Matrix based Key Generation to Enhance Key Avalanche in Advanced Encryption Standard

Paul A.J.
Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India.

Mythili P.
Cochin University of Science and Technology, Kochi, Kerala, India.

Paulose Jacob K.
Cochin University of Science and Technology, Kochi, Kerala, India.

ABSTRACT
In symmetric block ciphers, substitution and diffusion operations are performed in multiple rounds using sub-keys generated from a key generation procedure called key schedule. The key schedule plays a very important role in deciding the security of block ciphers. In this paper we propose a complex key generation procedure, based on matrix manipulations, which could be introduced in symmetric ciphers. The proposed key generation procedure offers two advantages. First, the procedure is simple to implement and has complexity in determining the sub-keys through crypt analysis. Secondly, the procedure produces a strong avalanche effect making many bits in the output block of a cipher to undergo changes with one bit change in the secret key. As a case study, matrix based key generation procedure has been introduced in Advanced Encryption Standard (AES) by replacing the existing key schedule of AES. The key avalanche and differential key propagation produced in AES have been observed. The paper describes the matrix based key generation procedure and the enhanced key avalanche and differential key propagation produced in AES. It has been shown that, the key avalanche effect and differential key propagation characteristics of AES have improved by replacing the AES key schedule with the Matrix based key generation procedure.

General Terms
Information security, Key schedule, Symmetric-cipher, Secure communication.

Keywords
Ciphertext, Encryption, Plaintext, Key avalanche, Secret key, Sub-key.

1. INTRODUCTION
Secure communication of information over insecure communication channels requires some kind of encoding to deal with security attacks [1]. Encryption is a powerful tool to provide information security [2]. There are two classes of cryptographic procedures in use, referred to as i) Symmetric-key cryptography (SKC) and ii) Public key cryptography (PKC). Public-key algorithms are slow, whereas Symmetric-key algorithms generally run much faster [3]. Symmetric-key cryptography has been (and still is) extensively used to solve the traditional problem of communication over insecure channels [4]. The block ciphers such as DES (Data Encryption Standard) [5], AES (Advanced Encryption Standard) [6], and EES (Escrowed Encryption Standard) [7] are used for information security services worldwide. A desirable feature of a block cipher is that a small change either in the plaintext or in the secret key should produce a significant change in the output ciphertext [8] block, called avalanche effect. The avalanche effect is achieved using a complex key generation procedure and powerful encryption primitives in cryptographic transformation algorithms. Even though the Data Encryption Standard, with its key size of 56 bits, is not secure enough today, it exhibits strong avalanche properties that any good cipher is expected to have. We have already proposed a matrix based cryptographic transformation that has high conversion speed [9] and simple key generation procedure using matrix in [10 & 11]. In this paper a complex key generation procedure based on matrix manipulations, for symmetric block ciphers, is proposed. The proposed key generation procedure offer two advantages. First, the procedure is simple to implement and has complexity in determining the sub-keys by crypt analysis. Secondly, the procedure produces a strong avalanche effect making many bits in the output block of a cipher to undergo changes with one bit change in the secret key. A strong key avalanche facilitates better diffusion of changes in key value on the ciphertext generated by the cipher and enhances the security of the cipher. As a case study, the key generation procedure of Advanced Encryption Standard (AES), a block cipher, has been replaced by the matrix based key generation procedure to evaluate the key avalanche and differential key propagation produced in the cipher. AES with 128 bits secret key and 10 rounds of diffusion operation is considered here. The paper describes the key generation procedure and discusses the key avalanche effect and differential key propagation produced in AES. Rest of the paper is organized in the following sections. In section 2: key-generation procedure is explained with algorithm and flow diagram. In section 3: performance results are presented highlighting the improvements obtained in Advanced Encryption Standard and conclusions are made in section 4:

2. KEY GENERATION PROCEDURE
The key Generation (key scheduling) procedure is based on a matrix initialized using secret key. The values of sub-keys used in various diffusion rounds are taken from selected rows and columns of this matrix. The selection of rows and columns for this purpose is based on the secret key value and other functional logic as explained in the following sub sections.

2.1 Nomenclature
M[i][j]–Element of matrix *M* with row i and column j
K(i) – ith character of secret key, *K*
Ks1(r), Ks2(r) – sub-keys used in rth round
P – Plaintext, *C*– Ciphertext,
ΔK – change in secret key value

2.2 Matrix Initialization

A matrix *M* with 16 rows and 256 columns is defined. Each column of every row is filled with a number between 0 and 255 (both the numbers included) in an order depending on the characters of secret key. The first column in the i[th] row of the matrix is filled with ASCII code of i[th] character of the secret key, *K* (that is, *M[i][1]* = Integer value of *K[i]*). The subsequent columns of the i[th] row of the matrix are filled with numbers that have increments of 1 from the previous column value till the number is 255. Subsequent columns are filled with numbers starting from 0 and ending with ASCII code of the i[th] character of secret key minus 1. The distribution of characters in the columns of all the sixteen rows of the matrix thus becomes key dependent. Without knowing the secret key the element in a column of any row of the matrix *M* cannot be determined by an adversary. Plate 1: shows the matrix initialization pseudo code.

```
For i ← 0 to 15   // rows
For j ← 0 to 255   // columns
M[i][j] = (int)K[i] + j
If M[i][j] > 255 { M[i][j] = M[i][j] – 256 }
EndFor  // columns
EndFor  // rows
```

Plate 1. Matrix initialization pseudo code

2.3 Sub key Generation

Sub-keys used in round operations are generated by key scheduling procedure. In this procedure two sub-key matrices Ks_1 and Ks_2 *(of size 16 * 16)* are derived from the base matrix *M.* These pairs of key can be used in substitution and diffusion operations performed in a typical block cipher. It is desirable that the key scheduling be a complex procedure so that an adversary must find it extremely difficult to derive the sub-keys during crypt analysis. Another desirable feature of key schedule is that a small change in the secret key should get well diffused in to the sub-keys. This means that one bit change in secret key should cause many bits to change in sub-keys. These two desirable features are considered while designing the key scheduling procedure. The procedure is explained in steps as follows:

1) Secret key, *K*, is transposed (T) to get *K1*. It is a byte-level transposing operation performed in this process where by the LS byte takes the place of MS byte position and the MS byte takes the LS byte position after the transpose operation. For example, if, bytes in array, *K*, is {K0,K1,K2,K3,K4,K5,….K14,K15} then after performing the transpose operation, *K1 = K* Transposed, the contents of *K1* will become {K15, K14,….K5,K4,K3,K2,K1,K0}.
2) *K1* is XOR ed with *K* to get *K2*. This operation can cause up to 2 bits to change in *K2* when 1 bit is changed in secret key *K*.
3) Left half of *K2* and right half of *K2* is XOR ed to get *K3.*
4) Transposed left half of *K2* and transposed right half of *K2* are XOR ed to get *K4*.
5) *K3* and *K4* are concatenated to get *K5*.
   With this operation 1 bit change in secret key, *K*, can cause up to 4 bits to change in *K5.*
6) Sum of integer values of bytes in *K5* is calculated to get *L*
7) *Kse1* is calculated such that *Kse1 = L* % 23.
   When secret key has 1 bit change, *Kse1* can have up to 4 counts change.
8) *Kse2* is calculated such that *Kse2 = L* % 15.
   When secret key has 1 bit change, *Kse2* can have up to 4 counts change.
   (*Kse1 + Kse2*) can have up to 8 counts change with one bit change in secret key.

Steps 1 through 8 in the key scheduling procedure are shown in figure 1.

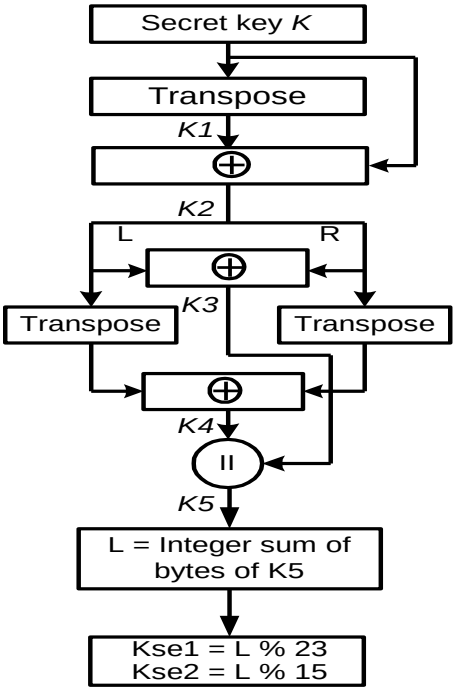

Figure 1. Flow chart of steps 1 through 8

9) Two matrices Ks1 and Ks2 of size 16 x 16 are derived from the base matrix, M, such that
   Ks1[row][column]=M[row][Kse1+Kse2+column]
   Ks2[row][column]=M[row][Ks1[row][column]]
   Columns of *Ks1* matrix are chosen from the base matrix *M* depending upon *Kse1* and *Kse2* Values. Here, an element of *Ks1* can have up to 8 counts change with one bit change in secret key.
   Columns of *Ks2* matrix are chosen from the base matrix *M* depending upon element values of columns of *Ks1* matrix. An element of *Ks2* can have up to 8 counts change with one bit change in secret key.
10) Ks1[row][column]=M[row][Ks2[row][column]]
    Columns of Ks1 matrix are chosen from the base matrix M depending upon element values in columns of Ks2 matrix.
    The regeneration of sub-key matrix, Ks1, is carried out in order to achieve further indirection for adding complexity.
11) Rotate vertically down i[th] column of matrix *Ks1* number of times equal to *((int(K[i]) % 12) + Kse1)*.
12) Rotate vertically down i[th] column of matrix Ks2 number of times equal to *((int(K[i]) % 10) + Kse1)*. The vertical rotations shuffle the elements of sub-key matrices thereby providing more changes in the sub-key values while one bit change is applied on the original secret key, *K*.

This procedure facilitates many bits to change in the sub-keys

due to one bit change in the secret key. This is a desirable feature of any key scheduling procedure that can produce high diffusion and hence enhances the security of the cipher. The sub-keys, *Ks1* and *Ks2*, for round operations (round 1: through round 10:), generated from a given secret key, *K* are shown in plate. 2 and plate. 3. The ten values shown under the heading key schedule represents the value of sub-keys (in hex format) to be used in ten rounds.

Secret Key, *K*: 4C 69 66 65 27 73 20 62 65 61 75 74 69 66 75 6C

Key schedule *(Ks1):*

1: 6D 6A 49 40 6D 52 F2 49 40 3D 83 67 6E 34 3D 31
2: 68 6F 35 3E 32 6E 6B 4A 41 6E 53 F3 4A 41 3E 84
3: 3F 33 6F 6C 4B 42 6F 54 F4 4B 42 3F 85 69 70 36
4: 34 70 6D 4C 43 70 55 F5 4C 43 40 86 6A 71 37 40
5: 6E 4D 44 71 56 F6 4D 44 41 87 6B 72 38 41 35 71
6: 39 42 36 72 6F 4E 45 72 57 F7 4E 45 42 88 6C 73
7: 74 3A 43 37 73 70 4F 46 73 58 F8 4F 46 43 89 6D
8: 50 47 74 59 F9 50 47 44 8A 6E 75 3B 44 38 74 71
9: 39 75 72 51 48 75 5A FA 51 48 45 8B 6F 76 3C 45
10: 49 76 5B FB 52 49 46 8C 70 77 3D 46 3A 76 73 52

Plate 2. Secret key *K* and Sub-key *ks1* for 10 rounds

Secret Key, *K*: 4C 69 66 65 27 73 20 62 65 61 75 74 69 66 75 6C

Key schedule *(Ks2):*

1: D8 D0 F8 F6 E0 DA F8 E6 A6 E0 DA D8 5C F4 4E D2
2: D1 F9 F7 E1 DB F9 E7 A7 E1 DB D9 5D F5 4F D3 D9
3: E2 DC FA E8 A8 E2 DC DA 5E F6 50 D4 DA D2 FA 8
4: DD FB E9 A9 E3 DD DB 5F F7 51 D5 DB D3 FB F9 E3
5: F8 52 D6 DC D4 FC FA E4 DE FC EA AA E4 DE DC 60
6: D5 FD FB E5 DF FD EB AB E5 DF DD 61 F9 53 D7 DD
7: E6 E0 FE EC AC E6 E0 DE 62 FA 54 D8 DE D6 FE FC
8: 55 D9 DF D7 FF FD E7 E1 FF ED AD E7 E1 DF 63 FB
9: E2 00 EE AE E8 E2 E0 64 FC 56 DA E0 D8 00 FE E8
10: DB E1 D9 01 FF E9 E3 01 EF AF E9 E3 E1 65 FD 57

Plate 3. Secret key *K* and Sub-key *ks2* for 10 rounds

3. PERFORMANCE IN AES

Advanced Encryption Standard algorithm has been modified by replacing the original AES key schedule with the proposed matrix based key generation procedure. It may be noted, here, that AES use only one set of 10 sub keys for the 10 diffusion rounds when the secret key size chosen is 128 bits. Using sub key set, *Ks1*, AES has been tested to evaluate the following performance criteria.

- Effect of 1 bit key change on sub-keys
- Key avalanche characteristics of AES
- Propagation of *ΔK* through data in AES

3.1 Effect of 1 bit Key Change on Sub-keys

Tests conducted to obtain the effect of 1 bit change in secret key on sub-keys would give an indication of the effectiveness of the key scheduling procedure. The number of bit changes in sub-keys due to one bit change in secret key has been observed. First, the key scheduling procedure has been executed with a given secret key and the sub-keys generated for 10 rounds have been recorded. Then, with another secret key, with a difference of only 1 bit (one count) from the first key, has been used to execute the key schedule and the sub-keys generated for 10 rounds has been recorded. The number of bits changed in sub-keys, in each round, has been calculated from the recordings and the result has been plotted. Fig 2: shows the number of bit changes in sub-keys generated by the proposed key schedule compared with the number of bit changes produced in sub-keys in AES.

3.2 Key Avalanche Characteristics

A block of plaintext data (128 bits or 16 characters of plaintext) has been used as input to the cipher in this test. With a given secret key, K the cipher has been executed. The output block produced in each round has been recorded. Then, with the same block of input plaintext, and a secret key value that differs by one bit has been used to execute the cipher. The output block produced in each round has been recorded. The number of bit changes that occurred in each round has been calculated. The number of bit changes, in each round, due to one bit change in the secret key value has been plotted. The key avalanche has been obtained in the case of original AES and AES modified with Matrix based key generation procedure. Fig. 3: shows the change of bits, in a block, for one bit change in key produced in AES and the same in AES with Matrix based key generation. It has been shown that AES with Matrix based key generation procedure is able to achieve enhanced key avalanche characteristics.

3.3 Propagation of ΔK through Rounds

The differential propagation of key through round outputs is presented here. A block of plaintext data (128 bits or 16 characters of plaintext) has been used as input to the cipher in this test. With a given secret key, *K*, the AES cipher has been executed. The output block produced in each round has been recorded. Then, with the same block of input plaintext and a secret key value that differs by one bit (*ΔK=1*) has been used to execute the AES cipher. The output block produced in each round has been recorded. The difference in byte values of the data blocks produced in respective round has been calculated. The differences in byte values showed how one bit change in secret key propagates through data in rounds. This is very important in assessing the resistance of the cipher against differential attacks. If the difference in byte value between round outputs due to one bit change (or for a given difference) in key value is not consistent then the cipher exhibits strength against differential crypt analysis. Fig. 4: shows the variation of difference in byte values (only one byte is shown in the graph. All bytes in a block exhibited similar characteristics) of the data blocks produced by each round due to one bit change in secret key. The figure indicates that the difference propagation is better in AES with matrix based key schedule

4. CONCLUSIONS

The Matrix based key generation procedure, incorporated in Advanced Encryption Standard is capable of generating effective sub-keys needed for all 10 rounds of substitution and diffusion operations in the cipher. The propagation of *ΔK* through the sub-keys exhibits enhanced key avalanche effect. The key avalanche effect produced on data blocks in diffusion

rounds has improved in AES with Matrix based key generation procedure. Propagation of ΔK through bytes of data block in diffusion rounds is also better in AES with Matrix based key generation indicating added resistance against differential attacks on the cipher. The key scheduling has complexity due to key dependant matrix element reference and multiple indirections in choosing data from matrix to form the sub-keys.

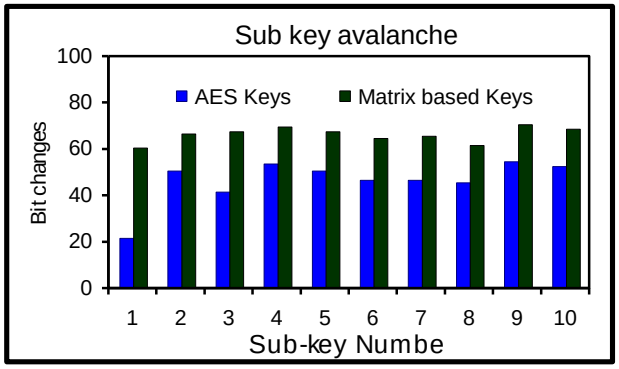

Figure 2. Effect of 1 bit change in secret key on sub-keys

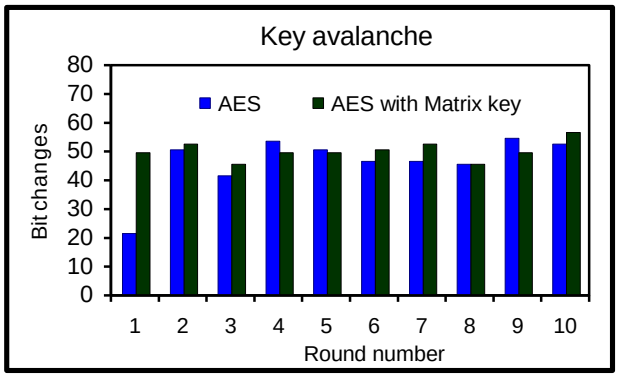

Figure 3. Key avalanche in AES with Matrix key schedule

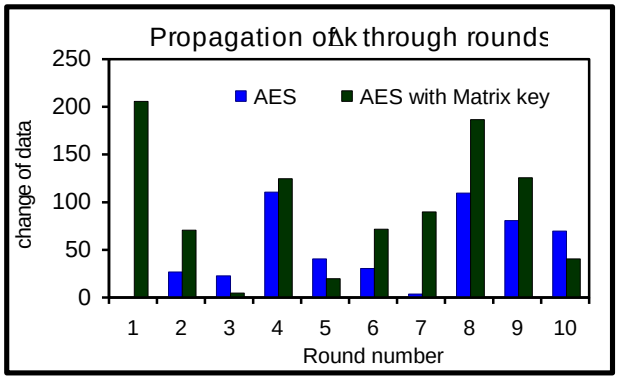

Figure 4. Propagation of Δ*K* through data block in rounds

The key based rotations, applied on the sub-key matrix, during the formation process, makes the determination of sub-keys harder by an adversary. Further increase in complexity of key scheduling can be attempted by more indirections applied while extracting sub-key matrices.

5. ACKNOWLEDGMENTS

The authors would like to acknowledge their deep gratitude to Dr. R. Gopikakumari, Head, Division of Electronics, School of Engineering, Cochin University of Science and Technology, Kochi, India for her valuable suggestions during reviews and encouragements given throughout the course of the research work.

6. REFERENCES

[1] William Stallings, "Network Security Essentials (Applications and Standards), " Pearson Education, pp. 2-80, (2004).

[2] Charles P. Pfleeger, Shari Lawrence Pfleeger, "Security in computing," Pearson Education, pp. 66-120, (2004).

[3] Jose J. Amador, Robert W. Green, "Symmetric-Key Block Ciphers for Image and Text Cryptography," International Journal of Imaging System Technology, Vol.15 – pp. 178- 188, (2005).

[4] Dragos Trinca, "Sequential and Parallel Cascaded Convolution Encryption with Local Propagation: Toward future Directions in Cryptography," Proceedings of The third International Conference on Information Technology-New Generations. (ITNG'06), 0-7695-2497- 4 / 2006, IEEE Computer Society, (2006).

[5] Data Encryption Standard :

http://csrc.nist.gov/publications/fips/fips46-3/fips- 46-3.pdf

[6] Advanced Encryption Standard:

http://csrc.nist.gov/publications/fips/fips197/ fips- 97.pdf

[7] Escrowed Encryption Standard:

http://csrc.nist.gov/publications/fips/fips1185/ fips-185.txt

[8] Krishnamurthy G.N, Ramaswamy V., Leela G.H, Ashalatha M.E, "Performance enhancement of Blowfish and CAST-128 algorithms and Security analysis of improved Blowfish algorithm using Avalanche effect," International Journal of Computer Science and Network Security, Vol.8, No. 3, March 2008, pp. 244-250.

[9] Paul A.J., Varghese Paul, P. Mythili, " Matrix Array Symmetric Key Encryption," Journal of Computer Society of India, Vol. 37, Issue No. 1, January – March 2007, pp. 48-53.

[10] Paul A.J., Varghese Paul, P. Mythili, " A Fast And Secure Encryption Algorithm for Message Communication," IETECH International Journal of Communication Techniques, Vol. 2, No. 3, 2008, pp 104-109.

[11] Paul A.J., Varghese Paul, P. Mythili, "Fast Symmetric Cryptography using Key and Data based Masking operations," International- Journal of Computational Intelligence - Research & applications, Vol 3, Number 1, January – June 2009, pp. 5-10.