Simulation Study of a Many-to-One Mapping for IPv6 Address Owner Identification in an Enterprise Local Area Network

Nashrul Hakiem[1], Mohammad Umar Siddiqi[2], and Hashum Mohamed Rafiq[3]
(Corresponding author: Nashrul Hakiem)

Department of Informatics Engineering, Faculty of Science and Technology[1]
UIN Syarif Hidayatullah Jakarta, Jl. Ir. H. Juanda 95 Jakarta 15412 Indonesia
Department of Electrical and Computer Engineering, Faculty of Engineering[1,2,3]
International Islamic University Malaysia, Jalan Gombak, Kuala Lumpur 53100, Malaysia
(Email: hakiem@yahoo.com)

(Received Nov. 19, 2012; revised and accepted Aug. 15, 2013)

Abstract

Owner identification is an important aspect of improving network visibility and enhancing network security within local area networks deploying IPv6. This paper presents a simulation study for owner identification in an enterprise local area network from their IPv6 addresses. The study is based around the reverse implementation (many-to-one mapping) of a one-to-many reversible mapping. The paper reviews the many-to-one mechanism and the associated simulation software development, followed by presentation of results obtained from required functional tests. The IPv6 address data can be obtained from the output of any network monitoring software. In addition to a text format for verification, it also uses a checksum for validation which is used during the IPv6 address generation and identification. The simulation software given here can easily identify an IPv6 address owner if the IPv6 address is properly generated by the mechanism and it can display particular verification messages.

Keywords: Checksum, IPv6 address, many-to-one mapping, network monitoring, network visibility, one-to-many mapping, owner identification.

1 Introduction

Identity is one of the most important aspects within the Internet [18]. It facilitates controlling user activity and access to the network in order to improve network visibility and thus improve network security. It plays a central role in the development of the Future Internet [15].

IPv6 represents a considerable improvement compared to the previous version, IPv4. However, some potential security problems still remain and require improvement [3]. Some research work has been carried out to improve IPv6 security [8, 12] and to overcome IP address deficiency and the lack of IP address security [14].

One-to-many reversible mapping [5] is a mechanism to enhance IPv6 address generation in terms of security and privacy. This one-to-many mapping between user space and IPv6 addresses is generated cryptographically using the Cipher Feedback (CFB) mode of operation of the Advanced Encryption Standard (AES).

This paper presents IPv6 address owner identification based on many-to-one mapping, the reverse implementation of a one-to-many reversible mapping. The paper is organized as follows. An overview of related works is given in Section 2. The many-to-one mapping mechanism is detailed in Section 3. Section 4 presents the IPv6 address identification mechanism and presents the results obtained from various functional tests. Conclusions are given in Section 5.

2 Related Works

2.1 Network Monitoring

Network administrators need to be aware of and have a handle on different types of traffic that is traversing their networks. Traffic monitoring and analysis is essential in the sense that it provides a more effective way to troubleshoot and resolve issues when they occur. This helps preventing network services from a state of "stand-still" over extended periods of time [2].

There are several popular network management software packages specifically designed with emphasis on network monitoring, measurement, and analysis which are available from commercial sources and open source vendors [13]. These tools help in monitoring the enterprise network activities in real time and analyzing the network for LAN usage. Thus, these tools not only help to correct network problems on time, but also to prevent network failure, to detect inside and outside threats, and

make good decisions for network planning [26].

2.2 IP Address Identification

An important aspect of network monitoring is to be able to identify who is using the resources within the network. A network administrator may take necessary action against a user who misbehaves or misuses the resources within an enterprise local area network.

Cryptographically Generated Addresses (CGAs) have been designed to solve the so-called IPv6 Address Ownership problem [1]. A CGA is used in SEcure Neighbor Discovery (SEND) [19] to safeguard the address of the sender. SEND has been proposed to improve the security of the Network Discovery (ND) protocol in environments where the physical security of the link is not guaranteed (for example in a wireless environment). However, the use of CGA is expensive and time consuming. There is a mechanism to reduce the generation time by moving most of the computation to the server [25]. Further enhancement has been undertaken to support Multi-key CGA (MCGA) [11] and the multiple hash algorithm in CGA [22].

A proposal has been made to generate the IPv6 address in the stateful mode which introduces a light-weight extension of anonymous communications in IPv6 networks [9]. It generates a changeable address using DHCPv6 which may be imported into onion routing-based anonymous communication systems. The objective of this method is to enhance the overall anonymity of the host [9].

A study on the advantages of interaction of DHCPv6 and CGA has been undertaken in [4, 24], followed by a proposal in which CGA is used to efficiently improve the security of DHCPv6 interaction. CGA may be used to authenticate the DHCPv6 server.

2.3 One-to-Many Reversible Mapping

A one-to-many reversible mapping provides a mechanism to enhance IPv6 address generation in terms of security and privacy. A different IPv6 address is given each time a node tries to access the local area network (LAN). This makes it more difficult for eavesdroppers to identify the owner of an IPv6 address. Thus, it protects user privacy as recommended by IETF [21].

The one-to-many mapping between user space and IPv6 addresses is generated cryptographically using the Cipher Feedback (CFB) mode of operation of the Advanced Encryption Standard (AES) [5]. The required software development for IPv6 address generation (one-to-many mapping) has been presented [7].

The mechanism used to generate the user IPv6 address [5] is able to link the dynamic IPv6 address to a particular user, if needed, to improve network visibility, and hence improve security within an enterprise local area network.

3 Many-to-one Mapping

The one-to-many reversible mapping [5], in the reverse mode (many-to-one), is capable to identify users from their IPv6 addresses to facilitate tracking of network anomalies or violations of policies and to improve network visibility. By the random generation of an IPv6 address, the privacy of the user is protected even though the communication is transparent end-to-end.

Figure 1 shows the proposed Interface ID format comprising of a 6-bit checksum, 2-bit 'u' and 'g', a 48-bit `encryptedUserID`, and an 8-bit `keyIdx`.

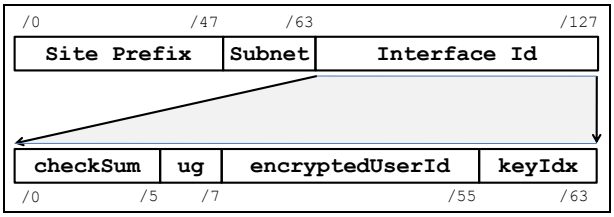

Figure 1: Proposed interface ID format

Figure 2 shows an activity diagram for Interface ID generation which has an 18-bit user ID as input and produces a dynamic 64-bit interface ID.

The 48-bit `encryptedUserId` is generated as per the activity diagram shown in Figure 3 which can be represented as:

$$f(p) \mapsto C_j, j = 1 \ldots n \tag{1}$$

where an 18-bit user ID p is randomly mapped to one of all the n permissible 48-bit encrypted user ID C_j with $n = 2^{48} / 2^{18} = 2^{30}$.

The detailed construction of the user ID encryption can be represented as follows:

$$conc(R, p) = P \tag{2}$$

where the 48-bit concatenated user ID P is a concatenation of a 30-bit R (random number) and an 18-bit p (user ID).

From Equation (2), it can be seen that the same p can generate many P (one-to-many mapping) because of additional randomly generated bits of R. However, the user ID is clearly visible which clashes with one of the objectives to protect user privacy. Encryption is, therefore, performed using CFB-AES which has a higher avalanche effect. Therefore, any change of even a single bit in P will significantly affect many bits of C to produce a pseudo-random value that actually corresponds to the same user ID p.

$$C = E_{CFB-AES}(K, IV, P) \tag{3}$$

where $E_{CFB\text{-}AES}$ denotes the encryption of P under the key K, IV Initialization Vector, and C is the encrypted user ID which is embedded in the Interface ID.

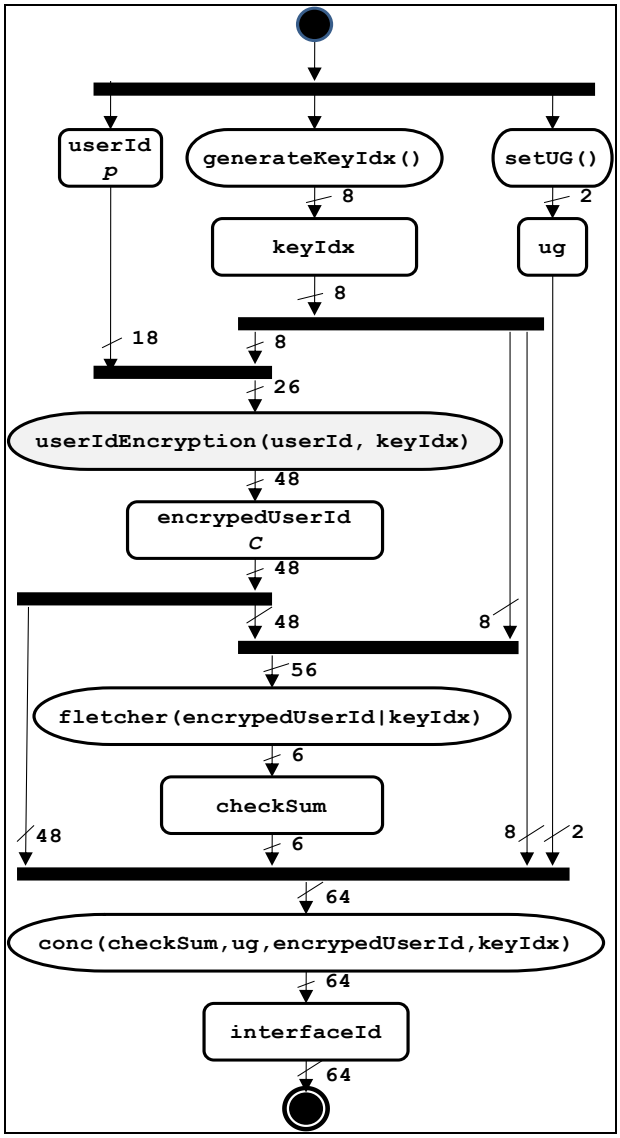

Figure 2: Interface ID generation

Details of the CFB-AES encryption operation are given in Equations (4) and (5) as follows:

$$C_k = P_k \oplus S_s\left[E\left(K, C_{k-1}\right)\right], k = 2 \ldots n \tag{4}$$

where k the sequence of blocks from the second to the last, and the first block encryption also depends on the IV (Initialization Vector) as follows:

$$C_1 = P_1 \oplus S_s\left[E\left(K, IV\right)\right]. \tag{5}$$

4 Results and Discussion

The generated dynamic address can be uniquely linked to a particular user if the need arises. There is a many-to-one mapping between the IPv6 addresses and user space. Figure 4 and Figure 5 show Interface ID owner identification and user ID decryption respectively.

4.1 User ID Identification

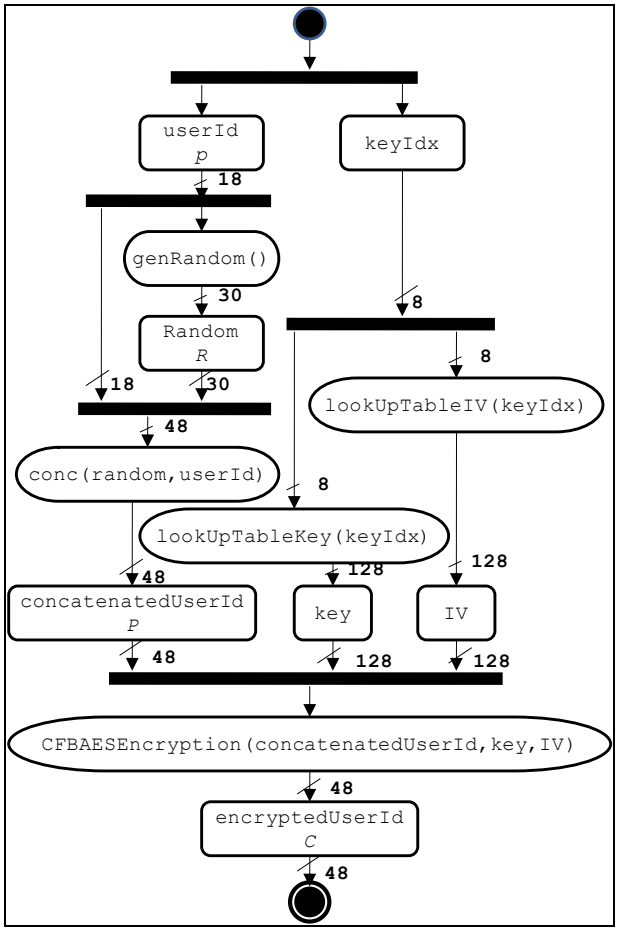

Figure 3: User ID encryption

User ID identification (many-to-one mapping) can be represented as:

$$f\left(C_j\right) \mapsto p, j = 1 \ldots n. \tag{6}$$

To obtain p to identify an 18-bit user ID from a member of C which is part of the Interface ID, the method has to perform validation first as depicted in Figure 4. The `userIdDecryption` process is illustrated in Figure 5 and can be represented as:

$$P = D_{CFB-AES}\left(K, IV, C\right) \tag{7}$$

where $D_{CFB\text{-}AES}$ denotes the decryption of C under the key K and Initialization Vector IV to produce a 48-bit user ID.

Subsequently, simply eliminate the first 30 bits (R) from 48-bit concatenated user ID P.

$$rem\left(P, R\right) = p. \tag{8}$$

This produces a user ID (p) from some P (many-to-one mapping).

For the identification process, the mechanism should yield P from C (Equation (7)). In CFB-AES, this requires encrypting both the first block and the rest of the blocks which can be seen in Equations (9) and (10).

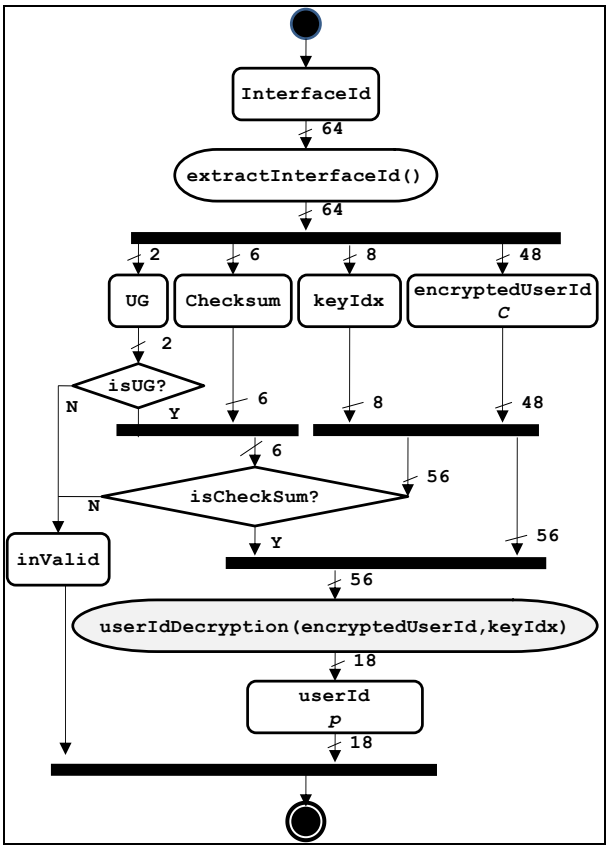

Figure 4: User ID identification

$$P_1 = C_1 \oplus S_s\left[E(K, IV)\right] \tag{9}$$

$$P_k = C_k \oplus S_s\left[E(K, C_{k-1})\right] \tag{10}$$

where k is the second block to the end of the blocks and s is the segment of unit of bits.

4.2 Checksum

A 6-bit checksum is inserted in the proposed Interface ID part of IPv6 address as illustrated in Figure 1 [6] in order to validate the generated Interface IDs.

A modified Fletcher checksum has been used because it is more effective in most situations and has a lower computational cost compared to the Adler checksum [6, 16].

$$Y = \sum_{l=1}^{14}\left(Y_{l-1} + \lambda \times X_l\right) \tag{11}$$

$$Z = \sum_{l=1}^{14}\left(Z_{l-1} + Y_l\right) \tag{12}$$

$$W = conc\left((Y \bmod 8),(Z \bmod 8)\right) \tag{13}$$

where X, Y, and Z are hexadecimal values and W is two octal digits with Y and Z both initialized to 0 (zero). Symbol λ is a parametric constant that can be arbitrarily chosen by the administrator, while l is the number of hexadecimal values.

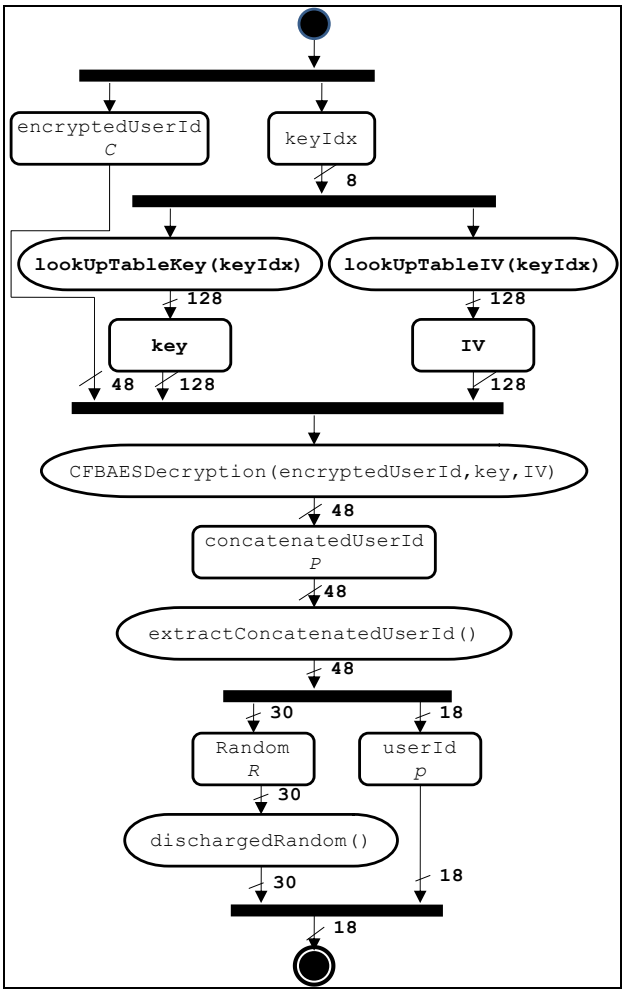

Figure 5: User ID decryption

Figure 6 shows the pseudocode of `function generateChecksum()` which returns a string data type representing the checksum value. It has two parameters which are a string and an integer data type. The string input is a combination of a 48-bit encrypted user ID and an 8-bit key index. This checksum is used for both the address generation and the IPv6 address identification.

```
  function generateChecksum( uid:String,
radix:int ) → String
  {
 c, s, y, z : String
 cInt, yInt, zInt : int = 0
 sumY, sumZ, i  : int = 0
 while ( i < uid.length() )
 {
   c = uid.substring( i, i + 1 )
   cInt = parseInt( c, radix )
   yInt = Constant * cInt
   sumY += yInt
   sumZ += sumY
   i++
 }
 y = toOctalString( sumY )
 z = toOctalString( sumZ )
 s = y + z
 ← s
  }
```

Figure 6: Function generateChecksum() pseudocode

4.3 Software Implementation

The user ID Identification, which is depicted in Figure 4, is implemented as `function userIdIdentification()`, with the pseudo-code shown in Figure 7.

Firstly, the IPv6 address format is verified, and then it takes the leftmost 48 bits to be compared with the current site prefix. After that, it checks the u and g bits as 0 respectively and finally it compares the embedded checksum in the Interface ID with the checksum computation [7].

After the verification process, user ID decryption is performed which is drawn from the activity diagram of Figure 4. User ID decryption is implemented into `function userIdDecryption()` as depicted in Figure 8.

The 128-bit key, 128-bit `initialization Vector`, and 48-bit `encryptedUserId` are used as input and an 18-bit `userId` is produced.

```
function userIdIdentification
(ipv6Address:String)
{
  userId : String
  sitePrefix : String
  interfaceId : String
  checksum : String
  key, encryptedUserId, iv : String
  if (isIpv6Address(ipv6Address) )
  {
    splitIPv6Address(ipv6Address)
    if (isSitePrefix(sitePrefix))
    {
      splitIID(interfaceId)
      if (isUG(interfaceId))
      {
        if (isChecksum(interfaceId))
        {
          userId = userIdDecryption(key,
encryptedUserId, iv)
        }
        else
        {
          message = "Incorrect checksum."
        }
      }
      else
      {
        message = "Incorrect u and g bit
values."
      }
    }
    else
    {
      message = "Incorrect site prefix within
enterprise."
    }
  }
  else
  {
    message = "Incorrect IPv6 address format."
  }
  userId = message
}
```

Figure 7: Pseudo-code for the function userIdIdentification()

```
function userIdDecryption(key: String,
encryptedUserId: String, iv: String)
{
  userId, userId18Bit: String
  cfbAes =  new cfbAes (key,
encryptedUserId, iv)
  cfbAes.decrypts
  userId = cfbAes.getOutStr()
  ← userId18Bit = removeR(userId)
}
```

Figure 8: Pseudo-code for the function userIdDecryption ()

Figure 9 is an example of the output from Wishark network monitoring and analysis [10, 23]. If the analysis shows any anomaly or suspicious activity, the offending IP address is indicated. This IPv6 address then becomes the input to the user ID identification procedure in order to identify the IPv6 address owner within the enterprise local area network.

Figure 10 shows a graphical user interface frame with a text field for the IPv6 address input. The IPv6 address input is from any network monitoring output which has produced an IPv6 address. The 'Identify' button within this frame calls the `function userIdIdentification ()` as depicted in Figure 7.

The IPv6 address owner or an error message is displayed in the user ID text field. Particular error messages are: incorrect IPv6 address format; incorrect site prefix; incorrect u and g bit values; and incorrect checksum.

4.4 Checksum Validation

A checksum is used for validation in the IPv6 address generation and IPv6 address owner identification as per Figure 2 and Figure 4 respectively. For example an IPv6 address is generated for an 18-bit `userId` $(321675)_8$ with an8-bit `keyIdx` $(fd)_{16}$ and a 30-bit random number $(77343\ 67271)_8$.

Based on Table 1, using this particular `keyIdx`, the 128-bit key and the 128-bit IV; the key and the 128-bit IV are $(972635b8\ 56825391\ 997548f7\ 14379866)_{16}$ and $(93348773\ 2882790e\ 58194495\ 8426894a)_{16}$ respectively. This produces 48-bit `encryptedUserId` $(fafa54\ 2ddf06)_{16}$, then it constructs a 6-bit checksum $(07)_8$ with a `keyIdx` and an `encryptedUserId` as parameters.

This results in `1cfa:fa54:2ddf:06fd` as the 64-bit Interface ID. This Interface ID is concatenated with the site prefix and the subnet ID provided by the enterprise local area network to produce the 128-bit IPv6 address [20].

For Interface ID owner identification, primarily it checks the correctness of the IPv6 address format. Then the IPv6 address is split into the site prefix, subnet ID, and Interface ID. If the site prefix matches with the current enterprise site prefix, then it checks the 7th and 8th bit as the 'u' and 'g' bits of the Interface ID [20].

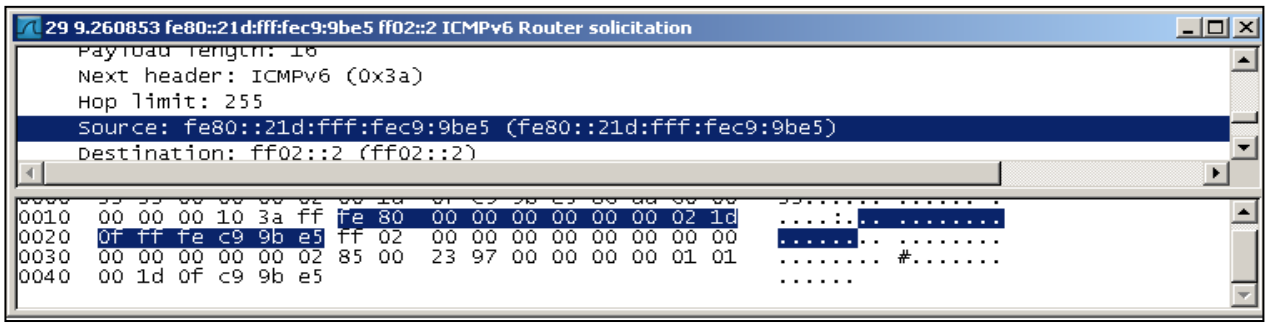

Figure 9: Output example of network monitoring and analysis

Table 1: Key and IV examples

Idx	Key	Initialization Vector
1	719382b603572138744295f461126613	680234479629537e328691705173641a
2	720383b604573139745296f462127614	681235480630538e329692706174642a
3	721384b605574140746297f463128615	682236481631539e330693707175643a
4	722385b606575141747298f464129616	683237482632540e331694708176644a
5	723386b607576142748299f465130617	684238483633541e332695709177645a
...	...	...
128	846509b730699265871422f588253740	807361606756664e455818832300768a
...	...	...
252	970633b854823389995546f712377864	931485730880788e579942956424892a
253	971634b855824390996547f713378865	932486731881789e580943957425893a
254	972635b856825391997548f714379866	933487732882790e581944958426894a
255	973636b857826392998549f715380867	934488733883791e582945959427895a
256	974637b858827393999550f716381868	935489734884792e583946960428896a

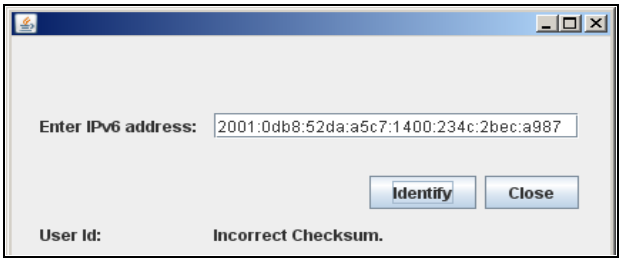

Figure 10: User interface

Table 2: DHCPv6 address generation mechanism

Mechanism	Advantages	Disadvantages
EUI-64 [20]	Unique identifier	Threatens the privacy of users
Random [4, 21]	Easy implementation	Difficult to identify IPv6 address owner
One-to-many reversible mapping [5]	• Unique identifier • Easy implementation • Respect user privacy • Security improvement	Increase processing time

Furthermore, a checksum is computed and compared with the embedded checksum in the Interface ID. If the two are equal, then the identification may proceed to the next stage to obtain a user ID as displayed in Figure 5.

Table 2 shows the relative advantages and disadvantages of standard DHCPv6 address generation mechanisms.

The CFB-AES mechanism is able to generate pseudo-randomly IPv6 address which makes it difficult to identify the owner [5], hence respects the user privacy. However it is possible for administrator to identify IPv6 address owner in the IP address layer in order to improve network security. Although the mechanism reduces processing speed, however it is still practical since it takes less than 100 milliseconds for generating address or identifying the IPv6 address owner [6].

5 Conclusion

This paper presents a method, based on the reverse implementation of an one-to-many reversible mapping, for identification of an IPv6 address owner in an enterprise local area network. The reverse implementation (many-to-one mechanism) has been reviewed and the development of the underlying software development has been given, followed by results of several functional tests. The IPv6 address data may be captured for evaluation from the output of any network monitoring and analysis system and the IPv6 address owner identification scheme may be implemented as a complement of the network monitoring software in order to improve network security. It may be noted that the performance impact of an enterprise wireless local area network, in general improves with improved network security [17].

References

[1] C. Castelluccia, "Cryptographically generated addresses for constrained devices," *Wireless Personal Communications*, vol. 29, pp. 221–232, 2004.

[2] A. Cecil, *A Summary of Network Traffic Monitoring and Analysis Techniques*, Student Paper, ed: Washington University in St. Louis, 2006.

[3] E. Durdağı and A. Buldu, "IPv4/IPv6 security and threat comparisons," *Procedia - Social and Behavioral Sciences*, vol. 2, pp. 5285-5291, 2010.

[4] S. Groat, M. Dunlop, R. Marchany, and J. Tront, "What DHCPv6 says about you," in *2011 World Congress on Internet Security (WorldCIS)*, pp. 146-151, 2011.

[5] N. Hakiem, A. U. Priantoro, M. U. Siddiqi, and T. H. Hasan, "Generation of IPv6 addresses based on one-to-many reversible mapping using AES," in *Recent Progress in Data Engineering and Internet Technology*, vol. 157, pp. 183-189, 2012.

[6] N. Hakiem, M. U. Siddiqi, and S. P. W. Jarot, "Collision probability of one-to-many reversible mapping for IPv6 address generation," in *2012 International Conference on Computer and Communication Engineering (ICCCE)*, pp. 599-602, Kuala Lumpur Malaysia, 2012.

[7] N. Hakiem and M. U. Siddiqi, "One-to-many reversible mapping for IPv6 address generation: simulation software development," *Journal Of Theoretical And Applied Information Technology*, vol. 47, pp. 892-901, Jan 2013.

[8] N. T. Hoa, K. Naoe, and Y. Takefuji, "Simplifed IPSec protocol stack for micro server " *International Journal of Network Security*, vol. 11, pp. 46-54, July 2010.

[9] Z. Jia, D. Haixin, L. Wu, and W. Jianping, "A light-weighted extension of anonymous communications in IPv6 network," in *2010 International Conference on Green Circuits and Systems (ICGCS)*, pp. 404-408, 2010.

[10] S. Kakuru, "Behavior based network traffic analysis tool," in *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, pp. 649-652, 2011.

[11] J. Kempf, J. Wood, Z. Ramzan, and C. Gentry, "IP address authorization for secure address proxying using Multi-key CGAs and ring signatures," in *Advances in Information and Computer Security*. vol. 4266, pp. 196-211, 2006.

[12] J. Li, P. Zhang, and S. Sampalli, "Improved security mechanism for mobile IPv6," *International Journal of Network Security*, vol. 6, pp. 291–300, May 2008.

[13] B. Li, J. Springer, G. Bebis, and M. H. Gunes, "A survey of network flow applications," *Journal of Network and Computer Applications*, vol. 36, pp. 567-581, 2013.

[14] X. Liu, G. Hu, W. Chen, and K. Xu, "IPv6 protocol simplification for the internet of things," *Qinghua Daxue Xuebao/Journal of Tsinghua University*, vol. 52, pp. 699-703, 2012.

[15] P. Martinez-Julia and A. F. Skarmeta, "A lightweight and Identity-Based network architecture for the internet of things," in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pp. 711-716, 2012.

[16] T. Maxino, *Revisiting Fletcher and Adler Checksums*, DSN 2006 Student Forum, ed. Pittsburgh: Institute for Software Research, Carnegie Mellon University, 2006.

[17] D. Nayak, D. B. Phatak, and A. Saxena, "Evaluation of security architecture for wireless local area networks by indexed based policy method: a novel approach," *International Journal of Network Security*, vol. 7, pp. 1–14, July 2008.

[18] J. Oltsik, *Identity-Aware Networking*, White Paper, ed: Enterprise Strategy Group, 2010.

[19] RFC3971, *SEcure Neighbor Discovery (SEND)*, Standards Track, IETF Network Working Group, 2005.

[20] RFC4291, *IP Version 6 Addressing Architecture*, Standards Track, IETF Network Working Group, Feb. 2006.

[21] RFC4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, Standards Track, IETF Network Working Group, Sep. 2007.

[22] RFC4982, *Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)*, Standards Track, IETF Network Working Group, 2007.

[23] C. Sanders, *Practical Packet Analysis Using Wireshark to Solve Real-world Network Problems*, 2nd ed. Canada: No Starch Press, 2011.

[24] S. Sean, L. Xiaodong, S. Zhili, and J. Sheng, "Enhance IPv6 dynamic host configuration with cryptographically generated addresses," *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, pp. 487-490, 2011.

[25] G. Su, *et al.*, "A quick CGA generation method," *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, pp. V1-769-V1-773, 2010.

[26] C. So-In, *A Survey of Network Traffic Monitoring and Analysis Tools*, Student Paper, ed: Washington University in St. Louis, 2006.

Nashrul Hakiem holds a master degree in Informatics Engineering (Software Engineering) from Institut Teknologi Bandung (ITB), Indonesia as of 2001. Prior to that, he obtained a Bachelor in Computer Science from Universitas Gadjah Mada (UGM), Indonesia in 1996. Currently, he is a PhD candidate in Electrical and Computer Engineering Department of International Islamic University Malaysia (IIUM). The author is a lecturer in the Informatics Engineering Department, Faculty of Science and Technology, Universitas Islam Negeri (UIN) Syarif Hidayatullah Jakarta, Indonesia. His research interests are in software engineering, object oriented technology, cryptography, and IPv6.

Mohammad Umar Siddiqi received his B.Sc. and M.Sc. degrees from Aligarh Muslim University (AMU Aligarh) in

1966 and 1971, respectively, and a Ph.D. degree from the Indian Institute of Technology Kanpur (IIT Kanpur) in 1976, all in Electrical Engineering. He has been in the teaching profession throughout, first at AMU Aligarh, then at IIT Kanpur and Multimedia University Malaysia. Currently, he is a Professor in the Faculty of Engineering at International Islamic University Malaysia. His research interests are in coding, cryptography, and information security.

Hashum Mohamed Rafiq received his B.Sc. degree in Computer Science from University of Dar Es Salaam (UDSM), Tanzania and holds an M.Sc. degree in Computer and Information Engineering from the International Islamic University Malaysia. Currently, he is a PhD candidate in Electrical and Computer Engineering Department of the International Islamic University Malaysia (IIUM). His research interests are in coding, cryptography, and information security.