Fortification of AES with Dynamic Mix-Column Transformation

Ghulam Murtaza[1], Azhar Ali Khan[2], Syed Wasi Alam[2], Aqeel Farooqi[3]

[1] National University of Science and Technology, Islamabad, Pakistan
[2] Sichuan University, Chengdu, China
[3] Center for Advanced Studies in Engineering (CASE), Islamabad, Pakistan
[1]azarmurtaza@hotmail.com
[2,3]{azhar_iiee,wasi.alam,aqeel_farooqi}@yahoo.com

Abstract. MDS Matrix has an important role in the design of Rijndael Cipher and is the most expensive component of the cipher. It is also used as a perfect diffusion primitive in some other block ciphers. In this paper, we propose a replacement of Mix Column Transformation in AES by equivalent Dynamic Mix Column Transformation. A Dynamic Mix Column Transformation comprises dynamic MDS Matrices which are based on default MDS Matrix of AES and *m*-bit additional key. Here *m* is a variable length that does not exceed the product of 31.97 and one less the number of encryption rounds. This mechanism increases a brute force attack complexity by *m*-bit to the original key and enforces the attackers to design new frameworks for different modern cryptanalytic techniques applicable to the cipher. We also present efficient implementation of this technique in Texas Instrument's DSP C64x+ with no extra cost to default AES and in Xilinx Spartan3 FPGA with no change in AES throughput. We also briefly analyze the security achieved over it.

Keywords: Dynamic Mix-Column Transformation (*DMCT*), Dynamic MDS Matrix, AES, Attacks on Block Ciphers, DSP, FPGA.

1 Introduction

The block cipher Rijndael [1] was designed by Daemen and Rijmen and standardized by NIST in 2000 as the Advanced Encryption Standard (AES) [2]. The AES follows the wide-trail design Strategy [22, 23] and was initially proposed mainly due to its resistance against powerful cryptanalytic techniques namely Differential Cryptanalysis and Linear Cryptanalysis. The security of Rijndael has been subjected to confer [4, 5, 6 and 16] since its acceptance as an AES. In [26], Ferguson, Shroeppel and Whiting derive a closed formula for AES that can be seen as a generalization of continued fractions. In [15], Courtois and Pieprzyck observe that the S-box used in the AES can be described by a number of implicit quadratic Boolean equations. In [27], Murphy and Robshaw define the block cipher BES and

shows that AES can be embedded into BES which operates on data blocks of 128 bytes instead of bits. According to Murphy and Robshaw, the algebraic structure of BES is even more elegant and simple than that of the AES. In [25], Barkan and Biham introduce the concept of Dual ciphers which is basically a generalization of the embedding technique.

Recently, there are many significant developments in the cryptanalysis of AES. In [12], the authors describe a related key attack on AES-256 with complexity 2^{119}. This is a follow-up to an attack [9] discovered earlier in 2009 by Biryukov, Khovratovich, and Nikolic, with a complexity of 2^{96} for one out of every 2^{35} keys. In [13], a new attack against AES-256 presents that uses only two related keys and 2^{39}time to recover the complete 256-bit key of a 9-round version, or 2^{45}time for a 10-round version with a stronger type of related sub-key attack, or 2^{70}time for a 11-round version. 256-bit AES uses 14 rounds. In [14], Henri and Thomas attack on AES-128 using known-key distinguishing attack with a computation complexity2^{48}, and a memory complexity of 2^{32}.

In AES, Mix Column Transformation is the most expensive operation where input matrix is multiplied (Over f_2^k) with MDS Matrix. This transformation plays an important role with respect to the wide trail strategy of the cipher. It is a vital component of diffuser part [2] of the cipher. It also guarantees a high number of active S-boxes over a round. The quantitative measure of this transformation is the branch number of the MDS Matrix used. The branch number of 4 by 4 MDS Matrix is 5 (optimal), so as the branch number of AES MDS Matrix. This property ensures that a linear approximation or a differential of one round always involves at least five active S-boxes using any 4x4 MDS matrix in Mix Column transformation. In [7], Murtaza and Nassar show that scalar multiplication of an MDS matrix results in an MDS matrix. The consequence leads an idea to introduce a dynamic MDS Matrix in AES keeping in view the enhanced security and unaltered (at least close enough) efficiency of the algorithm.

Grand Cru [20] a candidate cipher algorithm of NESSIE [21] project, is based on AES-128 and the strategy of multiple layered security. This tactic is aimed at combining different security motivations and hence is a useful practice to improve security of widely used existing crypto algorithms against known modern cryptanalytic techniques. It also provides a noble way to use long keys in existing algorithms without redesigning key-schedule and possibly encryption algorithms. This idea seems noble until unless the proposed cipher is efficient enough comparing with original one. Although the proposed algorithm was not selected for second phase of NESSIE project due to the cost of speed but no weakness was found in the algorithm itself.

To achieve unchanged throughput with purposed modification we use two different implementation platforms, TI DM6443 and Xilinx Spartan 3.

Modern networked device require diverse multi-processors, which enables them to be used in media and general purpose applications. The TMS320DM6443 is one of the TI DaVinci technology which grasps all the new epoch networked applications. It has dual-core, TMS320C64x+ DSP core and an ARM926EJ-S core. Very-long-instruction-word (VLIW) architecture with 594MHz clock rate of C64x+ core allows developer to meet all high-performance DSP programming challenges

[29].The Galois field multiplication is one of the 17 new instructions for M-unit of C64x+ and is referred as GMPY4 in their instruction set. This processor can perform four parallel operations on 8-bit packed data. The GMPY4 instruction works on Galois field GF(2^n), where n can range between 1 and 8 using any generator polynomial. The Galois field polynomial generator function register controls the field size and the polynomial generator [30].

Field Programmable Gate Array (FPGA) is considered to be the replacement of ASIC. Application Specific Integrated Circuit (ASIC) is considered to be inexpensive but it is true only for bulk production therefore FPGA is suitable for most of the applications. Another advantage of FPGA is the availability of different hard IP cores in almost all modern FPGAs. These IP cores are useful to increase algorithm throughput. In this paper we use Dual port Block RAMs [31] to manage throughput of AES.

The rest of the paper is organized as follows. A very short description of AES-k algorithm is given in next section. In Section 3, the main result i.e. replacement of MCT with *DMCT* in AES encryption algorithm is introduced. Section 4 briefly presents security aspects of the purposed alteration in the algorithm. In section 5 hardware implementations and relative efficiencies are discussed.

2 Short overview of advanced encryption standard

Advanced Encryption Standard [1, 2] is a 128-bit block cipher with a k-bit secret key where k=128, 192 or 256. Encryption function of the cipher consists of 10-round with 128-bit key, 12-round with 192-bit key and 14-round with 256-bit key. In this paper, we use AES-128, AES-192 and AES-256 to denote these three variants respectively. Each round of encryption function of AES consists of following functions in sequence.

Substitution Byte (SB)
Byte wise substitution using 8×8 S-box.

Shift Row (SR)
A cyclic shift of the i^{th} row by i number of bytes to the left,
Where $i = 0 \ldots 4$

Mix Column Transformation (MCT)
An MDS matrix multiplication applied to each column. The final round of each AES-k does not include this transformation.

Add Round Key (AR)
An exclusive-or with the round key.

The secret key of AES is passed through a key schedule in order to generate $nr + 1$ rounds sub keys to be used in each round, where nr is the number of rounds [2].

3 Dynamic Mix-Column Transformation

In this section, we define a method to implement dynamic MDS Matrix in Mix-Column operation of AES encryption algorithm. The new Mix-Column operation is called as Dynamic Mix-Column Operation. Theorem 1 provides the basis of our design.

Theorem 1

Let $A = [a_{i,j}]_{mxm}$, $a_{i,j} \in F_q$ be an MDS Matrix, for an element $e \neq 0 \in F_q$, eA is an MDS Matrix [7].

A generalization of above result given in [32], for elements $e_i \neq 0 \in F_q, i = 1,2,\dots,m$, multiplying $i-th$ row of matrix A where $e_k$not necessarily be the same as $e_l$for any $1 \leq k,l \leq m$ is as follows.

Theorem 2

Let $A = \begin{bmatrix} A_1 \\ \vdots \\ A_m \end{bmatrix}, A_i = [\, a_{i,1} \quad \cdots \quad a_{i,n}]$, $a_{i,j} \in F_q$ be an MDS matrix, and $E = [\, e_i], i = 1,2,\dots,m$. then scalar multiplication $EA = \begin{bmatrix} e_1A_1 \\ \vdots \\ e_mA_m \end{bmatrix}$, $e_iA_i = [\, e_i\, a_{i,1} \quad \cdots \quad e_i\, a_{i,n}]$ is an MDS Matrix.

The following result proves the new method as a replacement of Mix-Column operation of AES.

Theorem 3

Let $f: (A, P) \rightarrow P'$ defined by $P' = A \times P$ be the mix column operation in AES where A is MDS Matrix, and P is input plaintext vector for mix-column operation. It is implemented using two 8-bit lookup tables or equivalent operation. Then $f: e'A \times P \rightarrow P''$ where $e'A = [e_1R_1 \quad e_2R_2 \quad e_3R_3 \quad e_4R_4]^T$; $R_i = [a_{i,1} \quad a_{i,2} \quad a_{i,3} \quad a_{i,4}]$ $i = 1,\dots,4$ is a mix column operation that can be implemented using six 8-bit lookup tables on software.

Proof

AES Mix-Column Transformation is based on an MDS matrix. Replacing AES MDS Matrix with another MDS Matrix of same size does not affect diffusion properties of the cipher. To show that new Mix-Column operation is equivalent to AES Mix-Column operation, we only need to ensure that new generated matrix is also an MDS matrix. Theorem 2 provides us this surety.

For number of required lookup tables or operations, we know that MDS Matrix of AES is circulatory one having two non identity elements. In AES Mix-Column Transformation, MDS matrix is applied on a 32-bit input as follows

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} 02.p_1 + 03.p_2 + 01.p_3 + 01.p_4 \\ 01.p_1 + 02.p_2 + 03.p_3 + 01.p_4 \\ 01.p_1 + 01.p_2 + 02.p_3 + 03.p_4 \\ 03.p_1 + 01.p_2 + 01.p_3 + 02.p_4 \end{bmatrix} = \begin{bmatrix} p'_1 \\ p'_2 \\ p'_3 \\ p'_4 \end{bmatrix} \quad (1)$$

The above multiplication in case of dynamic MDS matrix is given as

$$\begin{bmatrix} e_1.02 & e_1.03 & e_1.01 & e_1.01 \\ e_2.01 & e_2.02 & e_2.03 & e_2.01 \\ e_3.01 & e_3.01 & e_3.02 & e_3.03 \\ e_4.03 & e_4.01 & e_4.01 & e_4.02 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} =$$

$$\begin{bmatrix} e_1.02.p_1 + e_1.03.p_2 + e_1.01.p_3 + e_1.01.p_4 \\ e_2.01.p_1 + e_2.02.p_2 + e_2.03.p_3 + e_2.01.p_4 \\ e_3.01.p_1 + e_3.01.p_2 + e_3.02.p_3 + e_3.03.p_4 \\ e_4.03.p_1 + e_4.01.p_2 + e_4.01.p_3 + e_4.02.p_4 \end{bmatrix} =$$

$$\begin{bmatrix} e_1.(02.p_1 + 03.p_2 + 01.p_3 + 01.p_4) \\ e_2.(01.p_1 + 02.p_2 + 03.p_3 + 01.p_4) \\ e_3.(01.p_1 + 01.p_2 + 02.p_3 + 03.p_4) \\ e_4.(03.p_1 + 01.p_2 + 01.p_3 + 02.p_4) \end{bmatrix} = \begin{bmatrix} e_1.p'_1 \\ e_2.p'_2 \\ e_3.p'_3 \\ e_4.p'_4 \end{bmatrix} \quad (2)$$

Equations 1 and 2 shows that we need only 4 extra lookup tables, say T_{e_i} for elements $e_i, i = 1,2,3,4$ in a single round to implement new dynamic MDS Matrix. In AES encryption algorithm, there is 2 lookup tables $T_{02} and\ T_{03}$ so we require overall 6 lookup tables or equivalent operations for one round encryption process and $(nr-1) \times 4 + 2$ -2 look up tables for nr round of encryption where $nr = 10, 12\ or\ 14$for AES-128, AES-192 or AES-256 respectively.

In similar fashion, we need 4 extra lookup tables for AES decryption process. The process of inverse of such MDS matrices is given in [7].

Mechanism to implement Dynamic Mix-Column Transformation

For this mechanism, we necessitate 36, 44 and 52 non-zero independent random bytes to implement Dynamic Mix Column Transformation in AES-128, AES-192 and AES-256 respectively. These bytes are obtained by some arbitrary key-function. We do not present any specific method as it is beyond the scope of this paper. However one may get these random bytes from any secure random bit generator [18] which takes $m - bit$ as a key or may design a key expansion algorithm. Each consecutive 4 bytes are used in a round of AES as a 4 scalar multipliers for 4 rows of MDS Matrix, resulting in a new Matrix. In software, this procedure can be implemented more efficiently as described in theorem 3.

In DSP C64x+, The Random 32-bit value can be stored in a single variable. During AES process, we can load this random value in a register of any file and then use GMPY4 instruction.

For implementation on FPGA, our focus is to maintain throughput, as it is necessary for high speed design. According to the design rule of FPGA or ASIC, one has to focus the implementation either speed optimized or area optimized. Implementation presented in this paper is focused on throughput therefore our implementation is speed optimized.

4 Security Analysis

In this section, we consider two primary aspects of security. First one is the variation in statistical properties of AES by introducing DMCT and the other is the status of different modern cryptanalytic techniques regarding AES.

4.1 Variation in Statistical Properties of AES

Reference to theorem 1; every newly generated matrix is a MDS matrix of size 4x4 and hence has branch number 5. The role of mix column transformation is to ensure the maximum number of active s-boxes and mainly depends on branch number of MDS matrix. Furthermore changing the coefficient of MDS Matrix, the resultant cipher is isomorphic [25] to the original so it is obvious that there is no difference in statistical properties of AES by introducing dynamic mix column transformation.

4.2 Status of Modern Cryptanalytic Techniques

In contemporary cryptology, new methods have been developed to break the cipher algorithms hence any cipher algorithm needs to reveal resistance against these techniques of analysis and attacks.

In dynamic MDS Matrix, there are $(2^8 - 1)^4 \approx 2^{31.97}$possible MDS matrices in each round. So a total number of promising MDS Matrices for 9-round encryption of AES-128 are $2^{31.97^9}$. These MDS Matrices are based on bits generated by secure random bit generator using a $m - bit$ key value where length of m is proposed to be less than $31.97 \times (number\ of\ rounds - 1)$.We call this value as additional key. For modern cryptanalytic techniques, an attacker has to encounter these unknown key based matrices as well. In the following sub-sections we briefly discuss the different attack scenarios and how dynamic MDS matrices strengthen the security of AES although the actual complexity will be the matter of subject.

4.2.1 Differential and Linear Cryptanalysis

Differential cryptanalysis [11] is a chosen-plaintext attack and exploits the high probability of certain occurrences of plaintext differences, Δx and differences Δy, into the last round of the cipher, where notion of difference can be adapted to suite the cipher under attack. In case of AES, it is XOR differences. On the other hand,

linear cryptanalysis [10] is a known-plaintext attack that exploits large correlations between parities (linear combinations of bits) at the input of the last round with which a parity in the plaintext.

Differential/Linear trail over multiple rounds is a major requirement [2, 10, 11, 13] for differential/linear (and its variants) attack on it. In the presence of Dynamic MDS Matrices output differences depend on additional key used. Furthermore, for any two different additional key values, we will have dissimilar differentials over multiple rounds and hence diverse differential/linear trails. So it is impossible for an attacker to launch current state of the art differential/linear cryptanalysis techniques. New frameworks are required other than native method of differential/linear cryptanalysis to incorporate unknown permutations defined by dynamic mix column transformation.

For related key attacks [6, 12, 16, 17], an additional requirement to launch respective attack is the computation of MC or MC-inv for some particular round which is infeasible in presence of DMC.

4.2.2 Algebraic Cryptanalysis

In algebraic cryptanalysis [15], an attacker takes advantage of algebraic nature of the cipher to extract secret information. Typically algebraic attack consists of two steps, Collecting Step and Solving Step. In Collecting Step, appropriate algebraic equations for input/output pair are constructed which contain unknown parameters e.g. key bits (round keys). In Solving Step, these equations are solved with suitable method to obtain values of unknown variables. AES is very algebraic in nature and can be expressed with elegant equations in several ways [24].

Now we may describe the impact of $DMCT$ in cipher BES [27] as $C_B^{(k)} = \kappa_i^{(k)}.C_B^{(k)}; k = 0, \dots, 7; i = 0, \dots, (nr-1) \times 4$ where $C_B^{(k)}$ is the equation set after MCT of a round and κ is unknown row multiplier of MDS matrix for a round. This insertion will definitely increase complexity of the system. Further multivariate quadratic system of equations derived from BES (more simple as derived from AES [15]) becomes complicated as $\omega_i = \kappa_{4i+l} M_B x_{i-1} + K_i; i = 0, \dots, 9; l = 0, \dots, 3$. For continued fraction case discussed in [26], we have added complexity from $C = \omega_{i,e,d}$ to $C = \kappa_i^r \omega_{i,e,d}$. We refer [15, 26 and 27] for more details.

For other attempts where attacker may acquire additional key used to generate key stream for $DMCT$, he has to consider this key stream generation mechanism as a part of the constructed equations, resulting an increase in the complexity of the equations. This enhancement of security against algebraic cryptanalysis depends on the algebraic structure of key stream generator from additional key. Using operators other than XOR in key stream generator further enhances the complexity of constructed equations of cipher.

4.2.3 Square Attack

Square Attack [3], take advantage of square structure of a cipher where active bytes propagate over a few rounds.

Since we are just replacing MDS Matrix round to round, this does not have any effect on active/passive byte structure of the AES described by Square Attack. So this structure does not have any effect on naive square attack on 4-round AES. However for improved square attacks [19, 28], since inverse of MCT is required to carry out attack, $DMCT$ presents resistance as it depends on key and for one round there are $2^{31.97}$ possible MDS matrices. Intuitively there seems no method to point the correct one matrix for that round without knowing particular 32 key bits.

5 Hardware Implementation

In hardware, we implemented above theorem on both DSP and FPGA. Both have their own advantages in embedded systems.

5.1 DSP

For DSP C64x+, GFPGFR initialization will be in parallel of AES initialization. When Mix Colum is in process, GMPY4 instruction of new matrix can be carried out in parallel.

The mix column in AES proceeds as column-by-column. In each round, four column operations are performed and then add round key is the last step of a round.

In our implementation, the first Galois field multiplication (GMPY4) will be executed during the 2nd column operation and 2nd GMPY4 with 3rd column operation and so on. The last GMPY4 will be in parallel with "Add round key" of each round. These indicate that complete AES-128 encryption will use no extra cycle. Random values of *DMCT* in each round will not add any overhead in AES operations.

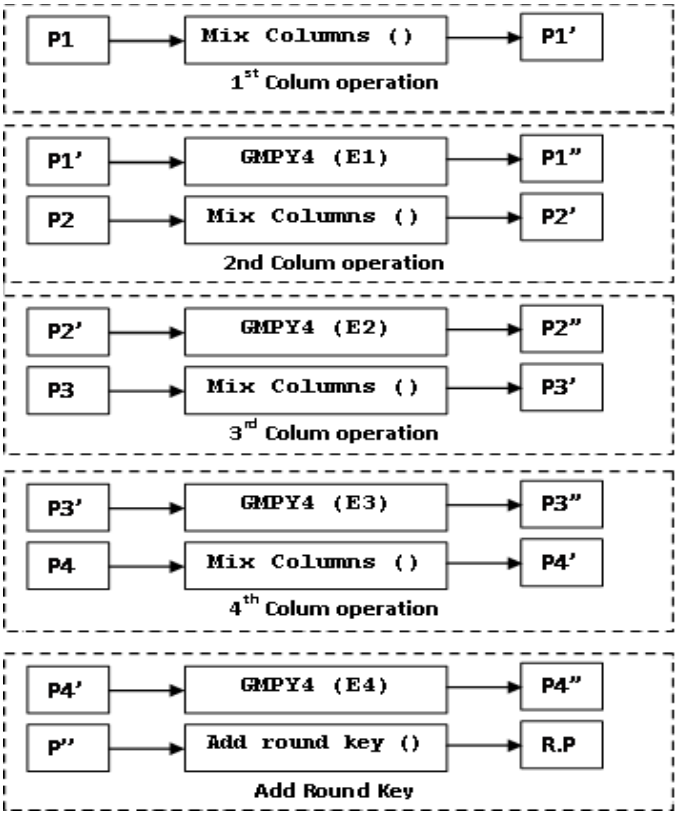

Fig. 1. Round Operations in DSP

5.2 FPGA

There are two parts of implementation of dynamic MDS, one is the generation of lookup tables and the other is the use of lookup tables in each round of the existing AES implementation. Thirty six lookup tables are required for nine rounds of AES, four for each round as explained in earlier sections. The generation of look up table starts in parallel with the key schedule in order to save time. To store the lookup table we use dual port Block RAMs. Thirty six lookup tables can be stored in eight dual port Block RAMs but it is not feasible to use eight Block RAMs as we can only access two elements at a time from a single dual port Block RAM. The requirement is to access four lookup tables at a time in order to maintain the throughput. Therefore we use sixteen dual port Block RAMs. Figure 2 is the block diagram of the implementation of generation of dynamic MDS matrixes.

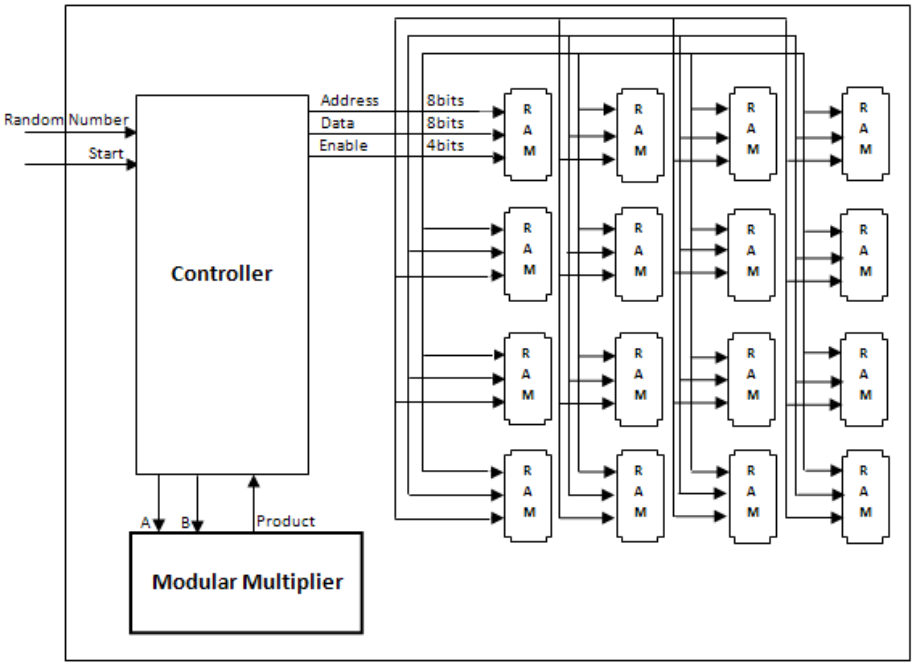

Fig. 2. Block Diagram of Dynamic MDS Generation Process

The basic implementation of AES on FPGA takes 42 cycles to generate 128 bits of cipher text. One cycle for each operation that is one cycle for Sub Byte, one for Shift Row, one for Mix Column and one for Add Round Key. On Spartan 3 FPGA with 100 MHz it gives the throughput of 304MHz. Although this design can run on speed more than 100 MHz but to keep the calculation simple we use 100 MHz for throughput calculation.

We utilize Dual port Block RAM to implement dynamic MDS matrix during encryption process. As explained earlier, we use sixteen Block RAMs during generation process of dynamic MDS so that four lookup tables can be accessed simultaneously in a single round. Because of redundant lookup tables we can perform Mix Column operation in single clock cycle thus it will not affect the throughput of AES.

6 Results:

These results are on the basis of ordinary implementation of AES-128 in C6443 processor and Spartan 3. This ordinary implementation can also have same manifestation on competent either considering high speeds or low memory base implementation. Table 1 shows that there is no change in AES throughput. However, to maintain the throughput, additional area and memory is required in FPGA. In case of DSP, there is no change in memory.

Table 1. Result of implementations on DSP and FPGA.

	DSP(594MHz)		FPGA(100MHz)	
	AES	AES with *DMCT*	AES	AES with *DMCT*
Complete Encryption Cycles	2047	2047	42	42
Throughput (Mbps)	35.4	35.4	304	304
Extra Area/Block RAM	-	1(word)	431/14	760/30

7 Conclusion

We have introduced DMCT in naive AES encryption by using dynamic MDS matrices based on m-bit additional secret key. We enhance *m* bit security level to default AES with respect to exhaustive key search. We briefly discuss the effectiveness of current naive and enhanced methods of cryptanalysis against AES by introducing DMCT and present possible security enrichment for different methods of attack. We also achieve same level of processing cost as default AES encryption algorithm using current processors. Our Hardware implementation indicates that AES become more secure with no effect on throughput. Our results also show that multiple layer security still stands as a noble candidate to enhance security of ciphers against developing cryptanalytic techniques and this area has research potential to a great extent.

References

1. Daemen, J., Rijmen, V.: AES Proposal: Rijndael, http://csrc.nist.gov/encryption/aes/rijndael/
2. FIPS-197: Advanced Encryption Standard, November 2001, available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
3. Daemen, J., Knudsen, L.R. Rijmen, V.: The Block Cipher Square, In Biham, E. (eds.), FSE'97, LNCS, vol. 1267, pp. 149—165, Springer Verlag, (1997).
4. Schneier, B; Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N., Kohno, T., Stay, M.,: The Twofish team's final comments on AES Selection, In Tadayshi, K., Mike S. (May 15, 2000).
5. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved cryptanalysis of Rijndael, FSE, LNCS'78, pp. 213—230, Springer Verlag (2000).
6. Biham, E., Dunkelman, O., Keller, N.: Related-key impossible differential attacks on AES-192, In CT-RSA'06, LNCS, vol. 3860, pp. 2--31, Springer Verlag (2006).
7. Murtaza, G., Ikram, N.: New Methods of Generating MDS Matrices. In: Proceedings of ICWC 2008, pp 129-133, ISBN: 978-983-44069, Kuala Lumpur, (2008).
8. Merkle, R.C.: Fast Software Encryption Functions, Proceedings of Crypto'90, pp.476--501, (1991).
9. Biryukov, A., Khovratovich, D., Nikolic, I.: Distinguisher and related-key attack on the full AES-256. In CRYPTO'09, LNCS. Springer Verlag (2009).
10. Biham, E.: On Masui's Linear Cryptanalysis, EUROCRYPT, (1994).

11. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis, In Davie, D.W. (eds.), Advances in Cryptology, Proc. Eurocrypt'90, LNCS 473, pp. 389—404, Springer Verlag (1991).
12. Biryukov, A., Khovratovich, D.: Related-key Cryptanalysis of the Full AES-192 and AES-256, IACR ePrint report 2009/317, 2009. Available online at http://eprint.iacr.org/2009/317.
13. Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., Shamir, A.: Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds. Cryptology ePrint Archive, Report 2009/374, 2009. Available online at http://eprint.iacr.org/2009/374.
14. Henri, G., Thomas, P.: Super-Sbox Cryptanalysis: Improved Attacks for AES-like permutations. Cryptology ePrint Archive, Report 2009/531, 2009. Available online at http://eprint.iacr.org/2009/531.
15. Courtois, N., Pieprzyk, J.: Cryptanalysis of block ciphers with overdefined systems of equations, In Zheng, Y., (eds.), ASIACRYPT'02, LNCS, vol. 2501, pp. 267—287, Springer (2002).
16. Kim, J., Hong, S., Preneel, B.: Related-key rectangle attacks on reduced AES-192 and AES-256. In FSE 2007, LNCS, vol. 4593, pp. 225-241, Springer Verlag (2007).
17. Yong Zhuang, W., YuPu, H.: New Related-Key rectangle attacks on reduced AES-192 and AES-256, Science in China Press, vol.52, n.4, pp. 617-626, Springer Verlag (2009).
18. NIST SP 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, March 2007
19. Biham, E., Keller, N.: Cryptanalysis of Reduced Variants of Rijndael, Available at http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html
20. Borst, J.: The bolck cipher: Grand Cru, Available at https://www.cosic.esat.kuleuven.be/nessie/workshop/submissions/grandcru.zip
21. NESSIE (New European Schemes for Signatures, Integrity and Encryption) was a European research project funded from 2000–2003 to identify secure cryptographic primitives. https://www.cosic.esat.kuleuven.be/nessie/
22. Daemen, J., Rijmen, V.: The Wide Trail Design Strategy. In Bahram Honary, editor, IMA Int. Conf., LNCS, vol. 2260, pp 222-238. Springer Verlag (2001).
23. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer Verlag (2002).
24. Rimoldi, A.: PhD Thesis: On algebraic and statistical properties of AES-like ciphers. PhD thesis, University of Trento (2009) Available at http://eprints-phd.biblio.unitn.it/151/
25. Barken, E., Biham, E.: In how many ways can you write Rijndael? LNCS, vol. 2501, pp 160-175, ASIACRYPT (2002).
26. Ferguson, N., Schroeppel, R., Whitinf, D.: A simple algebraic representation of Rijndael, LNCS, vol. 2259, pp. 103–111, Proc. of SAC (2001).
27. Murphy, S., Robshaw, M.: Essential algebraic structure within the AES, Proc. of CRYPTO 2002, LNCS, vol. 2442, pp. 1–16, Springer Verlag (2002).
28. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved Cryptanalysis of Rijndael, proceedings of FSE 2000, LNCS, vol. 1978, pp. 213–230, Springer Verlag (2001).
29. TMS320DM6443 Digital Media System-on-chip SPRS282E- December 2005 –Revised March 2007.
30. TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide, Literature Number: SPRU732H October 2008.
31. Using Block RAM in Spartan – 3 Generation FPGAs, Xilinx Application Notes XAPP463, March 2005.

32. Murtaza, G., Ikram, N.: Direct Exponent and Scalar Multiplication Classes of an MDS Matrix. IACR ePrint report 2011/151, 2011. Available online at http://eprint.iacr.org/2011/151.