Robust Modification and Implementation of Randomized Cryptographic Algorithm

Feroz Morab, Sadiya Thazeen, Mohamed Najmus Saqhib, Seema Morab

***Abstract*— We live in an age of internet where most part of the day is spent on-line and not to forget the recent past like WikiLeaks and other such instances where, unfortunately, the privacy in communication over electronic media has been compromised to big extent causing irreparable damages. This raises an alarm about the need for safety and security over the internet. One essential aspect of safe, secure and reliable communications is cryptography. While it plays an important role but cryptography as it is isn't sufficient to provide the ultimate security on both the ends. Hence we propose a concept where encryption and decryption is done using modified cryptography which uses randomly generated bits as the key to facilitate security and invulnerability over the channel.**

***Index Terms*— Cryptography, Decryption, Encryption, Modification of Randomized Cryptographic Algorithm (RCA).**

I. INTRODUCTION

Almost every internet application, be it mailing, online shopping, banking, financial transactions, military communications, demand strict privacy, confidentiality and security measures about every divulged detail and these are vital to the growth of electronic commerce and to the growth of the internet itself. Cryptography, defined as "the science of secret writing"[1], is the art of maintaining the privacy and integrity of the information by transforming it into some other form which cannot be deciphered by any third party. The encryption is done at one end and the decryption i.e. deciphering the ciphered information is done at the receiving end. The two basic methods of performing cryptography are symmetric and asymmetric cryptographic techniques. We focus on the symmetric one. The conventional technique namely Symmetric-key cryptography refers to encryption method in which both the sender and receiver share the same key.

The establishment of a shared secret key between communication parties has always been a difficult problem because the task needs a secure confidential channel and often such a channel requires physical delivery of keys by a special courier.

With a Symmetric Cryptosystem, it is necessary to transfer a secret key to both communication parties before secure communication can begin. An important edge of this new algorithm[2] is that the keys are generated automatically once the data bits come to the user for encryption. And every time the keys are generated, it is random and has no correspondence with the previously generated keys. The main objective of this paper is to provide secure exchange of secret keys. The novelty of this technical work is to use a variable reference bit size that reduces the number of operations thereby making efficient resource utilization.

II. LITERATURE SURVEY

A. Symmetric Key cryptography

Symmetric key encryption is also known as shared-key, single-key, secret-key, and private-key or one-key encryption. In this type of message encryption, both sender and receiver share the same key which is used to both encrypt and decrypt messages. Sender and receiver only have to specify the shared key in the beginning and then they can begin to encrypt and decrypt messages between them using that key. Examples include AES (Advanced Encryption Standard) and Triple DES (Data Encryption Standard)[3].

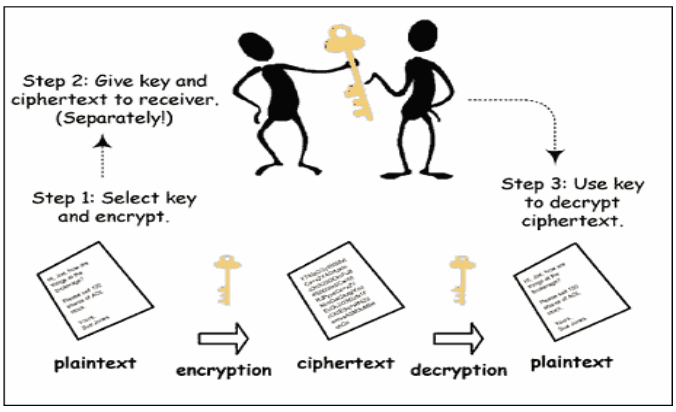


Fig. 1: Symmetric key cryptography

B. Randomized cryptographic algorithm

In this algorithm, the keys are generated randomly. Randomized cryptographic algorithm (RCA) uses a dibit as the reference bits for encrypting the data of different sizes. However for large data sizes, this procedure takes more operations for the generation of key[4].

III. PROPOSED METHODOLOGY

The randomized cryptographic algorithm involves two major phases namely:

ISSN: 2278 – 1323

(1) Parity Checking level where data bits are initially indexed and are checked bit-by-bit to ensure even parity. The even indexed data bits are used as the key for obtaining the pseudo-encrypted odd indexed data bits and these pseudo-encrypted odd indexed data bits act as the key for obtaining the pseudo-encrypted even indexed data bits. The partially-encrypted data bits are then obtained.

(2) Reference-bit checking level which involves the comparison of the input data bits with the obtained reference bits. The reference [1] demonstrates the use of a debit as reference for all the data sizes. In such cases, an n-bit data input will require '(n/2)-1' comparison operations, where n is a multiple of 2. The comparison operations can become very cumbersome for even larger data bits which will further result in more number of resources.

In the proposed modified cryptographic algorithm, n-bit data input requires '(n/k)-I' comparison operations where k is the reference-bit size. For a 256-bit data input and k=8(found using
Algorithm 2), only 31comparison operations are required as opposed to 127 in the case of the existing algorithm.

Hence considerable savings in terms of resource utilization leading to less silicon area can be achieved. Algorithm 2 details out the different steps involved in encrypting the data bits using modified RCA. Algorithm 2 shows the steps involved in optimum factor calculation for both encryption and decryption.

ALGORITHM 1: Reference bit calculation

Step 1: Obtain all the factors of 'n'.

Step 2: Count the number of factors.

- If (count =2)

Reference bit size =Default value, according to input data size.

- Else

Reference bit size = Median of the factors.

Step 3: End.

ALGORITHM 2: Encryption algorithm

Step 1: Index the data bits as 1,2,3 . . . n-l, n.

Step 2: Decimate the inputs to form even and odd indexed terms:

- Even: 2,4,6, . . . n
- Odd: l.3.5 . . . . n-1.

Step 3: Compute the pseudo-encrypted odd indexed data bits 1',3',5',…,n-1' using the even indexed data bits 2,4,6,….,n as the key.

Step 4: Compute the pseudo-encrypted even indexed data bits 2',4',6',….,n-l' using the pseudo-encrypted odd indexed data bits 1',3',5', ... n-1' as the key.

Step 5: Obtain the final partially encrypted data bits: 1',2',3',……n-1',n'.

Step 6: Compute the optimum factor of n. The optimum factor decides the size 'k' of the reference bits.

Step 7: Select the reference bits for comparison from the partially encrypted data bits.

Step 8: Divide the remaining partially encrypted data bit of size 'n-k' into 'k' number of equal sequences.
Step 9: for i = l; i<=k; i++
begin
Compare (reference bits, partially encrypted data sequence)
end

Step 10: Compute the final encrypted data.

Step 11: End.

The decryption algorithm follows a reverse procedure as that of the encryption algorithm and it begins with the factor calculation for finding the size of the reference bits. The reference bits are then selected for comparison and the partially decrypted data bits are found. The pseudo-decrypted even and odd indexed data bits are found respectively to decipher the original data.

IV. RANDOMIZED CRYPTOGRAPHIC ALGORITHM MODIFICATION

CASE STUDY

The new cryptographic algorithm can be understood easily by the following case study: Let the 8-bit data be 10 11 01 10.

Encryption algorithm:

Step 1: The 8 bits are indexed at the very outset for easy reference (Table I).

Step 2: The level 1 of encryption (parity checking level) starts here. The fundamental aspect of this level is that the key for encryption is stored in the data bits pattern itself and this key varies according to the varying data bits. The encryption is done in bit-by-bit operation. For encrypting data bits having index number 1 3 5 7, the corresponding key is data bits

having index number 2 4 6 8. The two nibbles are then checked bit-by-bit so that the final result has even parity with data bits 1 3 5 7. This final result is the pseudo-encrypted bit pattern 1` 3` 5` 7` (Table II). The pseudo-encrypted bit pattern 1` 3` 5` 7` becomes the key for the bit pattern 2 4 6 8. In the same manner, even parity is checked with the key and we get the pseudo-encrypted bit pattern 2` 4` 6` 8` (Table III). Let us illustrate this with one example. The same parity checking is done for the data bits having index number 2 4 6 8. The key in this case is the pseudo-encrypted bit pattern 1` 3` 5` 7`. Thus the complete pseudo-encrypted bit pattern having index number 1` 2` 3` ….8` becomes: 11 01 10 00.

Step 3: The level 2 (dibit checking level) starts here. In the same way as level 1, the 8-bits are indexed as 1` 2` 3`…8` for easy reference (Table IV).

Step 4: The bits having index numbers 1` and 8` are extracted from the 1-byte data. The dibit pair 1`-8` (consisting of the bits having index number 1` and 8`) becomes the reference dibit for the other three dibit pairs (viz. 2`-3`, 4`-5` and 6`-7`). Now a virtual table is prepared with four columns. The fields in the columns are NC (not complement), LSB-C (least Significant Bit-Complement), MSB-C (most significant bit-complement), FC (full complement).

In order to represent the four fields, we need at least 4 bits. Let the four fields, NC, LSB-C, MSB-C and FC be represented respectively as 00, 01, 10 and 11. Our objective in this step is to obtain the dibit pairs 2`-3`, 4`-5` and 6`-7` from the reference bit 1`-8` by any one of the four methods NC, LSB-C, MSB-C or FC; NC means the dibit pair is same as the reference dibit; LSB-C means the dibit pair is obtained from the reference dibit pair by complementing the least significant bit, MSB-C means the dibit pair is obtained from the reference dibit pair by complementing the most significant bit and FC means the dibit pair is obtained from the reference dibit pair by complementing both the bits.

The logic of the encryption then follows the following rule: Dibits 2`-3`, 4`-5` and 6`-7` are coded with their individual field codes and the bit 1` and 8` are appended both at the beginning and at the end respectively. Let us illustrate this with our current example. The dibit pair 1`-8` (viz 11) is extracted. This becomes the reference dibit for the other three dibit pairs 2`-3` (viz. 10), 4`-5` (viz. 11) and 6`-7` (viz. 01).

Now the logic goes like this: 10 (Dibit pair 2`-3`) is obtained from 11 (reference bit) by the method LSB-C; 11 (Dibit pair 4`-5`) is obtained from 11 (reference bit) by the method NC; 01 (Dibit pair 6`-7`) is obtained from 11 (reference bit) by the method MSB-C.

The virtual table looks like (Table V): NC corresponds to the code 00, LSB-C corresponds to the code 01, MSB-C corresponds to the code 10, FC corresponds to the code 11. Thus, the dibit encryption follows as: Dibit pair 2`-3` is coded as 01. Dibit pair 4`-5` is coded as 00. Dibit pair 6`-7` is coded as 10 and the bits 1` and 8` are appended at the beginning and at the end respectively. Thus the final encrypted code word becomes: 1 01 00 10 1.

Table I: Indexing of bits

INDEX	1	2	3	4	5	6	7	8
BITS	1	0	1	1	0	1	1	0

Table II: Encryption of bits 1 3 5 7

1	1	0	1	//data bits 1 3 5 7
0	1	1	0	//data bits 2 4 6 8 (key)
1	0	1	1	//pseudo-encrypted bits 1’ 3’ 5’ 7’

Table III: Encryption of bits 2 4 6 8

0	1	1	0	//data bits 2 4 6 8
1	0	1	1	//data bits 1’ 3’ 5’ 7’
1	1	0	1	//pseudo-encrypted bits 2’ 4’ 6’ 8’

Table IV: Indexing of bits

INDEX	1’	2’	3’	4’	5’	6’	7’	8’
BITS	1	1	0	1	1	0	1	1

Table V: Formation of virtual table

REFERENCE DIBIT 11	DIBIT PAIR	00 NC	01 LSB-C	10 MSB-C	11 FC
	2’ – 3’		△		
	4’ – 5’	△			
	6’ – 7’			△	

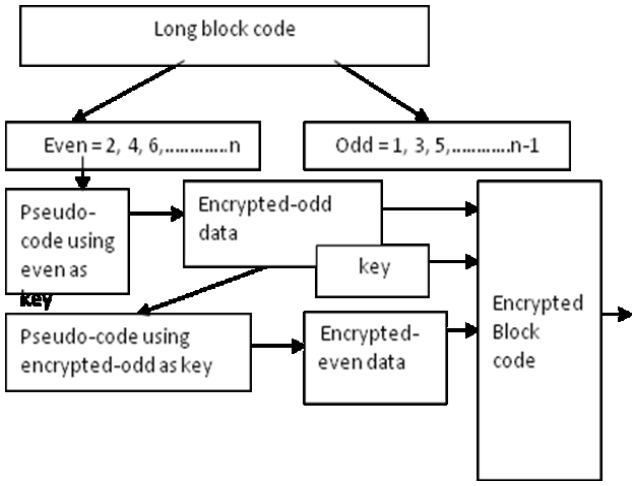


Fig. 2: Encryption process

ISSN: 2278 – 1323

Decryption algorithm:

Step 1: The 8 bits of encrypted data are indexed again at the very outset for easy reference (Table VI).

Step 2: The bits having index numbers 1 and 8 are extracted from the 1-byte data. The dibit pair 1-8 becomes the reference dibit for the other three dibit pairs (viz. 2-3, 4-5 and 6-7). The virtual table having the fields NC, MSB-C, LSB-C and FC is prepared again and the codes are entered accordingly in the appropriate places. The table looks like (Table VII).

For dibit 2-3 ▲ is placed in LSB-C field indicating the pseudo-decrypted dibit 2`-3` is obtained by complementing the least significant bit of the reference dibit. For dibit 4-5, ▲ is placed in NC field indicating the pseudo-decrypted dibit 4`-5` is the same as the reference dibit. And for dibit 6-7, ▲ is placed in MSB-C field indicating the pseudo-decrypted dibit 6`-7` is obtained by complementing both the MSB of the reference dibit. Thus, we get: 1 10 11 01 1, the pseudo-decrypted bit pattern.

Step 3: The pseudo-decrypted bit pattern goes for the next stage of decryption logic. For decrypting data bits having index number 2` 4` 6` 8`, the corresponding key is data bits having index number 1` 3` 5` 7`. The two nibbles are then checked bit-by-bit so that the final result has even parity with data bits 1` 3` 5` 7`.

This final result is the decrypted bit pattern 1 3 5 7. The decrypted bit pattern 1 3 5 7 becomes the key for the bit pattern 2` 4` 6` 8`. In the same manner, even parity is checked with the key and we get the decrypted bit pattern 2 4 6 8 (Table VIII). The same parity checking is done for the data bits having index number 1` 3` 5` 7`. The key in this case is the decrypted bit pattern 2 4 6 8 (Table IX). Thus the complete decrypted bit pattern having index number 1,2,3,…,8 becomes:10 11 01 10. This is exactly the same as the original plain text 10 11 01 10.

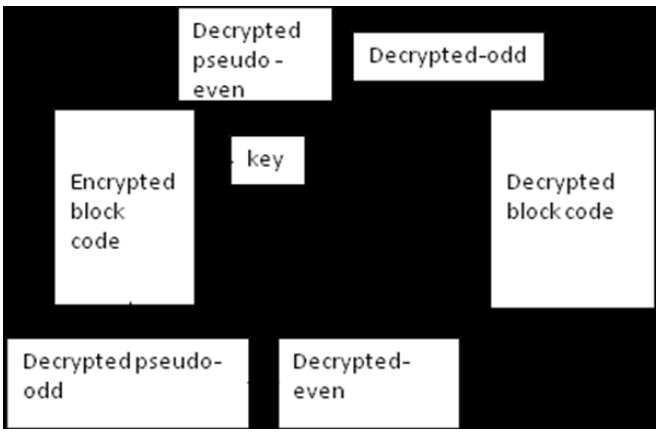


Fig. 3: Decryption process

Table VI: Indexing of bits

INDEX	1	2	3	4	5	6	7	8
BITS	1	0	1	0	0	1	0	1

Table VII: Formation of virtual table

REFERENCE DIBIT 11	00 NC	01 LSB-C	10 MSB-C	11 FC
		△		
	△			
			△	

Table VIII: Decryption of bits 2 4 6 8

1	1	0	1	//data bits 2’ 4’ 6’ 8’
1	0	1	1	//data bits 1’ 3’ 5’ 7’
0	1	1	0	//decrypted bits 2 4 6 8

Table IX: Decryption of bits 1 3 5 7

1	0	1	1	//data bits 1’ 3’ 5’ 7’
0	1	1	0	//decrypted data bits 2 4 6 8
1	1	0	1	//decrypted bits 1 3 5 7

V. RESULT AND ANALYSIS

The proposed modified cryptographic algorithm is implemented using Verilog HDL, the functionality of the algorithm is verified using ModelSim SE and synthesized using Xilinx ISE. The layout of the proposed design is generated using Cadence SoC encounter and the report summary is generated.

Fig. 4: Output of encryption and decryption process

The proposed cryptographic algorithm is synthesized using Synopsys Design Vision tool and the technology file used is UMC180 run. Area and power results are obtained for the existing RCA and the proposed cryptographic algorithm and the comparison is shown in Fig. 5 and Fig 6 respectively.

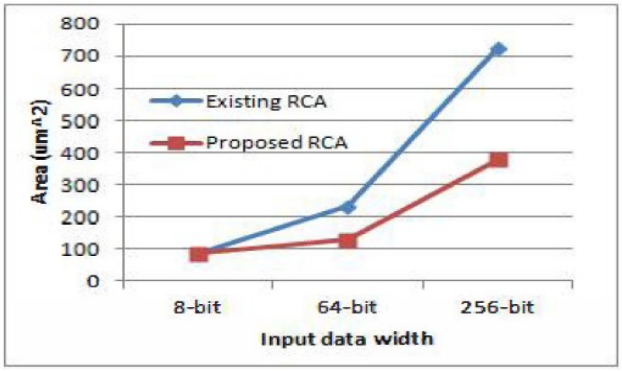


Fig. 5: Comparison of total area

Fig. 5 depicts that an increase in the input bit size increases the total area that is the sum of the combinational and non-combinational area. However the total area inferred by the proposed algorithm is comparatively lesser than that of the conventional algorithm. This nature can be attributed to the effort made in reducing the number of operations involved in the reference bit checking level of the encryption process[5]. This effect can be more profoundly seen for larger input data width.

The area due to logic cells in design is shown by the combinational (basic logic gates like ANDs, ORs, and the like) and the non-combinational (registers) factors. Reduction of the combinational area results in the reduction of the total number of logic gates involved in synthesizing the algorithm. Since a considerable amount of computational savings in the proposed algorithm has resulted in a reduced area as shown in Fig. 5, the total dynamic power subsequently reduces when compared to the conventional RCA[6]. For increased input data width this difference is more prominent as shown in Fig. 6.

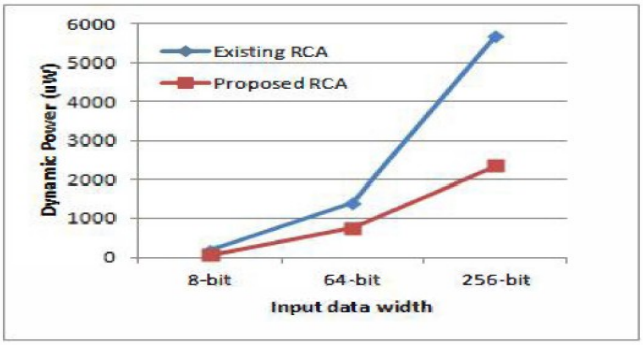


Fig. 6: Comparison of dynamic power

VI. Conclusion And Future Scope

The existing system can be more secured when it is sent in separate channel but its disadvantages are that it requires separate channel to communicate, where it is more expensive and it has to perform more operations to cipher the data. An important advantage that the proposed algorithm provides is the achievement of exchanging a secret key between remote communication parties with no need of a secure confidential channel. Also the key required for encryption and decryption are different making the algorithm comparatively tougher to decipher. The keys for encryption and decryption are not always the same data bits; they depend on the changing data bits[7]. If the plaintext message input to a basic cryptographic function has a random distribution, then the function provides a strong protection in hiding the plaintext information, even down to the level of an individual bit.

A padding scheme like our proposed algorithm has a random input value which adds randomness to the distribution of the result, that is, it makes the input to have a more random distribution. Thus, to sequentially combine the randomized padding scheme, we make use of the strong bit-security. The encrypted data can be sent through common channel. It takes less operational steps to cipher data.

The important application of this system is that it is used where encryption/decryption is especially important i.e., in wireless communications. This is because wireless circuits are easier to tap than their hard-wired counterparts. Encryption/decryption is a good idea when carrying out any kind of sensitive transaction, such as online financial transactions or the discussion of a company secret between different departments in the organization.

ISSN: 2278 – 1323

An area efficient robust and modified implementation of Randomized Cryptographic Algorithm is proposed and successfully implemented. Optimum factor calculation technique is introduced to change the reference bit size dynamically according to the input data width. The proposed algorithm is implemented using Verilog HDL, the functionality of the algorithm is verified using ModelSim SE and synthesized using Xilinx ISE. The layout of the proposed design is generated using Cadence SoC encounter. The implementation results show that this methodology provides obvious improvement in the area and dynamic power.

REFERENCES

[1] IEEE paper 2012: *Hardware Implementation of a Modified Randomized Cryptographic Algorithm*

[2] Soumyabrata Dev and Zia ul Haque Choudhury, *"A Randomized Cryptographic Algorithm and its Simulation in C and MATLAB with its Hardware Implementation in Veri log HDL"*. Anti-counterfeiting, Security and Identification in Communication, 2009, ASID 2009, 3rd International Conference, 20-22 Aug.2009.

[3] Mohammad Zakir Hossain Sarker and Md. Shafiul Parvez, *"A Cost Effective Symmetric Key Cryptographic Algorithm for Small Amount of Data"*. 9th International Multitopic Conference, IEEE INMIC 2005, 24-25 Dec.2005.

[4] Warwick Ford and Brian 0 'Higgins, *"Public-Key Cryptography and Open Systems Interconnection"*. IEEE Communication Magazine, July 1992.

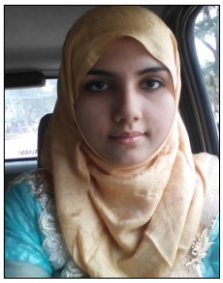
[5] Ashwini M. Deshpande, Mangesh S. Deshpande and Devendra N. Kayatanavar, *"FPGA Implementation of AES Encryption and Decryption"*. International Conference on Control, Automation, Communication and Energy Conservation, June 2009, pp 1-6.

[6] M. Alam, S. Ray, D. Mukhopadhyay, S. Ghosh, D. Roychowdhury and I. Sengupta: *"An Area Optimized Reconfigurable Encryptor for AES Rijndael"*. Proceeding of Design, Automation and Test in Europe, 2007, pp.1116-1121.

[7] An article on *"Introduction to Cryptography and Secure Communication"* by Edward J. Delp. Purdue University School of Electrical and Computer Engineering.

[8] William Stallings, 2006, *"Cryptography and Network Security"*. Principles and Practices, 4th edition. Pearson Education, pp.I-220.

Feroz Morab is currently pursuing M.Tech in DEC under VTU, Bangalore and is into research targeting to make communication channel more secured. He has publications in many International Journals to his credit. He is moving towards Ph.D and his areas of interest include Communication, Field Theory and Digital Signal Processing.



Sadiya Thazeen, pursuing M.Tech in DEC under VTU, Bangalore, has published many papers in International Journals and is moving towards her Ph.D. Her fields of interest are Digital Communication, Digital Signal Processing, Wireless Communication and Networking.

Mohamed Najmus Saqhib obtained Masters in Digital Electronics with high merit. He is currently working towards obtaining a Doctorate and has publications in many International Journals to his credit. His research areas are Communication and Networking.

Seema Morab is pursuing her Ph.D from AMITY UNIVERSITY, Noida, Delhi NCR and has published many papers in highly reputed International Journals.

