ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

Study and Development of TB-CGA Based Address Generation for IPv6

Priyanka Singh Baghel[1], Prof. Satpal Singh[2], Prof. Sumit Nema

M.Tech Student, Department of Computer Engineering, Global Engineering College, Jabalpur, India[1]

Assistant Professor, Department of Computer Engineering Global Engineering College, Jabalpur, India[2]

Assistant Professor and Head of the Department, Department of Computer Engineering Global Engineering College, Jabalpur, India[3]

ABSTRACT: An authentication method that is used in IPv6 network to authenticate the address proof of the owner is CGA (Cryptographically Generated Address). The identification of address of the owner is authenticated by RSA public key that is of variable length.Secure hash algorithms (SHA-1) have been used in this literature to provide security. In this algorithm an output of 160 bits is generated. Among these 160 bits' certain number of bits from the left-hand side has been selected as per our requirement in the Hash-1 and Hash-2 extension. The interface identifier part of a particular address is changed after a certain time using the method of CGA and brand-new address will be generated from the interface of the identifier.To use the method of CGA the node which is on sender's side would have to choose the parameter of security SEC.Input will be 128-bit modifier, RSA public key which is not of fixed length i.e.; 64-bit subnet prefix and 8-bit impact result for the uniqueness of the location that is created. For providing the security we used Secure Hash Algorithm-1 (output is 160 bits). We have also used the security parameter (3 bit), which states the security level of CGA against the Brute force attack. The output of the Time Based CGA is secure in IPv6 address using CGA including the Hash 1, modifier identification and Interface identification.We introduced a handy and programmed path to opt parameter of security related to CGA calculation. . Thus, factors determining the time taken by CGA Algorithm is crucial to evaluate t. The Factors include Sec value, Hash function. RSA must be restored by a public key cryptosystem that provides parallel cryptographic strength but has a faster key generation, shorter key length and less expensive signature generation and verification. When it is attained, CGA-based Authentication can be computationally realizable for mobile environment.

KEYWORDS: CGA algorithm; Time Based-Cryptographically Generated Address; SHA; Interfaceidentifier(IID);

I. INTRODUCTION

In the IPv6 address, 64-bits from the left of the address of 128 bits form the subnet prefix and 64 bits from the right forms the interface identifier [3]. IPv6 consist of 128 bits as compared to 32 bits in IPv4. With an aid of IPv6 address it is now possible to support 2^{128} unique IP addresses.

The principle objective for effective move is to permit IPv6 and IPv4 hosts for interaction. IPsec support is Optional in IPv4. But in IPv6 it is a requirement not an option. IPsec provides a combined set of cryptographic protocols which gives the security to data communication and security to exchange keys. The interface identifier segment and the dissection of IPv6 locations into particular topology raises a new hindrance to IPv6 in that an amended part of an IPv6 address (i.e. interface identifier) can contain an identifier that remaining parts steady despite when the topology segment of a location changes .In this manner, the interface identifier installed inside of a location could be utilized to track exercises of an individual, even as they move topologically inside of the web. The current system used to anticipate hub following, and, thus, to secure a client's protection in the system layer, is to change the hub's interface ID address. Most familiar way of setting IID is hinged on SLAAC implant's a network device Ethernet

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

Media Access Control (MAC) address into an IPv6 address. Since each MAC location is extraordinary, IPv6 could permit gadgets to be universally exceptionally recognized. Tragically, this distinctive property can consider following of an individual gadget accordingly abusing the client's privacy [6]. One such proposal is CGAs (Cryptographically Generated Addresses), RFC3972; random identifier is generated based on the node's public.

Cryptographically Generated Addresses (*CGAs*)[4], IPv6 addresses affirmations are provided by CGA and keep acrimonious hubs from declaring the authority for others addresses. It grows the computational cost for both the assailant and the area generator. The area generator needs, all things considered, 2^(16×Sec) brute force chase to satisfy Hash2 condition, i.e., 16×Sec-uttermost left bit of Hash2 equal to zero. Gigantic Sec quality may provoke critical and undesirable area time delay. Subsequently, the high retribution cost of CGA may keep its utilization and leave IPv6 frameworks exposed against a couple attacks which are related to area taken.

II. RELATED WORK

Cryptographically Generated Addresses [4], CGA are expected to offer the acceptance to IPv6 addresses and keep pernicious centre points from declaring the obligation regarding others addresses. CGA are planned to offer the approval to IPv6 addresses and keep malignant centres from stating the obligation regarding others' addresses. Along these lines, IPv6 area of the centre point is certain to its open key. Thusly, CGA is self-ensuring since it doesn't rely on upon other force, RFC4941. CGA affirms the character of the sender in perspective of open key cryptography. The recipient has the limit set up that the message begins from a veritable sender. The message which is sent from CGA area is stamped with the area proprietor private key and general society key is attached to the checked message. Since the message contains everything the recipient needs to affirm, the beneficiary does not need to have further exchanges with the sender for completing the approval procedure.

O'Shea and Roe [16], proposed the idea of cryptographically generated address. They used it in child proof authentication for MIPv6 (CAM). In their approach they added the interface ID portion of IPv6 address to the owner's public key. Later, the extended version of CAM is suggested by Nikander [17], he added some random data to hash input.

Aura [14], proposed the final model of CGA and his work was standardized in RFC 3972[15]. Castelluccia and Montenegro [18], proposed the same approach for Mobile IPv6. Main difference between Aura's proposal [14] and others was the induction of has extension.

P.C. van Oorschot and David Barrera [19] explained the state of IPv6 support using a tool named as security visualization and also defined the difficulties which were coming while supporting new protocols. They utilized a separating technique that aides reduce the impediment of IPv6 sources on charts.

S. Albert Rabara et al [20], they proposed a freshly configured IPv6 address and they called it as P46CGA, which incorporates the extension to IPv6 cryptographic methods, stateless addressing mechanism, cryptographic methods and IPv4 router address. To establish a connection between a router and the current location of IPV6 nodes, they used the IPv4 router address in IPv6 addressing. The focus of their proposal was to enables an IPv6 mobile node to also roam into IPV4 network.

User's privacy and protection was first given in Privacy Extension RFC [13]. That RFC unfortunately encompasses mistakes which compromises the user's privacy. To solve its problems, Christoph Meinel and Hosnieh Rafiee [21] proposed a new algorithm which maintains the lifetime of the nodes and also gives a method to produce a random interface ID (IID).

Ahmad AlSa'deh et al [22] give an analysis of security and defined the possible ways to attack CGA. They have given that the verification process of CGA is vulnerable to Denial-of –Service (DoS) attack. A CGA standard verification algorithm was proposed by them to mitigate the privacy related attack and DoS attack.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

Cheneaua et al [23], proposed an enhanced technique to generate CGA using Elliptic Curve Cryptograph (ECC) keys rather than RSA keys which are standardized. As compare to RSA, ECC has shorter key which assures smaller packet size. Appropriately, the utilization of Elliptic Curve Cryptography in CGAs can be more particularly suitable in resource limited devices.

C. Castelluccia et al [24] defines the Cryptographically Generated IPv6 addresses concept. CGA addresses are useful to secure operation redirect in many protocols and also proposed to give a solution to the IPv6 address ownership problem.

III. PROPOSED ALGORITHM

A. *Algorithm for CGA generation:*

- CGA era Algorithm starts with determining the location owner's open key and selecting the foremost possible Security quality.
- The Hash2 [4] processing circle then moves until locating the last modifier. The Hash2 worth is a hash of the combination of the modifier and the public key which are linked with a zero-quality for collision count and subnet prefix.
- The location generator tries diverse estimations of the modifier till 16×Sec-furthest left bits of Hash2 turn into zero.
- When a suit is determined, the circle for the Hash2 computing ends. At that point the last Modifier worth is provided and used as a data for the Hash1 processing.
- The Hash1 [4] quality is a hash of the combination of all CGA parameters. By composing the estimation of sec into three left-most bits and by setting bits 6 and 7 (i.e. , "u " and "g" bits) to zero ,the Interface identifier is then received from Hash1.
- Finally, the DAD algorithm is keep running by the customer to guarantee that the location is extraordinary inside of the same subnet. On the off chance that a location crash happens, increase the Collison Count and register Hash1 again to get the IID. On the other side, after three crashes, CGA calculation stops and reports a blunder.

B. *Description of the Proposed Algorithm:*

We are introducing to modify the CGA addresses intermittently to safeguard the users' privacy. Each CGA address has an allied lifetime that determines how long the address is linked to an interface [11]. Once the lifetime ends, the CGA address is execrated. While a CGA location is in a belittled state, its utilization is discouraged, yet not entirely illegal. New correspondence (e.g., the opening of another TCP association) ought to utilize another CGA address when conceivable.

There are many parameters on which the lifetime of a temporary CGA address depends. For example, the time which is required by the host to produce the new CGA address should depend upon. To break the generated CGA address the attacker will require the time.

When a host joins new subnet. For this situation, the new CGA parameters will be utilized to produce the new address. Another open key will be utilized for computing both the Hash1 and Hash2 values. Prior to the lifetime for the being used CGA [12] location has terminated. To guarantee that the CGA location is constantly accessible and substantial, new CGAs ought to be recovered ahead of time before the forerunner will be belittled. The time when the subnet prefix lifetime is over. Another CGA location will then should be recovered. It ought to additionally incorporate the recently made prefix used as a part of computing Hash1. Deciding the correct lifetime for a CGA location depends upon the protection and security level motives [11]

Necessary notations: -

TG: avg. time required for node to produce CGA.

TA: avg. time required for the imitation of address by an assailant.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

T1: time required to compute Hash1.

T2: time required to compute Hash2.

b: no. of bits available within the address, i.e.; trimmed output of Hash1 (IID).

g: sec. level in CGA.

S: no. of bits required to satisfy the Hash2 condition($s = g \times Sec$), i.e; trimmed o/p of Hash2.

The address generator needs on average (2^sx T2) to fulfil the condition of Hash2, adding T1 to build IID from Hash1. So, cost of generating address is:

$$TG = \left(2^{(g \times Sec)} \times T2\right) + T1 \quad \text{(1)}$$

When the assailant starts from Hash2

$$TA : (H2) = (2^S \times T2 + T1) \times 2^b \quad \text{(2)}$$

total time for imitation when we begin with Hash1 (H1) is given by

$$TA\,(H1) = (2^b \times T1 + T2) \times 2^S \quad \text{(3)}$$

Assailant has choice b/w two ways for the cost of attack to be reduced.

Hence, time for imitation address (TA) is:

$$TA = min\{(2^{59} \times T1 + T2) \times 2^{g \times Sec}, (2^{g \times Sec} \times T2 + T1)2^{59}\} \quad \text{(4)}$$

IV. PSEUDO CODE

4.1 Pseudo Code of CGA Generation Algorithm

Procedure generate CGA(Sec, subnet Prefix, publicKey, ExtFields):

```
1.  modifier := random(0x00000000000000000000000000000000,
        0xffffffffffffffffffffffffffffffff) // 16 octets

2.  Label1:
3.  Concat: = concatenate(modifier, 0x000000000000000000, publicKey, extFields)
4.  digest : = SHA1(concat)
5.  Hash2 : = digest[0:14]          // 14 × 8 = 112 leftmost bits
6.  if Sec ≠ 0 and Hash2[0:2*Sec] ≠ 0:    // 2*Sec*8 = 16*Sec leftmost bits
7   modifier := modifier + 1
8   goto label1
9.  end if
10. CollCount := 0x00               // 8-bit collision count
11. label2:
12. concat : = concatenate(modifier, subnetPrefix, collCount,publicKey, extFields)
13. digest := SHA1(concat)
14. Hash1 : = digest[0:8]           // 8*8 = 64 leftmost bits
15. intID := Hash1                  //Hash1 becomes interface identifier
16. intID[0] := intID[0] binary and 0x1c binary or (Sec << 5)  // after writing Sec and u/g bits
17. CGA := concatenate(subnetPrefix, intID)   // concatenate to form the CGA
18. if duplicate(CGA):
```

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

19. CollCount := CollCount + 1
20. if collCount = 3
21. abort
22. **endif**
23. **goto** label2
24. **end if**
25. **return** [CGA, [modifier, subnetPrefix, collCount, publicKey, extFields]]
26. **endProcedure**

4.2 Pseudo Code of CGA Verification Algorithm

CGA Verification[13] takes as input an IPv6 address and CGA parameters. If the verification succeeds, the verifier knows that Public key belongs to that address.

Procedure verifyCGA(CGA, [modifier, subnetPrefix, collCount, publicKey, extFields]):

1. **if** collCount > 2 or CGA[0:8] ≠ subnetPrefix:
2. **return** false
3. **endif**
4. concat := concatenate(modifier, subnetPrefix, collCount, publicKey, extFields)
5. digest := SHA1(concat)
6. Hash1 := digest[0:8] // 8 × 8 = 64 *leftmost bits*
7. Hash1[0] := Hash1[0] binary and 0x1c // *ignore Sec and u/g bits*
8. intID := CGA[8:16] // *interface identifier* (64 *rightmost bits*)
9. intID[0] := intID[0] binary and 0x1c // *ignore Sec and u/g bits*
10. **if** Hash1 ≠ intID:
11. **return** false
12. **endif**
13. Sec := CGA[8] >> 5 // *extract Sec from interface identifier*
14. concat:=concatenate(modifier,0x00000000000000000,publickey extFields)
15. digest := SHA1(concat)
16. Hash2 := digest[0:14] // 14 × 8 = 112 *left most bits*
17. **if** Sec ≠ 0 and Hash2 [0:2*Sec] ≠ 0: // 2 × *Sec* × 8 = 16 × *Sec left most bits*
18. **return** false
19. **endif**
20. **return** true // *verification succeeded*
21. **endProcedure**

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

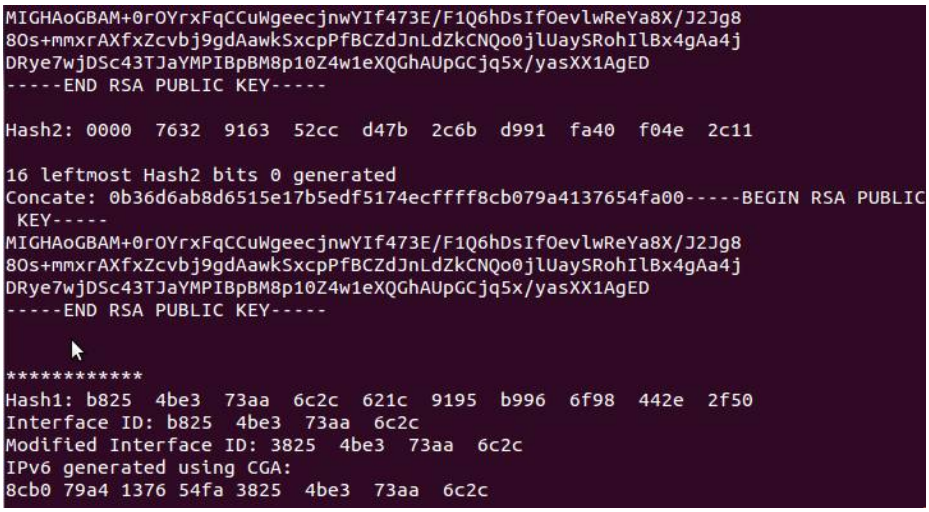
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

V. SIMULATION RESULTS

CGA era time can be stated as the cumulative term of complete CGA task. This integrates time to producing RSA open or private keys, time that is spend in processing Hash2 esteem related to condition 16×Sec-furthest bit left of value of Hash2 will be equivalent as null, also time required to register interface identifier that includes Hash1 figuring. Close to DAD check.Aggregate time for CGA era related to Sec =1 is more notable as compared to normal CGA era time requirement in case of sec=0. Sec esteem "2" could be utilized as a part of the following upcoming years. Although the computation of sec estemm3 is not attainable and cannot be achieved possessing to present speed of CPU.



Result of TB-CGA for the Sec value 0 with RSA Key 1024 bits

VI. CONCLUSION AND FUTURE WORK

We exhibited a handy and programmed path to select parameter of security to calculate CGA era. With this adjusted execution, time is assumed an information after that the parameter of security worth would be resolved as a yield of beast power assault so as to fulfil Hash2 quality. We introduced a useful and programmed path to select parameter of security related to CGA calculation. Level of security is resolved consequently in view of the processing gadget CPU power accessible for hash era. In this work, we have presented a detailed security/efficiency analysis of CGA together with a proposal to solve some security problems and limitations related to self-certifying address generation and verification in CGA. As it can be seen, some work has been done on optimizing CGAs for use in a mobile environment.

In a mobile environment, minimizing the time taken by CGA generation and verification algorithm is important. This is for two reasons. Firstly, handoff have be done in few milliseconds in order to maintain a sufficient quality of service. Secondly, mobile nodes have limited resources that have to be wisely used to remove delays. It is thus significant to view all the work related to factors that affect the time taken by CGA Algorithm. The Factors include Sec value, Hash function.RSA must be replaced by a public key cryptosystem that provides comparable cryptographic strength but has a faster key generation, shorter key length and less expensive signature generation and verification. Only when this is achieved, CGA-based Authentication can be computationally feasible for mobile environment.

REFERENCES

[1] T. Aura, “Cryptographically Generated Address”, RFC3972, Internet Engineering Task Force, March 2005, http://tools.ietf.org/html/rfc3972
[2] Hosnieh Rafiee and Christoph Meinel. “Privacy and Security in IPv6 Networks: Challenges and Possible Solutions”. The 6th International Conference on Security of Information and Networks (SIN2013),ACM,November 26-28,2013 Aksaray,Turkey

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

[3] Narten, T., Draves, R., Krishnan, S.: Privacy Extensions for Stateless Address Auto configuration in IPv6. RFC 4941, Internet Engineering Task Force (September 2007).
[4] Aura, T.: Cryptographically Generated Addresses (CGA). RFC 3972, Internet Engineering Task Force (March 2005), updated by RFCs 4581, 4982.
[5] S. Deering and R. Hinden. Internet Protocol, Version6 (IPv6) Specification. IETF, Dec. 1998. http://tools.ietf.org/html/rfc2460
[6] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Auto configuration (SLAAC). IETF, Sept. 2007. http://tools.ietf.org/html/rfc4862.
[7] F. Gont. A method for Generating Stable Privacy-Enhanced Addresses with IPv6 Stateless Address Auto configuration (SLAAC). Work in Progress, June 2013. http://tools.ietf.org/html/draftietf-6man-stable-privacy-addresses.
[8] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). IETF, Sept. 2007. http://tools.ietf.org/html/rfc4861.
[9] B. Carpenter and S. Jiang. Significance of IPv6 Interface Identifiers. Work in Progress, Aug. 2013. http://tools.ietf.org/html/draft-ietf-6man-ug-02.
[10] H. Rafiee and C. Meinel. A Secure, Flexible Framework for DNS Authentication in IPv6 Auto configuration. IEEE, Proceedings of the 12th IEEE International Symposium on Network Computing and Applications (IEEE NCA13), 2013.
[11] A practical guide to ipv6 network.
[12] Wikipedia ipv6
[13] T. Aura, “Cryptographically Generated Addresses (CGA),” in Information Security, vol. 2851, C. Boyd and W. Mao, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 29–43.
[14] T. Aura, “Cryptographically Generated Addresses (CGA),” RFC 3972, IETF, Mar-2005. [Online]. Available: http://tools.ietf.org/html/rfc3972.
[15] G. O’Shea and M. Roe, “Child-proof authentication for MIPv6 (CAM),” SIGCOMM Comput. Commun. Rev., vol. 31, no. 2, pp. 4–8, Apr. 2001.
[16] P. Nikander, “Denial of Service, Address Ownership, and Early Authentication in the IPv6 World,” in Security Protocols, vol. 2467, B. Christianson, J. Malcolm, B. Crispo, and M. Roe, Eds. Springer Berlin / Heidelberg, 2002, pp. 22–26.
[17] G. Montenegro and C. Castelluccia, “Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses,” in In Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS), 2002.
[18] Barrera, D.; Van Oorschot, P., "Security visualization tools and IPv6 addresses," Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on , vol., no., pp.21,26, 11-11Oct.2009 doi: 10.1109/VIZSEC.2009.5375538.
[19] Jayanthi, J.G.; Rabara, S.A., "IPv6 Addressing Architecture in IPv4 Network," Communication Software and Networks, 2010. ICCSN '10. Second International Conference on, vol., no., pp.461,465,26-28Feb.2010doi: 10.1109/ICCSN.2010.116.