ISSN 2320 2610

Volume 3, No.3, May - June 2014

International Journal of Multidisciplinary in Cryptology and Information Security

Available Online at http://warse.org/pdfs/2014/ijmcis01332014.pdf

FPGA Based Crypto Processor through Triangular Modulo Arithmetic Technique (TMAT)

Rajdeep Chakraborty[1], Santanu Basak[1], JK Mandal[2]

[1]Department of CSE, Netaji Subhash Engineering College, Garia, Kolkata - 700152, West Bengal, India, rajdeep.chak@gmail.com, shan.basak@gmail.com

[2]Department of CSE, FETM, University of Kalyani, Kalyani, West Bengal, India, jkm.cse@gmail.com

ABSTRACT

In this paper a block cipher based new cryptosystem has been proposed, where the encryption is done through Triangular Modulo Arithmetic Technique (TMAT), which consists of three phases. The original message is considered as a stream of bits. In Phase 1, bit stream is divided into a number of equal size blocks. Then the Triangular algorithm is performed on odd blocks, where even blocks are remain unchanged and together form a bit stream. In Phase 2, MAT is performed on that bit stream, which is divided into a number of blocks, each containing n bits, n is 2^k, k is 1, 2, 3 and so on. Then modulo addition is performed between first and second block and the content of the second block is replaced by the result, where the first block is remain unchanged. This is continuing till the last block is changed and also for block size n is 2, 4, 8, and so on. In case of modulo addition the carry out of the MSB is discarded. In Phase 3, the Triangular algorithm is performed on even blocks but odd blocks are remain unchanged and together form output stream. In case of decryption, reverse algorithm is used, where modulo subtraction technique is performed instead of performing modulo arithmetic technique.

Key words : Block cipher, Cryptosystem, FPGA, MAT, TMAT, Triangular, VHDL.

1. INTRODUCTION

The Lack of security may exist when a volume of data is transferred from its source to the destination if no measure is taken for its security. For one reason or the other, most of the data being transmitted must be kept secret from others. A very important reason to encode data or messages is to keep them secret. From e-mail to cellular communication, from secured web access to digital cash, cryptography [4]-[6] is an essential part of today's information systems. It can prevent fraud in electronic commerce and assure the validity of financial transactions. It can prove one's identity and protect one's anonymity. These electronic commerce schemes may fall fraud through forgery, misrepresentation, denial of service (DOS) [4],[5] and cheating if we do not add security to these systems. In fact, computerization makes the risks even greater by allowing attacks that are impossible in non automated systems. Only strong cryptography can protect against these attacks.

The Section 1 of this paper deals with the proposed scheme. A concept of key-generation is given in Section 2. Results and comparisons are illustrated in Section 3. Conclusions are drawn in Section 4, acknowledgements are shown in next section and the last Section lists the references.

1.1. The Triangular Modulo Arithmetic Technique (TMAT)

The proposed scheme has been developed from two algorithm, MAT [1],[2] and Triangular algorithm. In the proposed scheme the source file is taken as binary streams. The input stream size and input key size have been considered 512 bits and 128 bits for the implementation, though the scheme can be implemented for larger input stream sizes also. The proposed algorithm is consisted of three phase where the Triangular algorithm is performed in Phase 1 and in Phase 3 and MAT is performed in Phase 2. The key generation will be discussed in section 2. Figure 1 shows the proposed scheme (TMAT).

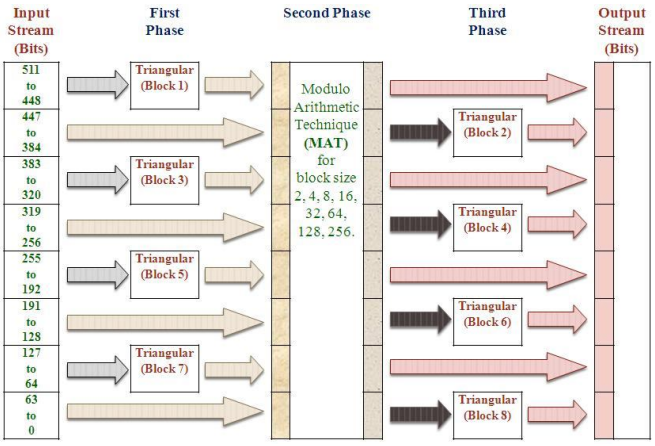

Figure 1: TMAT Encryption Procedure

1.2. The Technique

The operations of TMAT algorithm is performed in three phases, where the first and third phase are triangular technique and second phase is MAT.

Phase 1 : In Phase 1, 512 bits input stream, S, is broken into 8 numbers of equal size blocks, each containing 64 bits and the Triangular algorithm is implemented on block number 1, 3, 5, 7 (i.e. odd blocks) where the rest of the blocks are (even

blocks) remain unchanged. Consider that the source block size, t (here 64). In the Triangular Encryption technique an intermediate block of size (t-1) is generated from the source block, by applying the exclusive NOR (XNOR) operation between each two consecutive bits. In the next step a new block of size (t-2) is generated from previous block of size (t-1) and this process goes on until the generation of block size 1. All these blocks under consideration together form an equilateral triangle-like shape. After the formation of such a triangular shape, putting together either the MBSs or the LSBs of all the blocks under consideration in either sequence. The target block is formed in this regard, the key takes a vital role because only by knowing this key the receiver of the message can understand on how the target block is chosen from the triangular shape. The encrypted stream of bits is generated by putting together all the target blocks. Then the both changed and unchanged blocks are concatenated and formed bit stream of 512 bits, say S^1. Figure 2 shows the Triangular formation technique.

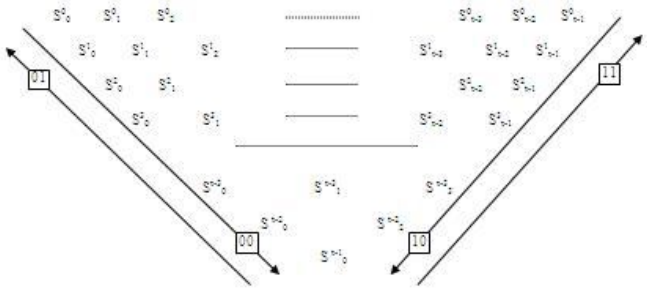

Figure 2: Triangular Formation

Phase 2 : . In this phase, Modulo Arithmetic Technique (MAT) is performed on that stream of 512 bits. This is performed in 8 rounds. The input stream, S^1, is broken into a number of blocks, each containing n bits where $n=2^k$, k=1,2,3,.....,8, k denotes the round number. So, $S^1 = B_1B_2B_3.....B_m$, where m=512/n. Starting from the MSB, the blocks are paired as $(B_1, B_2),(B_3, B_4),(B_5, B_6), \ldots,(B_{m-1}, B_m)$. The addition is performed between two blocks of each pair and the content of the second block of each pair is replaced by the result. This will be going on until the content of the last block B_m is replaced by the result. The process is repeated and each time the block size increases till n=256. So a new encrypted stream, S^2 is generated after MAT is performed with block size 256.

Round 1 : In this round of encryption, block size is taken as 2, it means k=1 and addition is performed between each pair of blocks and second block of each pair is replaced by the result. This round is repeated for a finite number of times and the number of iterations will form a part of the session key as discussed in Section II.

Round 2 : Same operation is performed as in Round 1 with block size 4 (i.e. k=2).

In this fashion several rounds are completed till we reach Round 8 (i.e. k=8) where the block size is 256 and we get the encrypted bit-stream. So after the completion of Round 8 another encrypted bit stream is generated, say, S^3 .

Phase 3 : In the phase 3, the binary stream, S^3 is divided into 8 equal size of blocks and Triangular algorithm is implemented on block no 2, 4, 6, 8 (i.e. even blocks) and rest of the blocks are (odd blocks) remain unchanged. After that changed and unchanged together are concatenated and produces final output stream, S^{en}.

During decryption, the reverse operation is performed. In the Phase 1, Triangular algorithm is performed on block no 2, 4, 6, 8 (i.c even blocks) and odd blocks are remain unchanged and then in the Phase 2, modulo subtraction, is performed instead of performing modulo addition where block size starts from 256 and end with 2 ($n=2^k$, k=8, 7, 6 ,..... 3, 2, 1). In the Phase 3, Triangular algorithm is performed on block no 1, 3, 5, 7 (i.e. odd blocks) and even blocks are remain unchanged. In case of Triangular decryption. selection of result is quite different than that of encryption technique. If selection is 00 or 11 during encryption, it is same for decryption technique but if it is 01 or 10, interchange is done between them.

1.3. The Modulo Addition

An alternative method for modulo addition has been proposed here to make the calculations simple. The need for computation of decimal equivalents of the blocks is avoided here since we will get large decimal integer values for large binary blocks. In the proposed method the carry out of the MSB has been discarded after the addition of two blocks of each pair. For example, if we add 1101 and 1000 we get 10101. In terms of decimal values, 13+8=21. Since the modulus of addition is 16 (2^4) in this case, the result of addition should be 5 (21-16=5). Discarding the carry from 10101 is equivalent to subtracting 10000 (i.e. 16 in decimal). So the result will be 0101, which is equivalent to 5 in decimal. The same is applicable for any block size.

1.4. Example of Encryption

Although the proposed scheme is applicable for a 512-bit input stream but here 16 bit input stream has been considered, to make the process simple for understanding.

consider a stream of 16 bits, say S = 1101110011010010.

Phase 1 : In this phase, the input stream is divided into four blocks with 4 bits each. The Triangular technique is performed on odd blocks (i.e. Block 1 and Block 3) and even blocks (i.e. Block 2 and Block 4) are remain unchanged.

Input

Block 1				Block 2				Block 3				Block 4			
1	1	0	1	1	1	0	0	1	1	0	1	0	0	1	0

Triangular implementation on Block 1 and Block 3.

Block 1	Block 3
1 1 0 1	1 1 0 1
1 0 0	1 0 0
0 1	0 1
0	0

Output

Block 1				Block 2				Block 3				Block 4			
1	1	0	0	1	1	0	0	0	1	0	1	0	0	1	0

Consider that the selection key for Block 1 is 00 and Block 3 is 11. Then output from Block 1 is 1100 and from Block 3 is 0101. Block 2 and Block 4 are remain unchanged. So after Phase 1 output, S^1 = 1100110001010010. This output is the input for Phase 2.

Phase 2 : In this phase, MAT Is performed for block size 2, 4 and 8 (as input is taken 16 bits, so maximum block size is to be 8). So total number of rounds is 3. Each round is performed only once to make the process simple for understanding.

Round 1 : Block size = 2, number of blocks = 8.

Input

B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
11	00	11	00	01	01	00	10

(B1, B2) Modulo Addition, B2 is replaced by result. Same operation is performed for (B3, B4), (B5, B6) and (B7, B8).

Output

B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
11	11	11	11	01	10	00	10

Round 2 : Block size = 4, number of blocks = 4.

Input

B_1	B_2	B_3	B_4
1111	1111	0110	0010

Output

B_1	B_2	B_3	B_4
1111	1110	0110	1000

Round 3 : Block size = 8, number of blocks = 2.

Input

B_1	B_2
11111110	01101000

Output

B_1	B_2
11111110	01100110

So, after Phase 2, generated output, S^2 = 1111111001100110, which is input for phase 3.

Phase 3 : The triangular technique is performed on even blocks, i.e. Block 2 and Block 4 where odd blocks (Blocks 1 and Blocks 2) are remain unchanged.

Input

Block 1				Block 2				Block 3				Block 4			
1	1	1	1	1	1	1	0	0	1	1	0	0	1	1	0

Triangular implementation on Block 2 and Block 4.

Block 2	Block 4
1 1 1 0	0 1 1 0
1 1 0	0 1 0
1 0	0 0
0	1

Consider that the selection key for Block 2 is 01 and Block 4 is 10. Then output from Block 2 is 1101 and from Block 4 is 0010. Block 1 and Block 3 are remain unchanged. So after phase 3 output, S^{en} = 1111011101100001. This is the final encrypted output.

1.5. Example of Decryption

Previous output stream, which was generated during encryption technique has been considered as input bit stream for decryption technique.

Phase 1 : In this phase, the input stream is divided into four blocks with 4 bits each. The Triangular technique is performed on even blocks (i.e. Block 2 and Block 4) and odd blocks (i.e. Block 1 and Block 3) are remain unchanged.

Input

Block 1				Block 2				Block 3				Block 4			
1	1	1	1	0	1	1	1	0	1	1	0	0	0	0	1

Triangular implementation on block 2 and block 4.

Block 2	Block 4
0 1 1 1	0 0 0 1
0 1 1	1 1 0
0 1	1 0
0	0

Output

Block 1				Block 2				Block 3				Block 4			
1	1	1	1	1	1	1	0	0	1	1	0	0	1	1	0

As the selection keys were considered during encryption technique, 01 for Block 2 and 10 for Block 4. Then output from Block 2 is 1010 and from Block 4 is 0011. Block 1 and Block 3 remain unchanged. So after Phase 1 output, S_1^1 = 1111111001100110, which is the input of Phase 2.

Phase 2 : In this phase, MAT is performed but instead of using modulo addition, modulo subtraction is used. Block size is used in reverse order (i.e. 8, 7, 6, …, 1).

Round 1 : Block size = 8, number of blocks = 2

Input

B_1	B_2
11111110	01100110

Output

B_1	B_2
11111110	01101000

Round 2 : Block size = 4, number of blocks = 4
Input

B_1	B_2	B_3	B_4
1111	1110	0110	1000

Output

B_1	B_2	B_3	B_4
1111	1111	0110	0010

Round 3 : Block size = 2, number of blocks = 8
Input

B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
11	11	11	11	01	10	00	10

Output

B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
11	00	11	00	01	01	00	10

So after phase 2 output, S_1^2 = 1100110001010010, which is the input of Phase 3.
Phase 3 : In this phase, the Triangular technique is applied on odd blocks (i.e. Block 1 and Block 3) and even blocks (i.e. Block 2 and Block 4) are remain unchanged.
Input

Block 1				Block 2				Block 3				Block 4			
1	1	0	0	1	1	0	0	0	1	0	1	0	0	1	0

Triangular implementation on block 1 and block 3.

Block 1				Block 3			
1	1	0	0	0	1	0	1
	1	0	1		0	0	0
		0	0			1	1
			1				1

Output

Block 1				Block 2				Block 3				Block 4			
1	1	0	1	1	1	0	0	1	1	0	1	0	0	1	0

As the selection keys were considered during encryption technique, 00 for Block 1 and 11 for Block 3. Then output from Block 1 is 1101 and from Block 3 is 1101. Block 2 and Block 4 remain unchanged. So after Phase 3, final decrypted output, S^{de} = 1101110011010010. So, S^{de} = S.

2. KEY GENERATION

In the proposed scheme, eight blocks (Block 1, Block 2, Block 3, Block 4, Block 5, Block 6, Block 7 and Block 8) have been considered for Triangular and eight rounds (for block size 2, 4, 8, 16, 32, 64, 128 and 256) have been considered for MAT. In case of the Triangular technique as 2 bits are required for selection from each blocks, so for eight blocks, 16 bits are required from 128 bits key. So 16 bits are selected from LSB of 128 bits key for eight rounds of the Triangular technique. Then 16 bits are equally divided into eight blocks (2 bits in each). From MSB of that 8 blocks are used for Block 1 to Block 8. Each round of MAT is repeated for a finite number of times and the number of iterations is a part of the 112 bits from MSB of 128 bits key. Then that 112 bits are equally divided and formed 8 blocks (14 bits in each). Each block of that 8 blocks are used as number of iterations of a round (from Round 1 to Round 8). In case of decryption technique, same key is used in the same way. Example in Section 2.1 illustrates the key generation process.

2.1. Example of Key Generation

The key generation procedure will be illustrated in this section. As proposed scheme consists of three phase and same procedure is used for Phase 1 and Phase 3, the example will be shown in two steps. Table 1 shows target block selection using selection key and Table 2 shows target block selection for the example. Consider a particular session, where 128 bits input key stream is :
00000000010010100000001010101000000100001011011010110110001100101011011001101110111110111011101100001011001110110010010111000.
So, 16 bits from LSB are used for the Triangular algorithm and remaining 112 bits are used for MAT.

2.1.1. Generation for Phase 1 and Phase 3

16 bits key string is : 1100100101111000.

Table 1: Target Block Selection using Selection Key

Sl. No.	Selection Key (2 bits)	Target Block Selection
1.	00	Taking all the MSBs starting from the source block till the last block generated
3.	01	Taking all the MSBs starting from the last block generated till the source block
1.	10	Taking all the LSBs starting from the source block till the last block generated
3.	11	Taking all the LSBs starting from the last block generated till the source block

Table 2: Key Distribution for the Triangular Technique

Block No.	Phase	Target Block Selection		
		Selection key (2 bits)	Used for Encryption	Used for Decryption
1	1	11	11	11
2	3	00	00	00
3	1	10	10	01
4	3	01	01	10
5	1	01	01	10
6	3	11	11	11
7	1	10	10	01
8	3	00	00	00

2.1.2. Generation for Phase 2

In this phase of key generation ,112 bits key from MSB is used in MAT for block sizes 2, 4, 8, 16, 32, 64, 128, and 256 bits, respectively. Table 3 shows the key distribution of Phase 2. 112 bits key is : 00000000010010100000001010101000000100000101101101011011000110010101101100110111011111011101101100001011001110.

Table 3: Key Generation for MAT

Round No.	Block Size	No. of Iteration	
		Binary	Decimal
1	2	00000000010010	18
2	4	10000000101010	8234
3	8	10000001000010	8258
4	16	11011010101101	13997
5	32	10001100101011	9003
6	64	01100110111011	6587
7	128	11101110111011	15291
8	256	00001011001110	718

3. RESULTS AND COMPARISONS

The Chi-Square test has been performed for testing the homogeneity of the source and the encrypted file. Table 3 and Figure 3 shows the source file name, size and the corresponding Chi-Square values (using TMAT, Triple DES, RSA) for ten different files. Barring some exceptions we see that the Chi -Square value increases with the increase in file size. Further, the high values prove that Chi-Square is highly significant at 1% level of significance.

Table 4: Test for Homogeneity using Chi-Square Method

Sl. No.	File Name	File Size (in KB)	Chi Square Value		
			TMAT	TDES	RSA
1.	img004b.jpeg	61.7	6154	6769	5997
2.	img103.jpeg	66.3	15885	17617	16375
3.	ReadMe-en.txt	71.5	1068376	1059938	1014981
4.	img089.jpeg	85.2	9026	9497	8835
5.	LICENSE.txt	89.8	1272665	1186808	1130494
6.	img046.jpeg	105	35642	39917	36896
7.	thirdpartylicenseadme.txt	173	2604122	2434168	2447989
8.	genesis-en.txt	202	3575115	3526358	3343875
9.	HelpCtr.exe	777	10632711	12398524	5166731
10.	Explorer.exe	1035	9834705	10488223	3504238

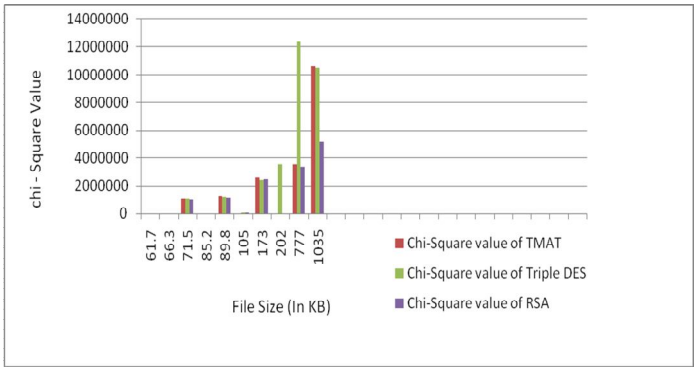

Figure 3: Graph showing Chi-Square Values for TMAT, Triple DES and RSA

Hence the source and the corresponding encrypted files are considered to be heterogeneous. Table 5 shows the encryption time of TMAT, TDES, RSA and Figure 4 shows the graphical view of encryption time. Table 6 shows decryption time of three techniques and Figure 5 shows graphical view of decryption time.

Table 5: Indicates Encryption Time for TMAT, Triple DES and RSA

Sl. No.	File Size (in KB)	Encryption Time (in Sec)		
		TMAT	TDES	RSA
1.	61.7	0.375	6.222	0.17
2.	66.3	0.435	6.746	0.17
3.	71.5	0.33	7.185	0.18
4.	85.2	0.595	8.608	0.21
5.	89.8	0.605	9.44	0.23
6.	105	0.675	10.625	0.21
7.	173	1.406	17.583	0.29
8.	202	1.59	20.417	0.35
9.	777	4.721	75.884	1.11
10.	1035	6.101	101.756	1.48

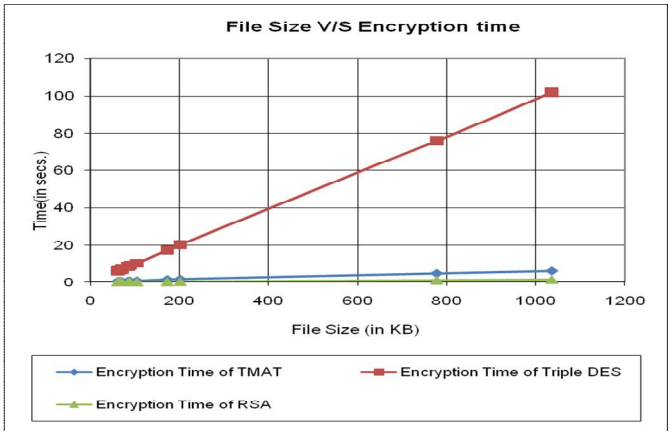

Figure 4: Graph showing Encryption Time for TMAT, Triple DES and RSA

Table 6: Indicates Decryption Time for TMAT, Triple DES and RSA

Sl. No.	File Size (in KB)	Decryption Time (Sec)		
		TMAT	TDES	RSA
1.	61.7	0.42	6.583	1.14
2.	66.3	0.595	7.75	1.17
3.	71.5	0.358	7.628	1.28
4.	85.2	0.603	9.105	1.82
5.	89.8	0.484	9.582	1.67
6.	105	1.064	11.359	1.87
7.	173	1.554	18.588	3.09
8.	202	1.188	21.582	3.62
9.	777	4.56	80.1	13.09
10.	1035	5.475	108.295	17.50

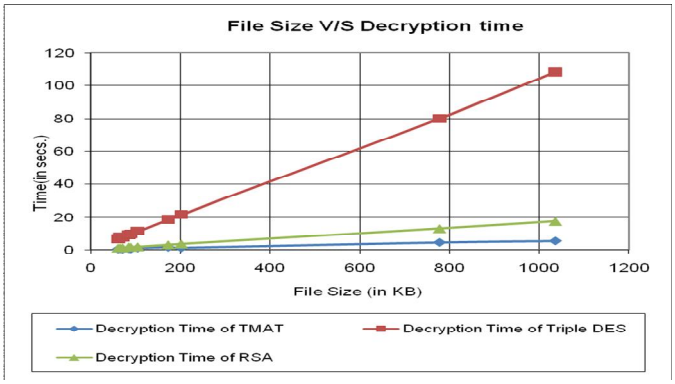

Figure 5: Graph showing Decryption Time for TMAT, Triple DES and RSA

It can be seen that the time taken to encrypt a file using TMAT is very little compared to Triple DES and little greater than that of RSA, but in case of decryption it is less than that of triple DES and also RSA. Another way to analyze the scheme is to analysis the avalanche ratio. Table 7 shows the avalanche ratio of TMAT, TDES, RSA and Figure 6 shows graphical representation of the avalanche ratio.

Table 7: Indicates Avalanche Ratio for TMAT, Triple DES and RSA

File Size (In KB)	Avalanche Ratio		
	TMAT	TDES	RSA
61.7	79.97	79.96	80.10
64.6	76.35	76.31	76.53
71.5	99.28	99.33	99.62
85.2	31.58	31.55	31.63
89.8	99.21	99.32	99.64
105	40.25	40.24	40.32
173	98.98	98.71	99.43
202	99.41	99.58	99.62
777	98.94	98.53	99.61
1035	95.97	95.74	96.71

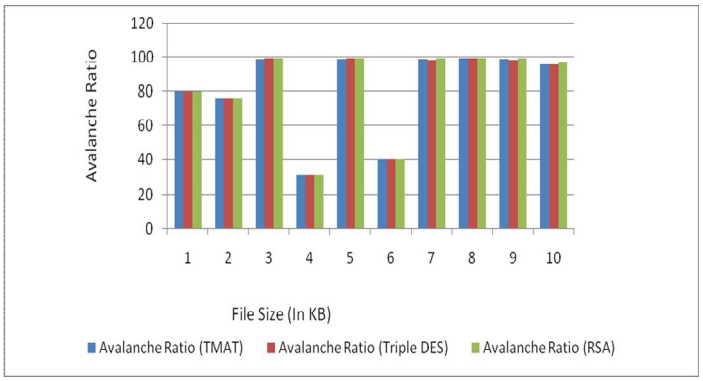

Figure 6: Graph showing Avalanche Ratio for TMAT, Triple DES and RSA

Figure 7 shows the RTL schematic diagram [7],[8] of encryption and decryption engine of 512 bits input and 128 bits key and 512 bits output. Figure 8 shows the encryption result and Figure 9 shows the decryption result. Table 8 shows the synthesis reports and Table 9 shows the Timing Summary. In the RTL schematic diagram, in_data pin is for input data (512 bits), in_key pin is for input key (128 bits), clk pin denotes system clock, enable pin is used to enable the chip (1 is for enabling), opp pin works for type of operations (o for encryption or 1 for decryption) and reset works to reset. On the other side out_data pin is for output data (512 bits) or results (encrypted or decrypted) and valid pin denotes for data validation (1 for valid and 0 for in valid).

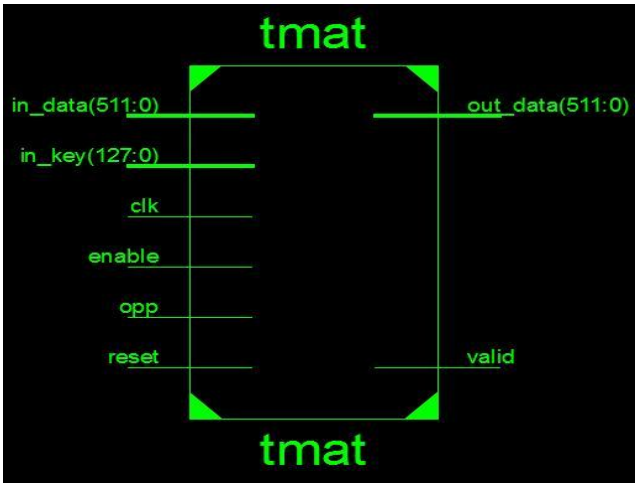

Figure 7: RTL Schematic Diagram of TMAT Encryption / Decryption

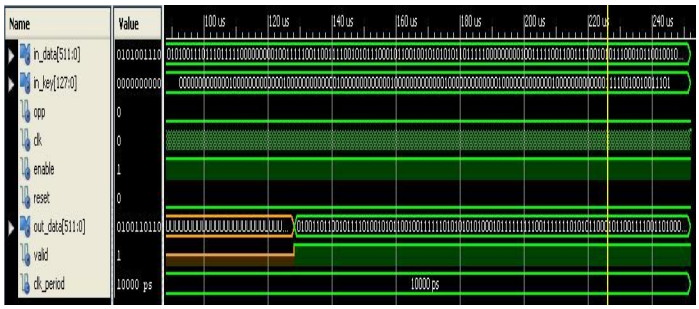

Figure 8: Output of TMAT Encryption

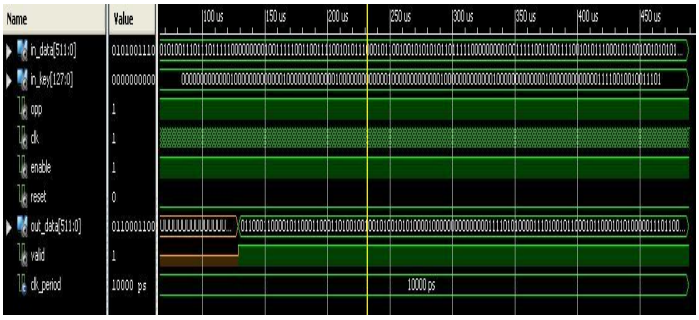

Figure 9: Output of TMAT Decryption

Table 8: HDL Synthesis Report for RSA and TMAT

Netlist Components	Number	
	RSA	**TMAT**
# Adders / Subtractors	3	64
14-bit adder	0	16
32-bit adder	0	48
34-bit adder	1	0
34-bit subtractor	2	0
# Registers	430	11366
1-bit register	403	11283
32-bit register	7	48
64-bit register	0	16
96-bit register	4	0
128-bit register	16	0
512-bit register	0	19
# Comparators	1	64
14-bit comparator equal	1	16
14-bit comparator not equal	0	16
32-bit comparator equal	0	16
32-bit comparator not equal	0	16
# Xors	192	48
1-bit xor	192	48

Table 9: Timing Summary for RSA and TMAT

Timing Constraint	Values	
	RSA	**TMAT**
Speed Grade	-4	-11
Minimum period	9.895ns	10.394ns
Maximum Frequency	101.06MHz	96.213MHz
Minimum input arrival time before clock	6.697 ns	10.315ns
Maximum output required time after clock	4.31ns	4.221ns

4. CONCLUSION

The proposed technique takes little time for encryption and also decryption though the block size is high. This technique takes the input as binary stream as a multiply of 512, so there is possibility of generation overhead but it is very little. The input binary stream further can be increased beyond 512, so the block size also can be increased beyond 256, which may enhance the security. The scheme can be implemented in an FPGA chip [9] using VHDL [7],[8], but high configured device is required. If input stream length decreases, required configuration will be decreased.

ACKNOWLEDGEMENT

Authors express their deep sense of gratitude to the Department of Computer Science & Engineering, Netaji Subhash Engineering college, West Bengal University of Technology. Authors also express their deep sense of gratitude to the Department of Computer Science & Engineering, FETM, University of Kalyani, Kalyani.

REFERENCES

1. Rajdeep Chakraborty, Debajyoti Guha and J. K. Mandal, **A Block Cipher Based Cryptosystem Through Forward Backward Overlapped Modulo Arithmetic Technique (FBOMAT)**, *International Journal of Engineering & Science Research Journal (IJESR)*, ISSN 2277 2685, Volume 2 – Issue 5 (May 2012) , Article number 7, pp-349 – 360. Email : rajdeep_chak@indiatimes.com, guha_debajyoti@yahoo.com, jkmandal@sancharnet.in.
2. Debajyoti Guha, Rajdeep Chakraborty, Abhirup Sinha, **A Block Cipher Based Cryptosystem through Modified Forward Backward Overlapped Modulo Arithmetic Technique (MFBOMAT)** , *IOSR Journal of Computer Engineering (IOSR-JCE)*, e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 1 (Jul. - Aug. 2013), PP 138-146.
3. Sushanta Kumar Sahu, Manoranjan Pradhan, **FPGA Implementation of RSA Encryption System**, *International Journal of Computer Applications* (0975 – 8887), Volume 19– No.9, April 2011.
4. W. Stallings. ***Cryptography and Network Security : Principles and Practices (Prentice Hall***, Upper Saddle River, New Jersey, USA, Third Edition, 2003).
5. Atul Kahate. ***Cryptography and Network Security*** (TMH, India,2nd Ed,2009).
6. Behroz Forouzan. ***Cryptography and Network Security*** (TMH, India, 4th Ed,2010).
7. Douglas L. Perry. ***VHDL : Programming by Example,*** McGraw-Hil , Fourth Edition.
8. J. Bhaskar. ***VHDL Primer***, Pearson Education, Third Edition, 2002.
9. Wayne Wolf. ***FPGA-Based System Design***, Pearson Education, India, 1st ED, 2009.
10. Mandal, J. K., Sinha, S., Chakraborty, R., **A Microprocessor-based BlockCipher through Overlapped Modulo Arithmetic Technique (OMAT)**, *Proceedings of 12th International Conference of IEEE on Advanced Computing and Communications - ADCOM-2004*, December 15-18, Ahmedabad, India, pp. 276 - 280, 2004.