VOL. 11, NO. 15, AUGUST 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DISCOVERY OF RIGHT BINARY PATTERN IN KEY MANAGEMENT USING SECURITY MATRIX AND MAGIC CARDS

K. R. Sekar, P. Saravanan and J. Sethuraman
School of Computing, SASTRA University, Thanjavur, Tamil Nadu, India
E-Mail: sekar_kr@cse.sastra.edu

ABSTRACT

In the era of networking, authentication is a critical issue. In any intra or inter subnet communication key management and authentication play a vital role. In the past two decades different methodologies were employed in the key management and its authentication. In this research work, a new ideological factor with collaborative techniques has been proposed. A series of procedures like substitution, transpose and compression have been used to encrypt the plain text and the selected key text. At sender side a critical region has been generated using traditional encryption. The work also studies varying issues with the traditional encryption mechanism and proposes the solution for the problem. At the receiver end the critical region has been verified through the proposed magic square.

Keywords: encryption, substitution, transposes, compression, magic cards.

1. INTRODUCTION

In the unsecured network authorization and authentication is playing a major role in sending and receiving the messages. For the past two decades the above said administrative processes have been accelerated in the market and strenuous research is still going into this arena. Encoding, compression, decompression and decryption are the normal phenomenon in the existing field. A certificate-less PGP-like trust establishment scheme for MANET by using self-certifying ID-based cryptography has been researched [1]. The scheme determines public-keys from the node's identities and their trust levels Another counter measure to effectively manage a number of keys for image encryption, privacy protection and user authentication, which selectively implement a Kerberos approach, a round key approach and a double hash chain approach were proposed [2]. An RFID mutual-authentication protocol with synchronous updated-keys using hash function was recommended for key management [3].

In this research work after decompression, the generated bit pattern should match with the receiver side generated pattern. While the receiver decompressing the compressed bits there may be chances of getting more number of patterns. Many schemes established to reduce the number of patterns in the research domain. A secure and efficient AKA protocol, called SE-AKA, which can fit in with the entire group authentication scenarios in the LTE networks to provide strong security and help manage group devices [4].An automated key management system to automatically generate, distribute and update keys for a collection of keying groups. The proposed protocol for key management ensures security in the form of authentication, integrity, confidentiality, protection against replay attacks, and robustness across reboots [5]. Performance enhancements to the LTE network to solve its security related drawbacks were another approach in secured key management [6]. An ID-based remote mutual authentication has been proposed with key agreement scheme on ECC which supports a session key agreement between the user and the server [7].

Even though, these schemes were reducing the number of patterns, the receiver finds a hectic problem to identify which pattern is the correct one as sent by the sender. To avoid this situation, the magic square information has been encapsulated with the packet sent over the channel. The magic square is an integer matrix generated based on the size of the key and consist the critical region through which the receiver can identify the right pattern.

Any message can be divided into equal number of characters and referred as a plain text pattern. A key text was used for the key management process to authenticate the whole message. This key management has sub-processes such as substitution, transposing, encoding, and compressing the data on the sender side. The resultant data is added with magic square information. At the receiver, the right pattern is identified using the magic squares. Then the original message will be regenerated by the reverse processes of the above said sub-processes.

In section 2, related works are presented. The proposed methodology is described in section 3. The result of the proposed work is discussed in section 4. Section 5 concludes the research work.

2. RELATED WORK

A polynomial based self-healing key management scheme with broadcast authentication and enhanced collusion resistance has been exhibited [11]. To achieve a vehicle user's privacy preservation while improving the key update efficiency of location based services (LBSs) in vehicular ad hoc networks (VANETs), it is advised a dynamic privacy-preserving key management scheme called DIKE [12]. Further, the vulnerability of de-synchronization attacks in handover key management has been identified [13]. Providing secure and correct association of a group of sensors with a patient and satisfy the requirements of data confidentiality and integrity in BANs is another challenging work and it is enhanced by a novel enhanced secure sensor association and key management protocol based on elliptic curve cryptography and hash chains [14]. Another secure user

authentication and key distribution scheme for mobile open IPTV has been discussed in [15]. The mechanism transmits the data among users with efficient authentication, secured transmission and least delay time. To link the incompatible key management architecture for supporting various cryptosystems in smart meters the remote key generation and distribution system has been worked out. This reduces the total computation time in smart meters [16]. Maximum number of cryptographic keys was produced using gray level co-occurrence matrix and textured image for a flexible method of key management [17]. Authenticated asymmetric group key agreement is the certificate based system for the key agreement process. The public keys are authenticated and the inherent escrow problem has been resolved [18]. Authenticated key distribution is also the challenging area in key management. To distribute the key secretly, a set of prime numbers has been used to construct and reconstruct the secret using Chinese Remainder Theorem has been proposed [19]. Secured clustering and routing with dynamic key management for heterogeneous mobile wireless sensor networks has been researched using particle swarm optimization logics [20].

3. PROPOSED METHODOLOGY

In client-server, mobile environment key authentication and management are the most challenging jobs for the researcher. New user authentication and key agreement protocol using bilinear pairings for mobile client–server environment has been proposed [8]. To overcome the security weaknesses of Turkanovicetal's scheme for heterogeneous wireless sensor, an improved UAKAS has been tailored [9]. A Novel architecture of the Wireless Sensor Networks (WSN) environment has been presented for user authentication and key agreement scheme [10].

3.1 Security matrix

The security matrix is formed by the given data, text and the selected key text. The following reference table has been used for forming the security matrix. For both upper and lower case of English character the corresponding row and column values have been considered as elements in the matrix.

Table-1. Security Matrix reference table.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	A/a	C/c	C/c	D/d	E/e	F/f	G/g	H/h	I/i
2	J/j	K/k	L/l	M/m	N/n	O/o	P/p	Q/q	R/r
3	S/s	T	U/u	V/v	W/w	X/x	Y/y	Z/z	

Using the table any data, text or key text can be converted to an equivalent security matrix using the substitution method.

Any message or key which is to be transferred will be considered as data in this context and it is divided into fixed sized characters. In this research work, the data are divided into five character word and first converted into an equivalent matrix using the reference Table.

For example, consider the data 'saran' is transferred using the key called 'zebra'. The equivalent matrixes for these words are as follows:

$$Dm(saran)=\begin{bmatrix}3 & 1\\ 1 & 1\\ 2 & 9\\ 1 & 1\\ 2 & 5\end{bmatrix} \qquad Km(zebra)=\begin{bmatrix}3 & 8\\ 1 & 5\\ 1 & 2\\ 2 & 9\\ 1 & 1\end{bmatrix}$$

By combining the Dm and Km the following matrix has been generated

$$Sm(saran)=\begin{bmatrix}3 & 1 & 3 & 8\\ 1 & 1 & 1 & 5\\ 2 & 9 & 1 & 2\\ 1 & 1 & 2 & 9\\ 2 & 5 & 1 & 1\end{bmatrix}$$

$$\text{implies } S^{T}m(saran)=\begin{bmatrix}3 & 1 & 2 & 1 & 2\\ 1 & 1 & 9 & 1 & 5\\ 3 & 1 & 1 & 2 & 1\\ 8 & 5 & 2 & 9 & 1\end{bmatrix}.$$

Considering a row at a time the decimal pattern formed from the matrix has been given in the equation (1).

$$3121211915311218529 1 \qquad (1)$$

This decimal pattern is converted into binary pattern using Huffman tree as follows

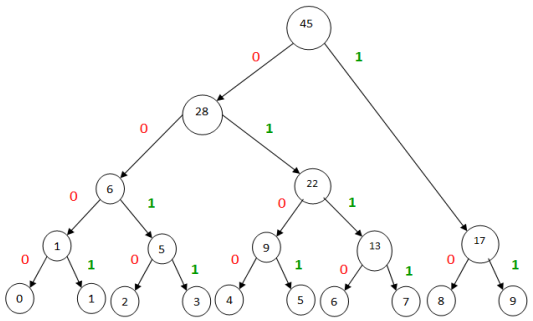

Figure-1. Traditional Huffman tree.

Table-2. Huffman conversion table.

Decimal Value	0	1	2	3	4	5	6	7	8	9
Binary value	0000	0001	0010	0011	0100	0101	0110	0111	10	11

From the table the pattern '31212119153112185291' is converted into the binary pattern as follows:

00110001001000010010000100011100010101001100010
00100100001100101001011000 1 (2)

This is encrypted text in the binary sent over the channel for transmission. At the receiver side the cipher text is verified for the error detection in two levels. The first level uses the number 1's in the binary pattern and second level uses the sum of decimal digits in decimal pattern. In both level the numbers will be calculated and attached with the packet with the following packet format.

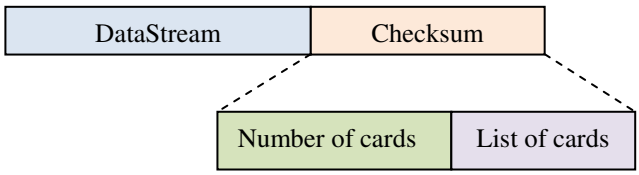

Figure 2. Packet format for the proposed system.

Case 0: Using the Huffman Tree

From equation (2) the number of 1's in a binary pattern of data 'saran' is 26.

Total number of bits in that pattern is 74.

From the pattern 31212119153112185291, it is understood that most of the digits are one or almost least digits. In the Huffman conversion table, the lowest digits are encoded as four binary digits, whereas 8 and 9 are encoded with only two digits. But most of the decimal pattern for the data consists of less number of digits. This scenario increases the total number of binary bits for a single word high and high. If the number of bits in any transmission is high, it reduces the overall performance of data transmission. Hence the proposed approach modifies version of Huffman tree and conversion table as in the next case.

Case 1: Using the modified Huffman Tree

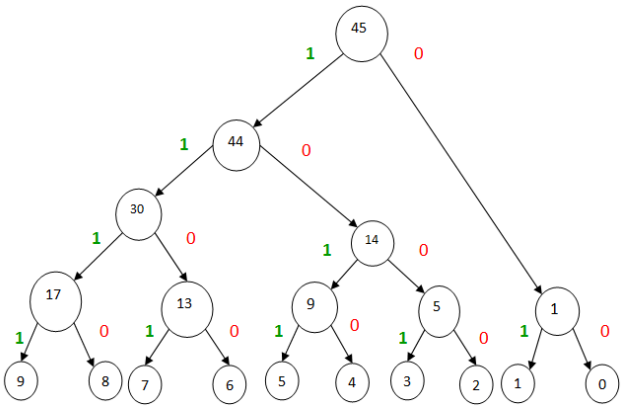

Figure-3. Modified Huffman Tree.

Table-2. Modified Huffman table.

Decimal value	0	1	2	3	4	5	6	7	8	9
Equivalent binary value	00	01	1000	1001	1010	1011	1100	1101	1110	1111

From the table the pattern '31212119153112185291' is converted into the binary pattern as follows:

10010110000110000101111101101110010101100001111
010111000111101 (3)

Number 1's in binary pattern of data 'saran' is 34. Total number of bits in that pattern is 62. From the proposed method the total number of bits in the binary pattern just 62 and reduced from 72 for the data 'saran'.

Similarly, the proposed method tested the actual Huffman tree methods and the modified Huffman tree method for various data and proved that the proposed modified Huffman tree method performed well with high positive scale.

Case 3: Security Matrix generated without additional process

Case 1 and case 2 uses the data and the key matrix to form the security matrix. If the methodology is applied continuously a vulnerable point occurs and hackers can track the key easily.

Consider another data 'sekar' and it is transferred using the key called 'zebra'. The equivalent matrixes for these words are as follows:

$$Dm(sekar)=\begin{bmatrix}3&1\\1&5\\2&2\\1&1\\2&9\end{bmatrix} \text{ and } Km(zebra)=\begin{bmatrix}3&8\\1&5\\1&2\\2&9\\1&1\end{bmatrix}$$

By combining the Dm and Km the following matrix has been generated

$$Sm(sekar)=\begin{bmatrix}3&1&3&8\\1&5&1&5\\2&2&1&2\\1&1&2&9\\2&9&1&1\end{bmatrix}. \text{ This gives } S^Tm(sekar)=\begin{bmatrix}3&1&2&1&2\\1&5&2&1&9\\3&1&1&2&1\\8&5&2&9&1\end{bmatrix}.$$

Considering a row at a time the decimal pattern formed from the matrix is:

3121215219311218529 1 (4)

When an intruder tracks the continuous data pattern, it is so easy to hack the key text. Because, as above approach follows attaching the key text as its original the second part of the pattern for all data is same.

For Example,

The pattern for 'saran' from (1) is 31212119153112185291 and

The pattern for 'sekar' from (4) is 31212152193112185291.

If these decimal patterns are converted to equivalent binary pattern, all patterns have similar second half. From the similar bit pattern the hacker may easily identify the common portion and easily hack the key. Hence the proposed system moved to the next level, which increases the complexity.

Case 4: Security Matrix generated by adding columns

In the next recipe of the proposed work the key matrix is encoded using addition to the data matrix. By this process for two different data the generated pattern will be different.

Consider the data 'saran' and 'sekar' with the key 'zebra'. The matrices are

$$Dm(saran)=\begin{bmatrix}3&1\\1&1\\2&9\\1&1\\2&5\end{bmatrix} Dm(sekar)=\begin{bmatrix}3&1\\1&5\\2&2\\1&1\\2&9\end{bmatrix} Km(zebra)=\begin{bmatrix}3&8\\1&5\\1&2\\2&9\\1&1\end{bmatrix}$$

$$Sm(saran)=\begin{bmatrix}3&1&3+3&1+8\\1&1&1+1&1+5\\2&9&2+1&9+2\\1&1&1+2&1+9\\2&5&2+1&5+1\end{bmatrix}=\begin{bmatrix}3&1&6&9\\1&1&2&6\\2&9&3&11\\1&1&3&10\\2&5&3&6\end{bmatrix}. \text{ Then } S^Tm(saran)=\begin{bmatrix}3&1&2&1&2\\1&1&9&1&5\\6&2&3&3&3\\9&6&11&10&6\end{bmatrix}$$

Then the derived decimal pattern for the data 'saran' is 3121211915623339611106 (5)

Similarly,

$$Sm(sekar)=\begin{bmatrix}3&1&3+3&1+8\\1&5&1+1&5+5\\2&2&2+1&2+2\\1&1&1+2&1+9\\2&9&2+1&9+1\end{bmatrix}=\begin{bmatrix}3&1&6&9\\1&5&2&10\\2&2&3&4\\1&1&3&10\\2&9&3&10\end{bmatrix}. \text{ Then } S^Tm(sekar)=\begin{bmatrix}3&1&2&1&2\\1&5&2&1&9\\6&2&3&3&3\\9&10&4&10&10\end{bmatrix}.$$

Considering a row at a time the decimal pattern formed from the matrix is:

3121215219623339104 1010 (6)

Case 5: Security Matrix generated by subtracting columns

In the next recipe of the proposed work the key matrix is encoded using addition to the data matrix. By this process for two different data the generated pattern will be different.

Consider the data 'saran' and 'sekar' with the key 'zebra'. The matrices are

$$Dm(saran)=\begin{bmatrix}3&1\\1&1\\2&9\\1&1\\2&5\end{bmatrix} Dm(sekar)=\begin{bmatrix}3&1\\1&5\\2&2\\1&1\\2&9\end{bmatrix} Km(zebra)=\begin{bmatrix}3&8\\1&5\\1&2\\2&9\\1&1\end{bmatrix}$$

$$Sm(saran)=\begin{bmatrix}3&1&3-3&1-8\\1&1&1-1&1-5\\2&9&2-1&9-2\\1&1&1-2&1-9\\2&5&2-1&5-1\end{bmatrix}=\begin{bmatrix}3&1&0&-7\\1&1&0&-4\\2&9&1&7\\1&1&-1&-8\\2&5&1&4\end{bmatrix}. \text{ Then } S^Tm(saran)=\begin{bmatrix}3&1&2&1&2\\1&1&9&1&5\\0&0&1&-1&1\\-7&-4&7&-8&4\end{bmatrix}$$

Then the derived decimal pattern for the data 'saran' is 3121211915001-11-7-47-84(5)
Similarly,

$$Sm(sekar)=\begin{bmatrix}3 & 1 & 3-3 & 1-8\\ 1 & 5 & 1-1 & 5-5\\ 2 & 2 & 2-1 & 2-2\\ 1 & 1 & 1-2 & 1-9\\ 2 & 9 & 2-1 & 9-1\end{bmatrix}=\begin{bmatrix}3 & 1 & 0 & -7\\ 1 & 5 & 0 & 0\\ 2 & 2 & 1 & 0\\ 1 & 1 & -1 & -7\\ 2 & 9 & 1 & 8\end{bmatrix}.\ \text{Then } S^{T}m(sekar)=\begin{bmatrix}3 & 1 & 2 & 1 & 2\\ 1 & 5 & 2 & 1 & 9\\ 0 & 0 & 1 & -1 & 1\\ -7 & 0 & 0 & -7 & 8\end{bmatrix}.$$

Considering a row at a time the decimal pattern formed from the matrix is:
3121215219001-11-700-78 (7)

Case 6: Security Matrix generated by subtracting minimal element

In the next recipe of the proposed work the key matrix is encoded using addition to the data matrix. By this process for two different data the generated pattern will be different.

Consider the data 'saran' and 'sekar' with the key 'zebra'. The matrices are

$$Dm(saran)=\begin{bmatrix}3 & 1\\ 1 & 1\\ 2 & 9\\ 1 & 1\\ 2 & 5\end{bmatrix}Dm(sekar)=\begin{bmatrix}3 & 1\\ 1 & 5\\ 2 & 2\\ 1 & 1\\ 2 & 9\end{bmatrix}Km(zebra)=\begin{bmatrix}3 & 8\\ 1 & 5\\ 1 & 2\\ 2 & 9\\ 1 & 1\end{bmatrix}$$

To generate the security matrix, the minimal element among data matrix and key matrix has been considered. For the instance, for 'saran', the minimum element is '1' is subtracted from first and second columns the data and taken as third and fourth columns.

$$Sm(saran)=\begin{bmatrix}3 & 1 & 3-1 & 1-1\\ 1 & 1 & 1-1 & 1-1\\ 2 & 9 & 2-1 & 9-1\\ 1 & 1 & 1-1 & 1-1\\ 2 & 5 & 2-1 & 5-1\end{bmatrix}=\begin{bmatrix}3 & 1 & 2 & 0\\ 1 & 1 & 0 & 0\\ 2 & 9 & 1 & 8\\ 1 & 1 & 0 & 0\\ 2 & 5 & 1 & 4\end{bmatrix}.\ \text{Then } S^{T}m(saran)=\begin{bmatrix}3 & 1 & 2 & 1 & 2\\ 1 & 1 & 9 & 1 & 5\\ 2 & 0 & 1 & 0 & 1\\ 0 & 0 & 8 & 0 & 4\end{bmatrix}$$

Then the derived decimal pattern for the data 'saran' is 31212119152010100804 (8)
Similarly,

$$Sm(sekar)=\begin{bmatrix}3 & 1 & 3-1 & 1-1\\ 1 & 5 & 1-1 & 5-1\\ 2 & 2 & 2-1 & 2-1\\ 1 & 1 & 1-1 & 1-1\\ 2 & 9 & 2-1 & 9-1\end{bmatrix}=\begin{bmatrix}3 & 1 & 2 & 0\\ 1 & 5 & 0 & 4\\ 2 & 2 & 1 & 1\\ 1 & 1 & 0 & 0\\ 2 & 9 & 1 & 8\end{bmatrix}.\ \text{Then } S^{T}m(sekar)=\begin{bmatrix}3 & 1 & 2 & 1 & 2\\ 1 & 5 & 2 & 1 & 9\\ 2 & 0 & 1 & 0 & 1\\ 0 & 4 & 1 & 0 & 8\end{bmatrix}.$$

Considering a row at a time the decimal pattern formed from the matrix is:

31212152192010104 1048 (9)

The sum of digits of the decimal pattern is 48. And this pattern has the equivalent binary pattern as follows.

10010110000110000110111000011111100000010001001010010010101 11 (10)

In this example, the binary pattern has got 27 1's in the total of 62 binaries. The same has to be checked using magic cards.

Magic cards are simple mxn matrix, generated using magic cards generation algorithm. The total number of cards generated according to the requirements. Each card will be given an alphabetical identifiers as A, B, C etc.,. The same has incorporated for the reference in this research work. This is first research work uses magic cards for checking errors and validating the received patterns at the receiver side. The validation has done in two levels, one for binary pattern and another for decimal pattern.

For this checking process the sender has to send the list of magic cards where the number 27 appeared. The number 27 has identified by the receiver through the list of cards B, C, D, E, F and G. In the above said cards, some random numbers are appeared. The receiver has to take the least number of each magic card and to take resultant sum, so that to get the result has 27.

Card - A

28	36	55	1	66
78	91			

Card - B

14	20	27	28	36
44	54	55	2	65
66	77	78	90	91

Card - C

14	20	25	27	28
33	36	42	44	52
54	55	63	3	65
66	75	77	78	88
90	91			

Card - D

14	20	22	25	27
28	33	36	39	42
44	49	4	52	54
55	60	63	65	66
72	75	77	78	85
88	90	91		

Card - E

14	18	20	22	25
26	27	28	33	35
36	39	42	44	45
49	52	54	55	56
60	5	63	65	66
68	72	75	77	78
81	85	88	90	91

Card - F

18	20	21	22	25
26	27	28	30	33
35	36	39	40	42
44	45	49	51	52
54	55	56	60	63
65	66	68	6	72
75	76	77	78	81
85	88	90	91	

Card - G

15	18	21	22	24
25	26	27	28	30
33	34	35	36	39
40	42	44	45	49
51	52	54	55	56
57	60	63	65	66
68	70	72	75	76
77	78	7	81	85
88	90	91		

Card - H

15	17	21	24	26
30	33	34	35	36
38	39	40	42	44
45	49	50	51	52
54	55	56	57	60
63	65	66	68	70
72	75	76	77	78
81	85	8	88	90
91				

Card - I

17	19	24	30	34
35	38	39	40	42
44	45	49	50	51
52	54	55	56	57
60	63	65	66	68
70	72	9	75	76
77	78	81	85	88
90	91			

Card - J

19	34	38	40	45
46	49	50	51	52
54	55	56	57	60
63	65	66	68	70
72	75	76	77	78
81	85	88	10	90
91				

Card - K

23	38	46	50	51
56	57	60	63	65
66	68	70	72	75
76	77	11	78	81
85	88	90	91	

Card - L

23	46	50	57	68
70	72	75	76	12
77	78	81	85	88
90	91			

Card - M

46	70	13	76	81
85	88	90	91	

Figure-4. Magic cards.

Similarly, for the decimal pattern the list of cards have to be sent with data stream to check the error in decimal pattern.

4. RESULTS AND DISCUSSIONS

In this research work, the proposed methodology reveals plenty numbers of philosophies that are furnished for the kind notice. In case 1, while doing the compression using the traditional Huffman conversion tree, small numbers have got four bits and big numbers have got only two bits. In the decimal pattern the small numbers occurred 'n' number of times and the large numbers occur in minimum. When the decimal pattern was converted to a binary pattern, this leads to a lengthy bit pattern. To avoid

this problem, in case 2, the work proposes the modified Huffman tree that is all exhibited in Fig 1 and Fig 2, literally slashes down the bit pattern up to 14%. In case 3, the security matrix for the given data has generated using a simple combination of data and key metrics. This produces a common pattern in the second half. To avoid this common pattern, in case 4, the columns of data and key matrices are added for generating security matrix. But, the elements of the resultant matrix produce number more than 9, because, in Huffman theory the threshold permissible from 0-9. To resolve the above said, in case 5, the security matrix has been created by subtracting the key matrix from the data matrix. This idea has produced many negative digits in the decimal pattern which is also not on the threshold. To stick into the threshold the proposed work applies subtracting the minimal set of pairs among data and key matrix as presented in case 6. To validate the decompressed data magic cards were incorporated.

5. CONCLUSIONS

This research paper has proposed a methodology for identifying correct pattern in key management processes. From the various cases of analyzing the bit pattern, this work reveals the setbacks including threats with binary data which has common bit pattern, encoding the negative number and identifying the right pattern at the receiver side. In each case the complexity level of authentication and encryption have been increased at remarkable rate. After decompression the valid pattern has spotted out by two levels of validation. In first level the number of 1's in binary pattern was used and in second level the sum of digits in the decimal digits was used. Both numbers are hided in the magic information and this information has encapsulated with packet sent over the communication channel. Using the magic square information the receiver can discover the right binary pattern if it is needed. The output of the research work has been proved and showed that the complexity level of authentication has increased prominently.

REFERENCES

[1] Hamouid K. and Adi K. 2015. Efficient certificateless web-of-trust model for public-key authentication in Manet. Computer Communications. 63: 24-39.

[2] Lee K., Yeuk H., Kim J., Park H. andYim K. 2016. An efficient key management solution for privacy masking, restoring and user authentication for video surveillance servers. Computer Standards and Interfaces.44: 137-143.

[3] Xiaohong Z. andYingmeng H. 2015.RFID mutual-authentication protocol with synchronous updated-keys based on Hash function. The Journal of China Universities of Posts and Telecommunications. 22(6): 27-35.

[4] Lai C., Li H., Lu R. andShen X. S. 2013. SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks. Computer Networks. 57(17): 3492-3510.

[5] Somanatha R. B. and Atwood J. W. 2015. Router authentication, key management and adjacency management for securing inter-router control messages. Computer Networks. 79: 68-90.s

[6] Degefa F. B., Lee D., Kim J., Choi Y. and Won D. 2015. Performance and security enhanced authentication and key agreement protocol for SAE/LTE network. Computer Networks.

[7] Debiao H., Jianhua C. and Jin H. 2012.An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security. Information Fusion. 13(3): 223-230.

[8] He D. 2012. An efficient remote user authentication and key agreement protocol for mobile client–server environment from pairings. Ad Hoc Networks. 10(6): 1009-1016.

[9] Farash M. S., Turkanović M., Kumari S. andHölbl M. 2016. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. Ad Hoc Networks. 36: 152-176.

[10] Amin R. andBiswas G. P. 2016. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Networks. 36: 58-80.

[11] Sun X., Wu X., Huang C., Xu Z. andZhong J. 2016. Modified access polynomial based self-healing key management schemes with broadcast authentication and enhanced collusion resistance in wireless sensor networks. Ad Hoc Networks. 37: 324-336.

[12] Lu R., Lin X., Liang X. andShen X. 2012.A dynamic privacy-preserving key management scheme for location-based services in vanets. Intelligent Transportation Systems, IEEE Transactions on. 13(1): 127-139.

[13] Han C. K. and Choi H. K. 2014.Security analysis of handover key management in 4G LTE/SAE networks. Mobile Computing, IEEE Transactions on. 13(2): 457-468.

[14] Shen J., Tan H., Moh S., Chung I., Liu Q. and Sun X. 2015. Enhanced secure sensor association and key management in wireless body area networks. Communications and Networks, Journal of. 17(5): 453-462.

[15] Doh I., Chae K., Lim J. and Chung M. Y. 2015. Authentication and Key Management Based on Kerberos for M2M Mobile Open IPTV Security.

Intelligent Automation and Soft Computing. 21(4): 543-558.

[16] Lv X., Mu Y. and Li H. 2015. Key management for Smart Grid based on asymmetric key-wrapping. International Journal of Computer Mathematics. 92(3): 498-512.

[17] Omotosho A. and Emuoyibofarhe J. 2015.Private Key Management Scheme Using Image Features. Journal of Applied Security Research, 10(4): 543-557.

[18] Lv X., Li H. and Wang B. 2014.Authenticated asymmetric group key agreement based on certificateless cryptosystem. International Journal of Computer Mathematics. 91(3): 447-460.

[19] Chandramowliswaran N., Srinivasan S. and Muralikrishna P. 2015. Authenticated key distribution using given set of primes for secret sharing. Systems Science and Control Engineering. 3(1): 106-112.

[20] Thiruppathy Kesavan V. and Radhakrishnan S. 2015. Secure clustering and routing for heterogeneous mobile wireless sensor networks with dynamic key management. Journal of Experimental and Theoretical Artificial Intelligence. pp. 1-18.