random Test tool for observation and conclusion.

Corresponding Author:

Zolidah Kasiran,
Faculty of Computer & Mathematical Sciences,
Universiti Teknologi MARA,
40450 Shah Alam, Malaysia.
Email: zolidah@tmsk.uitm.edu.my

1. INTRODUCTION

The IoT ecosystem can be composed from smart object with physical hardware limitation (Low memory & processing power, limited energy and communication capabilities), and also identifications system that uniquely identify thing globally. Wireless Communication Technology such as 6LoWPAN for wireless sensor, Bluetooth Low Energy (BLE) and Wi-Fi low power implemented together with identification system such as Electronic Product Code (EPC) and Near Field Communication (NFC) [1].

The new concept of extending internet is feasible with the implementation of IPv6. IPv6 offer wide range of addressing space compare to IP v4 that allow all the connected device to connect directly and secure to internet. IPv6 is design to provide mobility and secure communication not only to user but devices attached to anything, not only user. The ability of IPv6 in providing features such security, scalability, flexibility, ubiquitous, open and end-to-end connectivity to the internet [2] is considered the most suitable technology for the IoT.

IPv6 Addressing as stated in the RFC 4941 - Privacy Extensions for Stateless Address Auto configuration (SLAAC) in IPv6 [3] dictate the use of interface ID as way for device automatically assign IPv6 addresses to itself. This use of fix part of address can be traced back to the device. By tracking the assign IPv6 address to the device could enable the movement and activities of the device to be recorded. As a result, there is requirements for device to regularly generate unique IPv6 addresses and the pair of generated IPv6 address by the same device must be unlinkable [4]. The mechanism to generate the IPv6 address by the RFC 4941- Privacy Extension for SLAAC require the availability of either high quality random bit streams or long-term state (or both) in the device. This addressing technique require a stable storage to store the random bits but small IoT devices may lack of stable storage. Thus, this project proposed a scheme that will allow none-stable storage device to be able to generate randomised address bits.

In recent years many works have been carried out pertaining the Internet of Thing in various angle such as the technology, addressing scheme and the security espect of it.

1.1. Addressing Scheme And IOT Security

As an evolution from RFID and sensor network, the IoT devices currently connected and integrate multiple technologies. The IoT that integrating multiple technology enable user to bring the physical hardware into the cyber/virtual world. Integrating Internet Protocol (IP), Electronic Product Code (EPC), Barcode, Wireless Fidelity (Wi-Fi), Bluetooth, ZigBee, Near Field Communication (NFC) and Wireless Sensor Network meet up the requirements for network layer of IoT architecture for connection to the internet [5].

An addressing scheme that would be able to generates a unique address for each device is important in order to identify each device that are connected to the internet. There is currently two addressing scheme that are commonly used in identifying IoT devices. Namely Electronic Product Code (EPC) and IPv6, it can uniquely assign billion of address to any connected devices [6].

Few addressing scheme in IPv6 research have been carried out such as the variance of Double Address Detection (DAD) [7-8], Low-Energy ADdress Allocation Scheme (LEADS), Spatial IP Address(SIPA) and CLOSA (CLOck Skew Addressing Scheme) [9].

1.2. Randomness

The need for random numbers comes from various Cryptographic application. Cryptographic application use random bit number is various point. A random bit number can be interpreted as result of flipping a coin with side label 1 or 0. The probability each flip generating 0 or 1 is ½ or 50% each time. The current result does not affect the future generated number. This is the perfect example of Random number generator, where the next number cannot be predicted [10].

There is two type of random number generation. The first type is Hardware base Random Number Generator (RNG) or True RNG, where the source of entropy is a non-deterministic source (i.e. Electrical Noise from Microprocessor Pin). The output of RNG can be use directly or as a feed to pseudorandom generator.

The pseudorandom number generator (PRNG) use one or multiple source of input in generating random number. The input source is called seeds. The seeds itself must be random and unpredictable [11-12]. A method has been proposed the for the clients to generate IPv6 address interface identifiers through hashing the IPv6 prefix advertised by the router advertisement [13].

1.3. Cryptographically Generated Address (CGA)

CGAs had define a method for securely associating a public key to an IPv6 address. A cryptographic hash of the public key is used as an interface identifier of the IPv6 address, which can later be verified by the recipient of the packet or message. Since CGAs tie a public key to an IPv6 address, even as hosts switch networks, they are uniquely identifiable [14]. The CGA implementation increase the security of IPv6 and produce random ID for IPv6 addressing. But CGA implementation has its own drawback. The cost of running CGA algorithm is high [15]. It uses high computational power and generally it would generate similar keys. The method is not suitable for IoT devices with low compututional power.

1.4. Secured Neighbor Discovery (SEND)

To counter the possible attack against NDP and SLAAC, Secured Neighbour Discover (SEND) is develop as a more secured way to implement NDP. With SEND it provide address ownership proof mechanism, message integrity/identity and authorization of router. SEND use Cryptographically Generated Address (CGA) as part of it component in generating secured message and identity for IPv6 communication. SEND protocol use CGA to prevent address spoofing. CGA can authenticate IPv6 address without installing addition 3rd party software and servers [16-17].

2. RESEARCH METHOD

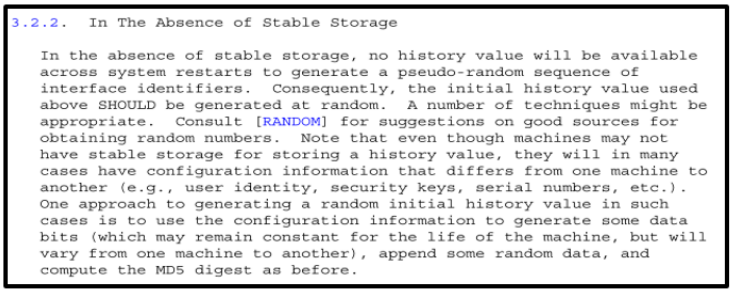
The purpose of this study is to enhance randomize algorithm in order to solve the problem with small IoT device that has no stable storage to store the generated random IPv6 address number. The algorithms to generate IPv6 address were developed based on specification in RFC 4941- Privacy Extension for SLAAC [3] and algorithm proposed by[14]. The default random engine used by the IoT device is used to generate the IPv6 address.

This study had adapted and enhanced three algorithms for IPv6 Address generation as followings:

1. RFC4941-Privacy Extension for SLAAC[3]– Random IP address Generate using random engine come with operating system.
2. Improvement to the algorithm 1 above by applying MD5 hashing mechanism to the random IPv6 address.
3. Rafiee-Meinel – Based on the algorithm recommended by Rafiee-Meinel with adjustment to comply with hardware limitation. Researchers [4], [15,] have recommended improvement in the implementation of Cryptographically Generated Address (CGA) by reducing the granularity of security level to lower the cost of computation. Another researcher recommends a solution in generating a random number and combine with a MD5 hashing mechanism of device interface to generate a new random IPv6 address [18].

2.1. Algorithm 1: Random IPv6 address generation based on method 2 RFC4941-Privacy Extension of SLAAC.

The algorithm 1 is based on method 2 for implementing RFC4941 – Privacy Extension for IPv6 SLAAC as in Figure 1.



3.2.2. In The Absence of Stable Storage

In the absence of stable storage, no history value will be available across system restarts to generate a pseudo-random sequence of interface identifiers. Consequently, the initial history value used above SHOULD be generated at random. A number of techniques might be appropriate. Consult [RANDOM] for suggestions on good sources for obtaining random numbers. Note that even though machines may not have stable storage for storing a history value, they will in many cases have configuration information that differs from one machine to another (e.g., user identity, security keys, serial numbers, etc.). One approach to generating a random initial history value in such cases is to use the configuration information to generate some data bits (which may remain constant for the life of the machine, but will vary from one machine to another), append some random data, and compute the MD5 digest as before.

Figure 1. RFC4941-Privacy Extensions for IPv6 SLAAC

Arduino device can generate a random number using it build in random generator using the random ( ) feature of the software IDE. The device random generator engine was using Arduino PIN number 1 value as the source of random seed when generating a random number.

The process started by generating 4 set of random address using default random engine generator based on the 16-bit max value per address set. The algorithm works as follows:

a) 4 set of Random number is generated in Hexadecimal value.
b) The set's is then combined with network bit to generate a full IPv6 address.
c) The IPv6 address is then applied to the network interface.

2.2. Algorithm 2: Improvement to Algorithm 1

Algorithm 2 is an improvement to algorithm 1 where MD5 hash process is added. The process started by generating 4 set of random address as in algorithm 1 using default random engine generator. The output is then Hashes using MD5 before it is combine with network address to form full IPv6 address. The IPv6 address is then applied to the network interface. The process flow chart is as in Figure 2.

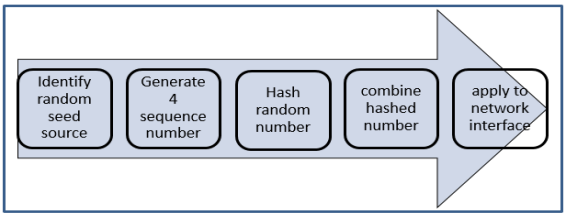


Figure 2. Algorithm 2- Generating IPv6 Address from Random Hash Value

2.3. Algorithm 3: Improvement to Rafiee-Meinel Proposed Algorithm

The algorithm 3 was adapted from algorithm suggested by [15]. The approach by Rafiee-Meinel suggested to add other source of random value in the IPv6 address generation process. It did not depend only to the device default random generator. The value is then concatenate and Hash using SHA 256.

Improvement to the approach by Rafiee-Meinel was the change of Hashing engine. The Hashing engine use by [15] was SHA256. To improve the algorithm MD5 was selected as the new Hashing engine. Reason for MD5 selection as the Hashing engine is due to the followings.

a) MD5 is faster than SHA256 in generating hash value [19].
b) SHA256 cannot run on Arduino Uno due to controller hardware limitation.
c) MD5 is also use as default IPv6 Hash function by the RFC4941 – Privacy Extension for SLAAC and used the following as input values:
   1) 128-bit random value.
   2) Timestamp from RTC
   3) 64-bit Network Address value

V=h(R||T||P)
Where J = leftmost 64 bit of V
*J is the host IID portion of the IPv6 Address
R= 128bit hexadecimal random number.
T = Timestamp
P = 64-bit Network Address
h = MD5 Hash function *- modification from the original SHA256 Hash function

The sequences of the algorithm are as follow:

1. Generate R: 128-bit hexadecimal random value by using analog read PIN 1 as seed values.
2. Get T: Timestamp from DS1302 modules – format hhmmss. The value is converted to millisecond before hashing. (hh x 32000, mm x 6000, ss x 1000)
3. Get P: RA@Network address.
4. Appending all input (R, T, P) as one long string.
5. Compute result 4 using MD5 Hashes. Output as Hex.
6. Take the 64 leftmost bits to be used for host bit IPV6.
7. Combine the number with IPv6 network address. Currently IPv6 network address is static assign.
8. Apply the generated IPv6 Address to network interface.
9. Test the connection workability. Ping test.

3. RESULT AND ANALYSIS

The result of experiment was evaluated using FormiLab.Ch. This entropy test tool was developed by John Walter, the co-founder of AutoDesk, Inc. The tool applies various tests to sequences of bytes stored in files and reports the results of those tests. Amongst the test perform by the tool is Entropy test, Arithmetic mean and Monte Carlo value for Pi. Detail of the test is described in the methodology section. The program is useful for evaluating pseudorandom number generators for encryption and statistical sampling applications and compression algorithms [20].

The test output was compared against normal value as defined in Table 1.

Table 1. ENT Comparative Result Value

Test	Comparative result
Entropy	8 bit max
Compression	
Arithmetic mean value of data bytes is	Closes to 127.5
Monte Carlo value for Pi is	Closes to 3.14159265

("Entropy and Random Number Generators," 2013)

3.1. Observation of the Test Result for 64-Bit Random Number

The experiment has shown that the randomness of the host portion of IPv6 address generated scored high of all 3-test set by the ENT test tool when compared to the expected comparative value. It can be concluded that the Host IPv6 address generated by the algorithm is random with Algorithm 1 scores less than algorithm 2 and algorithm 3.

Algorithm 1 is the standard recommendation algorithm proposed by RFC4941- Privacy Extension for SLAAC. Entropy test addressed the density of the test file used for the test. A compress file such as a JPEG image scores high for entropy test because the file is already compress and no repetition of data in the file.

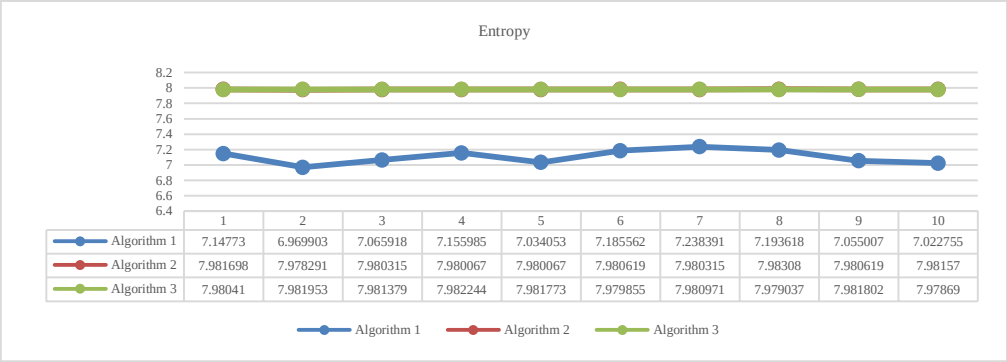


Figure 3. Ipv6 Address Entropy Test Result Comparison Value

Overall, the entropy test result for 64-bit IPv6 score is almost reach the comparative value of 8-bit with average entropy of 7.0246926 for algorithm 1 as shown in Figure 3.

Algorithm 2 & algorithm 3 entropy test result is also score high with the value of 7.9808114 & 7.9810166 for algorithm 2 & Algorithm 3 respectively. Algorithm 1 is expected to score lower then algorithm 1 and algorithm 2 since is depend only on default Arduino random function and MD5 hash to generate random Ipv6 address.

Arithmetic mean is simply the result of summing the all the bytes in the file and dividing by the file length. If the data are close to random, this should be about 127.5. For 64-bit IPv6 address the result of arithmetic mean around 121 to 128 for all the algorithms. The nearer the test result to the comparative value of 127.5 is the more random the IPv6 address. Like the entropy result, Algorithm 1 score is further from the comparative value than algorithm 2 and algorithm 3. Table 2 shows the comparative value of the Arithmetic Mean Test.

Table 2. ENT Result Output for 64-bit (Algorithm1 to Algorithm 3)

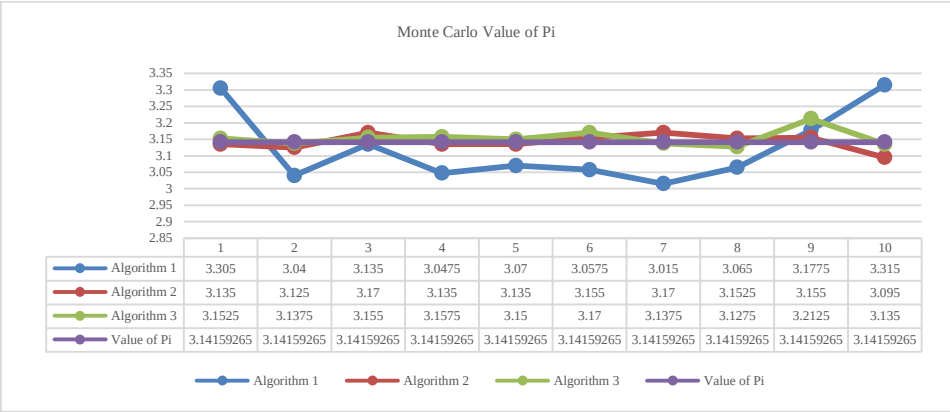
	Comparative value	Algorithm1	Algorithm2	Algorithm3
Entropy	8 bit max	7.0246926	7.9806641	7.9808114
Compression		12%	0%	0
Arithmetic mean value of data bytes is	127.5	124.14702	127.72758	127.36148
Monte Carlo value for Pi is	3.14159265	3.17925	3.14275	3.1535

Monte Carlo value for Pi was calculated by plotting 6 successive sequence of byte into 24bits X and Y axis coordinates within a square. Apercentage of the random generated Hit point within the square was used to calculate the value of Pi.

Figure 4 shows the Pi value generated by the algorithm is nearer to the value of Pi (3.14159265) and its indicate the number generated is random. The Pi value for generated by Algorithm 1 has an average of 3.17925. Meanwhile the Pi value generated by Algorithm 2 and Algorithm 3 have average of 3.14275 and 3.1535 subsequently. As comparison A Monte Carlo value for Pi for a 500000-byte file created by radioactive decay, (very random) is 3.143580574 (error of 0.06 percent).

It has been observed that the algorithm scores high (random) for 64-bit IPv6 address using the ENT test tool lead to the following conclusion.

1. The default Arduino random engine generator was generating IPv6 address that consider fair in ENT test when used to generate random IPv6 address number. The result was expected based on previous work that prove Arduino should not be independently rely on as True Hardware Random Generator. The combination of Arduino as PRNG and other mechanism such as MD5 had shown an evidence as able to increase the randomness of generated IPv6 address.
2. The uses of MD5 for hashing mechanism does not affect the ability to generate random IPv6 address. This has been proven in the test result for random IPv6 address by algorithm 2 and algorithm 3. Both algorithm generated IPv6 address score high in the random test score even when it is using MD5 hashing mechanism.



	1	2	3	4	5	6	7	8	9	10
Algorithm 1	3.305	3.04	3.135	3.0475	3.07	3.0575	3.015	3.065	3.1775	3.315
Algorithm 2	3.135	3.125	3.17	3.135	3.135	3.155	3.17	3.1525	3.155	3.095
Algorithm 3	3.1525	3.1375	3.155	3.1575	3.15	3.17	3.1375	3.1275	3.2125	3.135
Value of Pi	3.14159265	3.14159265	3.14159265	3.14159265	3.14159265	3.14159265	3.14159265	3.14159265	3.14159265	3.14159265

Figure 4. Ipv6 Address Monte Carlo Value of Pi Comparison Value

4. CONCLUSION

The use of MD5 as the hashing mechanism has its own advantages over it weaknesses. MD5 has been proven in other works as is not a collision resistance. Other device with the same configuration generating a collision address is possible with the use of MD5. But when looking into the unlinkable requirements of RFC4941- Privacy Extension for IPv6, it means that when other device have the possibility to produce the same IPv6 address it may link the address to other device. Hence the IPv6 address generated is unlinkable to the device that generate the IPv6.

RFC 4941 – Privacy Extension for IPv6 SLAAC[3] objective is to protect individual activity from being link back to user and the device that the user use, hence violate the user privacy. Even when IoT devices which is more to machine to machine interaction have less user privacy violation challenges, it is always a possibility it can be trace back to user activity.

ACKNOWLEDGEMENT

This research was financially supported by the Ministry of Higher Education through Fundamental Research Grant Scheme 600-RMI/FRGS 5/3(0006/2016).

REFERENCES

[1] Julien Montavont, Cosmin Cobarzan, Thomas Noel. "Theoretical analysis of IPv6 stateless address autoconfiguration in Low-power and Lossy Wireless Networks", *International Conference on Computing & Communication Technologies Research, Innovation and Vision for Future*, 2015, 198-203.
[2] Jara, A. J., Ladid, L., & Skarmeta, "The Internet of Everything through IPv6: An Analysis of Challenges, Solutions and Opportunities". *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 2013; 4(3), 97–118.
[3] IETF Standard, RFC 4941, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC, 2007
[4] AlSa'deh A., Rafiee H., Meinel C; "IPv6 Stateless Address Autoconfiguration: Balancing between Security, Privacy and Usability". *Lecture Notes in Computer Science*, 2013; vol 7743. Springer, Berlin, Heidelberg
[5] Madakam, S., Ramaswamy, R., & Tripathi, S. "Internet of Things (IoT): A Literature Review". *Journal of Computer and Communications*,2015; 3(3), 164–173.
[6] Rui Ma,Yue Liu, Chuan Shan, Xiao Lin Zhao, Xu An Wang. "Research on Identification and Adressing of the Internet of Things". *10th International Conference on P2P, Parallel Grid, Cloud and Internet Computing*, 2015
[7] Chi-Chien Liu, Huai-Jen Liu,Chih-Hu Wang, "Performance Improvement on Duplicate Address" , *8th International Conference on Ubi-Media Computing*, 2015, 029-032
[8] Dana Blouin; Steven Gordon. "A proxy-based Passive Duplicate Address Detection protocol for IPv6 wireless sensor networks", *IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2015; 248 - 253
[9] H.Zormati, J.Chebil, J.Bel Hadj Taher. "Addressing Scheme for IoT Network Using IPv6", *International Journal of Electronics and Communication Engineering*, 2015; Vol 11 No 4.
[10] Rukhin, A., Soto, J., Nechvatal, J., Miles, S., Barker, E., Leigh, S., … Vo, S.. "A statistical test suite for random and pseudorandom number generators for cryptographic applications". *National Institute of Standards and Technology*, 2010
[11] Xuejing Kang, Zhao Han, Aiwei Yu; Peiqi Duan. "Double random scrambling encoding in the RPMPFrHT domain", *IEEE International Conference on Image Processing (ICIP)*,Beijing China, 2017; 4362 - 4366

[12] Mohd Izuan Mohd Saad, Kamarularifin Abd Jalil And Mazani, "Preserving User Privacy With Anonymous Authentication In Cloud Computing", *ARPN Journal Of Engineering And Applied Sciences*, 2015; 10(23),pp 17937-17944
[13] Barrera, D., Wurster, G., & van Oorschot, P. C. "Back to the Future: Revisiting IPv6 Privacy Extensions". *The Advanced Computing Systems Association*, 2010; vol 36, No 1
[14] Sana Qadir; Mohammad Umar Siddiqi, "An Investigation of Cryptographically Generated Address (CGA) Based Authentication for Mobile IPv6". *International Conference on Computer and Communication Engineering*, 2014; 300-303
[15] Rafiee, H., & Meinel. "Privacy and Security in IPv6 Networks : Challenges and Possible Solutions". *6th International Conference on Security of Information and Networks*, Turkey, 2013;218-224
[16] Amjed Sid Ahmed, Rosilah Hassan, Nor Effendy Othman. "Secure Neighbor Discovery (SeND): Attacks and Challenges", *6th International Conference on Electrical Engineering and Informatics (ICEEI)*,2017; 1-6
[17] Sumathi P; Saroj Patel Prabhakaran. "Secure Neighbor Discovery (SEND) Protocol challenges and approaches" , *10th International Conference on Intelligent Systems and Control (ISCO)* 2016; 1-6
[18] Supriyanto P, Hasbullah E.H, Kadhum M. "Security Mechanism for IPv6 Stateless Address Autoconfiguration". *International Conference on Automation, Cognitive Science, Optic, Micro Electro-Mechanical System and Information Technology*, Bandung,2015; 31-36.
[19] Gupta, P., & Kumar, S. (2014). "A Comparative Analysis of SHA and MD5 Algorithm". *International Journal of Computer Science and Information Technologies (IJCSIT)*, 5(3), 4492–4495.
[20] José Campos; Rui Abreu; Gordon Fraser; Marcelo d'Amorim, "Entropy-based test generation for improved fault localization", *28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2013; 257-267.