International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 35 • 2017

FPGA Implementation of OTPES Cryptographic Algorithm for Channel Security

B. Murali Krishna[a], Syed Shameem[b], P. Srikanth Reddy[c], V. Sasikanth Reddy[d], J.S.S. Akhil[e], D. Prasanna Kumar[f], Ch. Dileep Kumar[g], K. Praveen Kumar[h] and M. Sowmya[i]

[a]*Corresponding author, Assistant Professor Department of ECE in K.L. University Green fields-522502, AP, India. Email: muralikrishna@kluniversity.in*
[b-g]*Associate Professor Department of ECE in K.L. University Green fields-522502, AP, India. Email: [b]shameemsyed@kluniversity.in*
[h,i]*UG Student Department of ECE in K.L. University Green fields-522502, AP, India. Email: [h]praveenkumar205.pk@gmail.com*

Abstract: Electronic devices are turn out to be a part of human life due to rapid growth of smart technology. Communication between two different electronic devices involves either wired or wireless channel. Data security in wired or wireless channel is prime constraint. Cryptography is a solution for safe and secure transmission of data through channel. Data usage is increasing proportionally with increase of usage web sources, banking transactions and mobile communication etc. The proposed new algorithm performs some permutations to One Time Pad (OTP) algorithm. OTP algorithm encrypts the message multiplexed with a random key. Cracking possibility of the algorithm is less due to random key for each byte. Security level is enhanced in the proposed algorithm, which is designed using Verilog HDL, Synthesized & Simulated in Xilinx-ISE Simulator and results are tested on Spartan FPGA.

Keywords: Data Security, Cryptography, Channel, One Time Pad Algorithm, FPGA.

1. INTRODUCTION

Due to the advancements in Electronic Technology, people are erected for use of smart devices. Growth in wireless communication like 3G, 4G, and Wi-Fi services which are integrated with smart devices plays a crucial role in transferring bulk data over media like videos, music, photos and banking information from one place to another through internet in seconds. Banking transactions can be done through smart devices [1]. Valuable information can be theft by hackers. Cryptography is a common solution for data theft. Antivirus uses crypto mechanism which provides a 128 bit security SSL Layer encrypt the user's data in safe Internet browsing while performing banking and authorized web transactions [2]. Attacks can be done on data either by reading or modifying them known as passive and active attacks. Cryptography forms a secure channel between sender and receiver which encrypts information at sender with key by using variety of algorithms and decrypts the encrypted data with key, and then the original message is retrieved back at receiver shown in Figure 1.

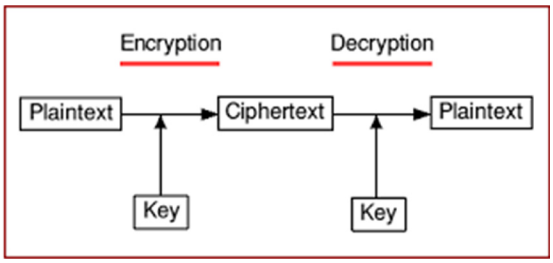

Figure 1: Encryption and Decryption

Although having multiple encryption mechanisms available for data security [3]. Each technique has its own protocol in securing data from hackers. Depth of encryption technique depends on several parameters like length of message and no of keys & permutations which enhances the security level, when passes through channel. Cryptography algorithms are classified into two types they are Symmetric and an Asymmetric encryption techniques.

1.1. Symmetric Encryption Technique

In Symmetric Encryption Technique uses a common key i.e., shared between sender and receiver as an agreement. Sender encrypts the message with key produces a cipher which sends to receiver. Receiver performs several permutations on cipher along with key to retrieve or decrypt the original text message. Same key is used to encode the message with some permutations at encrypt and decode the message with reverse permutations decrypt the message shown in Figure 2.

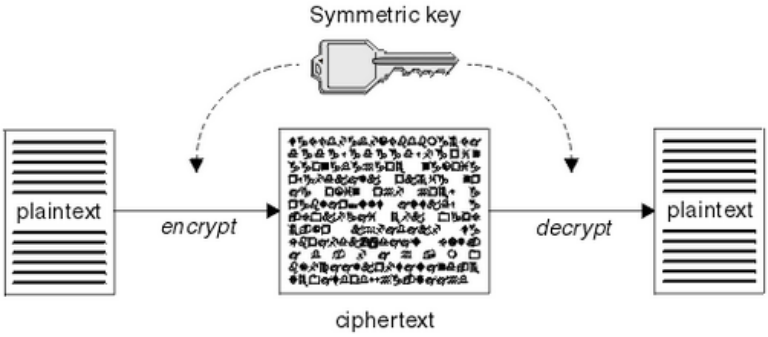

Figure 2: Symmetric Key for Encryption & Decryption

1.2. Asymmetric Encryption Technique

In Asymmetric Encryption Technique uses a common key i.e., shared between sender and receiver as an agreement known as private key. Sender uses two different keys. One is private key another one is public key. Sender encrypts the message first with private key produces a pre-cipher later encrypts message with public key produces cipher. Sender shares a cipher and public key to receiver. Receiver decrypts the message from cipher by performing permutation with public key produces a partial data. Performing permutations on partial data with pre shared secret key called private key to decrypt the original text message [4]. Receiver never decrypts the original text message without a pre sharing private or secret key shown if Figure 3.

2. ONE TIME PAD ALGORITHM

One Time Pad (OTP) algorithm is cryptography technique that message is encrypted with a secret randomly generated key [5]. Design methodology of OTP algorithm is each character or byte in a message is paired with

Figure 3: Asymmetric Key for Encryption & Decryption

random key using modular addition. Key is generated using Linear Feedback Shift Register (LFSR). LFSR generates random sequences by shift and xor based on polynomial mechanism [6]. Randomness in key can be enhanced by using preloaded seed value or by changing polynomial [7]. Several types of LFSR's are available based on the application LFSR is chosen as Pseudo Random Sequence generator. Key should not be reused completely and also the part of the message should not be reused. Due to Pseudo Random nature of key mechanism sender assigns separate key for each byte of the message shown in Figure 4.

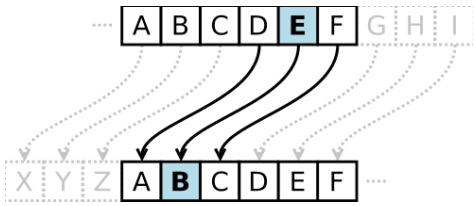

Figure 4: Pseudo Random Sequence Key for Encryption & Decryption

It is very difficult for hacker to break the cipher to original message. Length of the message is same as length of key [8]. Sender should share message and key to receiver to decrypt the original message.

2.1 One Time Pad Encryption Technique

Encrypted message is obtained by combining randomly generated key and original message using modular addition mechanism. Let the input message to be encrypted as HELLO. Randomly generated key is XMCKL. Numbers assigned to each alphabet from *a* to *z* starting from 0, so H is numbered as 7. X in key is numbered as 23. Performing modular addition to H & X obtained value is 30. The number obtained after modular addition is greater than 26 perform mod 26, so the value obtained is 4. Alphabet assigned for number 4 is E, the generated final cipher text is EQNVZ shown in Figure 5.

```
      H        E        L        L        O  message
   7 (H)    4 (E)   11 (L)   11 (L)   14 (O) message
+ 23 (X)   12 (M)    2 (C)   10 (K)   11 (L) key
= 30       16       13       21       25     message + key
=  4 (E)   16 (Q)   13 (N)   21 (V)   25 (Z) (message + key) mod 26
      E        Q        N        V        Z  → ciphertext
```

Figure 5: OneTime Pad Encryption Technique

2.2. One Time Pad Decryption Technique

Decrypted message is obtained by combining randomly generated key and cipher shared from sender [1]. Original message is obtained by using modular arithmetic mechanism. Let the input cipher message to be decrypted as EQNVZ. Key from sender is XMCKL. Numbers assigned to each alphabet from *a* to *z* starting from 0. X in key is numbered as 23, E is as 4. Performing modular subtraction with key from cipher to E & X obtained value is

–19. The number obtained by performing mod 26, is 7. Alphabet assigned for number 7 is H; the generated final decrypted message is HELLO, shown in Figure 6.

```
       E       Q       N       V       Z  ciphertext
    4 (E)  16 (Q)  13 (N)  21 (V)  25 (Z) ciphertext
-  23 (X)  12 (M)   2 (C)  10 (K)  11 (L) key
= -19       4      11      11      14     ciphertext - key
=   7 (H)   4 (E)  11 (L)  11 (L)  14 (O) ciphertext - key (mod 26)
       H       E       L       L       O  → message
```

Figure 6: OneTime Pad Decryption Technique

3. PROPOSED OTES ALGORITHM

Proposed One Time Pad Enhanced Security algorithm is cryptography technique. Message is encrypted with a secret key generated by LFSR type I & II using preloaded seed value or by changing polynomial in runtime to obtain randomness in key shown in Figure 7. Design methodology of OTPES algorithm is each character or byte in a message is paired with random key. Applying some permutations on existing OTP algorithm helps to overcome the limitation of message and key length.

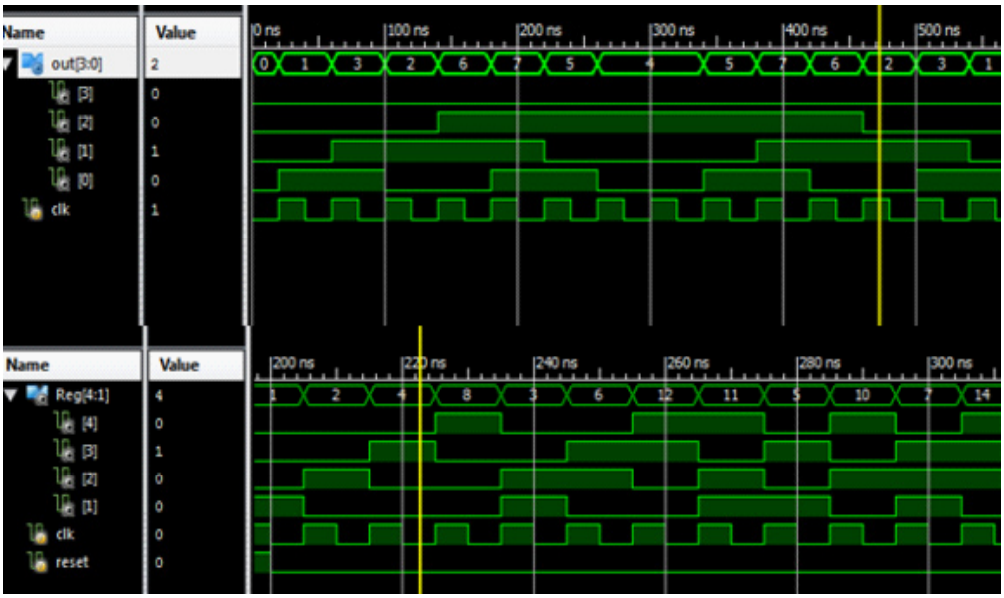

Figure 7: LFSR type I & II Seed Value and Polynomial

3.1. Proposed OTPES Encryption Technique

Step_1: Plain text (original Message) should be xored with key.

Step_2: The result obtained at Step_1 string is reversed.

Step_3: Obtained result in Step_2 is sliced into two parts, circular left shift should be applied to left part of the string and circular right shift should be applied to right part of the string.

Step_4: Performing 1's complement on the result generated in Step_3.

Step_5: The sequence obtained in Step_4 is reversed to produce cipher text message shown in flowchart for proposed OTPES Encryption Technique Figure 8.

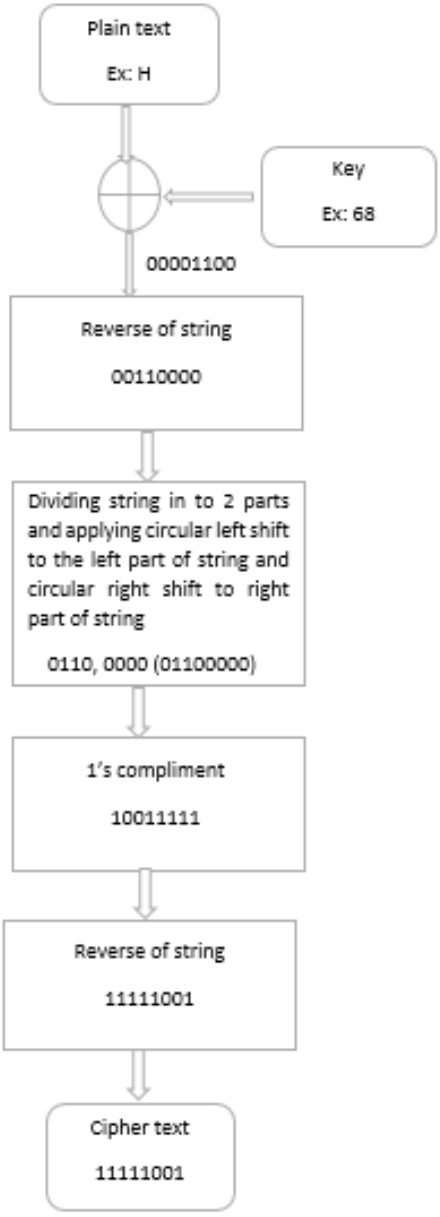

Figure 8: Flow Chart for Proposed OTPES Encryption Technique

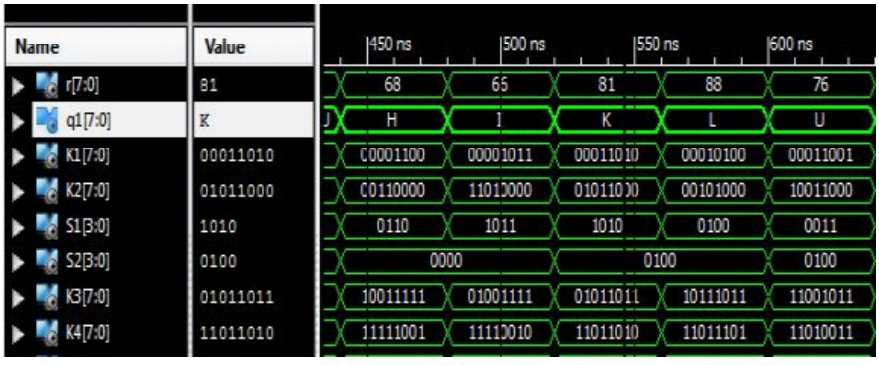

Figure 9: Proposed OTPES Encryption Technique

Simulation Result of Proposed OTPES Encryption Technique shown in Figure 9. In this proposed method, *r* shows random number, q1 shows the message each character or byte of a message is paired with random key. Xor operation between paired message and random key generates K1. K2 is obtained by reversing K1 sequence. Dividing the K2 sequence into two parts named as S1 & S2. Perform circular right shift on S2 and circular left shift on S1.

After shift permutation the result in register is 01100000. K3 is obtained by performing one's complement on register. K4 is obtained by reversing K3 sequence. The final cipher obtained at K4. It is difficult for hacker to crack the key due random sequence. Even if the hacker guesses the key there are several permutations to be performed on cipher along with key to break the message.

3.2. Proposed OTPES Decryption Technique

Step_1: Cipher text should be reversed.

Step_2: Performing 1's compliment on the Result obtained in the Step_1.

Step_3: The result produced in Step_2 is partitioned in to two parts, circular right shift should be applied to the left part of the string and circular left shift should be applied to the right part of string.

Step_4: The obtained string in Step_3 is reversed which produces Step_4.

Step_5: The outcome of Step_5 was generated by performing xor operation between key & Step_4 to obtain Original text message (Plain Text) shown in flowchart for proposed OTPES Decryption Technique Figure 10.

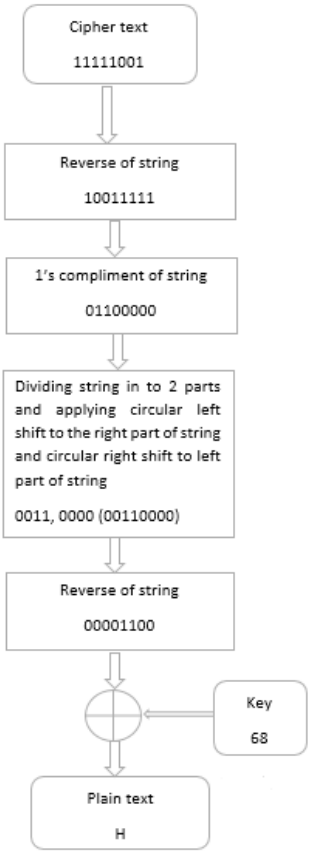

Figure 10: Flow Chart for Proposed OTPES Decryption Technique

From the above Figure 11 shows Simulation Result of Proposed OTPES Decryption Technique. In this proposed method, the final cipher obtained at K4 is reversed to obtain K5. Performing ones complement on K5, K6 is generated. Dividing the K6 sequence into two parts, left part named as S3 & right part as S4. Perform circular right shift on S3 and circular left shift on S4.After shift permutation the result in register is 00110000. K7 is obtained by re arranging S3 & S4 sequences. K8 is generated by reversing K7 sequence. The original message is decrypted at K9 by performing xor operation between K8 and key. Several permutations on cipher and key is enhance the security level before decrypting original message. Final encryption and decryption simulation result is shown in Figure 12. Random nature of key is enhanced by composing few DNA properties in key generation shown in Figure 13. RTL & Technology schematic of proposed OPES algorithm is shown in Figure 14.

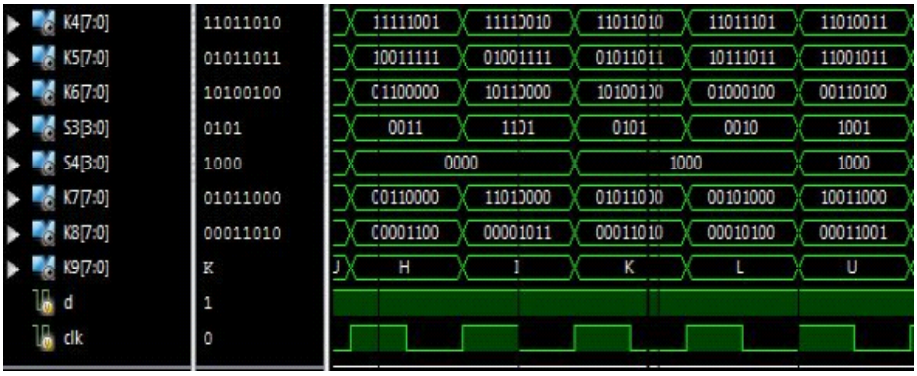

Figure 11: Proposed OTPES Decryption Technique

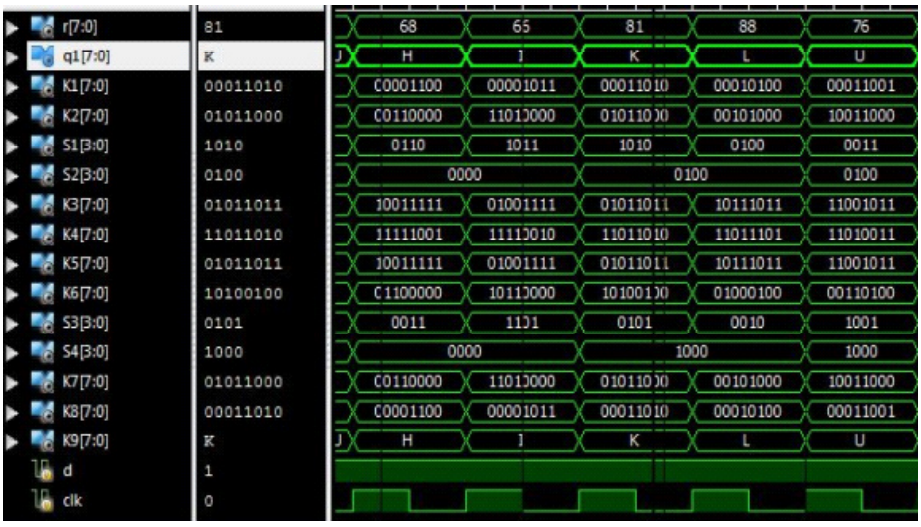

Figure 12: Proposed OTPES Encryption & Decryption

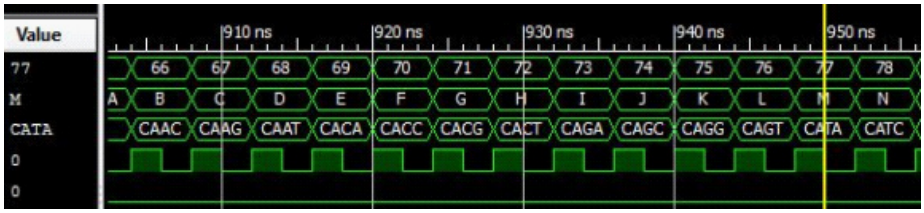

Figure 13: Key generation with DNA Properties

Table 1
Summary of Utilized Resources

Device Utilization Summary (estimated values)			*[-]*
Logic Utilization	*Used*	*Available*	*Utilization*
Number of Slices	16	4656	0%
Number of Slice Flip Flops	17	9312	0%
Number of 4 input LUTs	29	9312	0%
Number of Bonded IOBs	112	232	48%
Number of GCLKs	1	24	4%

The above table shows the resources utilized for the proposed OTPES algorithm for Spartan XC3S500E.

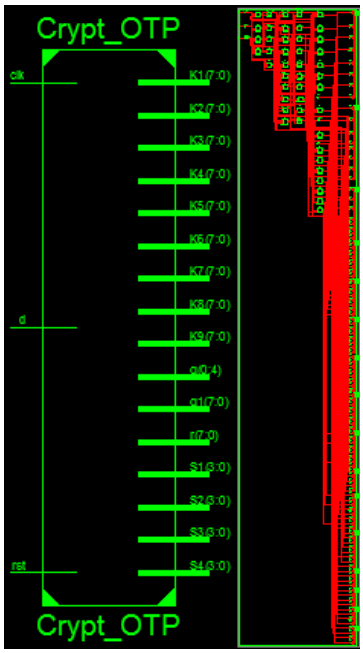

Figure 14: RTL and Technology Schematic of Proposed OTPES Encryption-Decryption Technique

4. HARDWARE IMPLEMENTATION PROPOSED OTPES ALGORITHM

Due to evolvable nature of programmable hardware FPGA adopts for configured designs on run time. Cryptographic algorithms should prove on hardware rather than simulation, which depict the depth of the algorithm on evolvable hardware. Hardware Implementation proves the scenario of real time challenge for the designs and realizes the working nature of algorithm between sender and receiver. Chipscope Pro analyzer provides efficient view of real time hardware simulation of the adopted design. The Hardware Simulation of proposed OTPES algorithm using Chipscope Pro on Spartan3E (Nexys2) FPGA is shown in Figure 15.

Figure 16 shows FPGA Implementation of On-Chip Simulation using Chipscope Pro of proposed OTPES algorithm on Virtex XC7z020clg484-1architecture. Figure 17 reveals RTL Schematic view of Proposed OTPES Algorithm with Integrated Logic Analyzer to monitor signals on runtime using Logic Analyzer.

Figure 18 proves physical view of OTP Algorithm Implementation on reconfigurable hardware using Zed Board. Figure 19 exhibits the Resources Utilized for the proposed OTPES algorithm for Virtex FPGA Architecture.

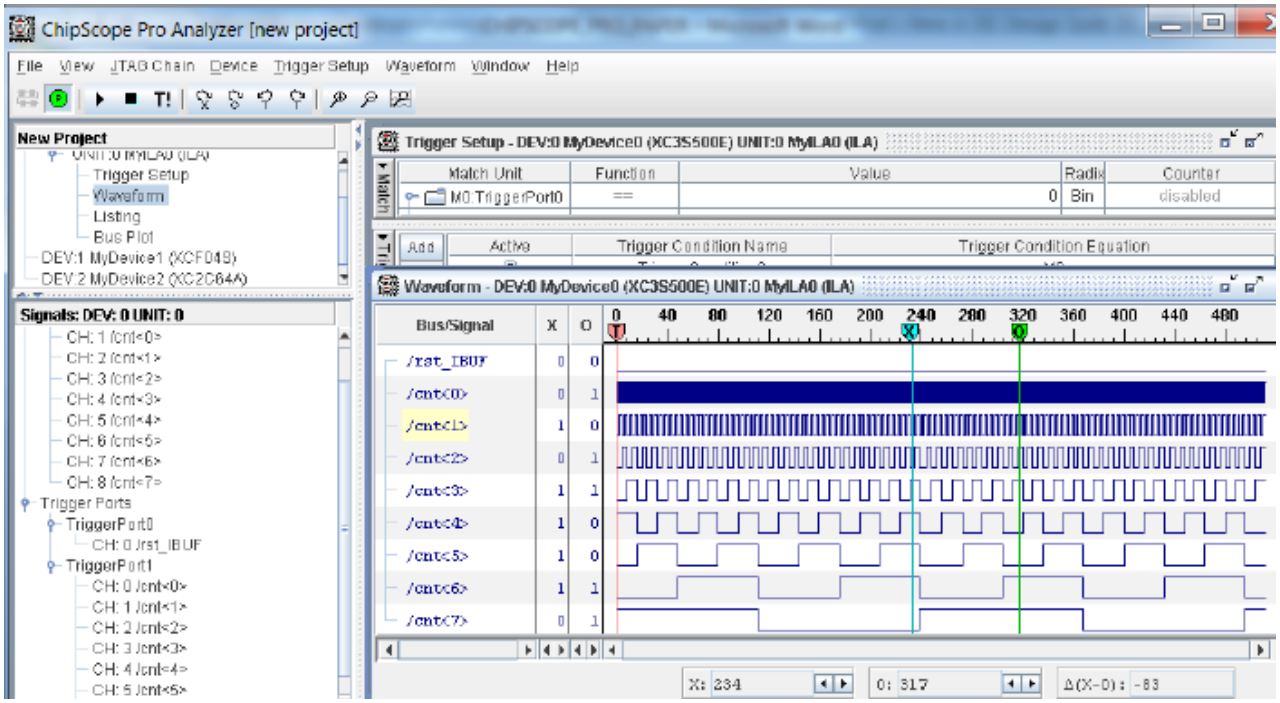

Figure 15: FPGA Implementation of On-Chip Simulation of proposed OTPES algorithm on Spartan XC3S500E

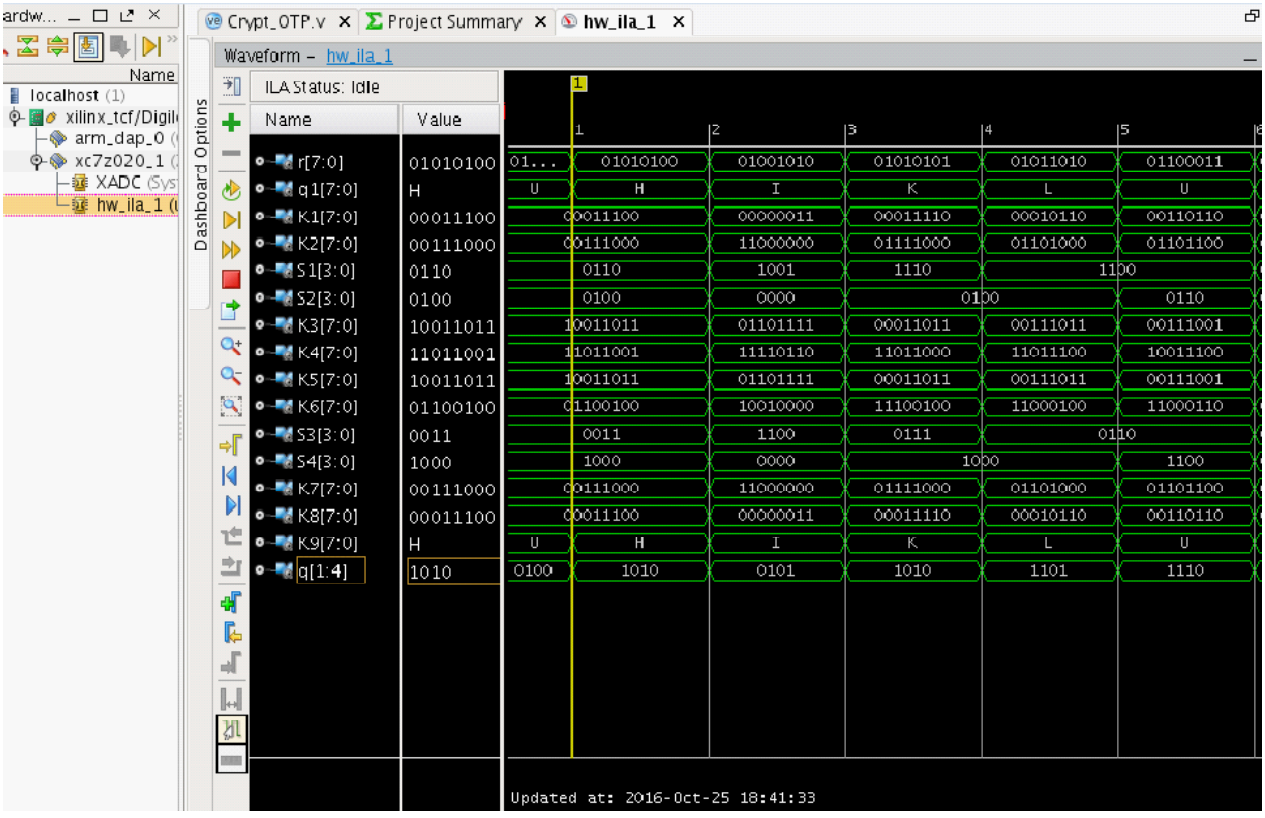

Figure 16: FPGA Implementation of On-Chip Simulation of proposed OTPES algorithm on Virtex XC7z020clg484-1

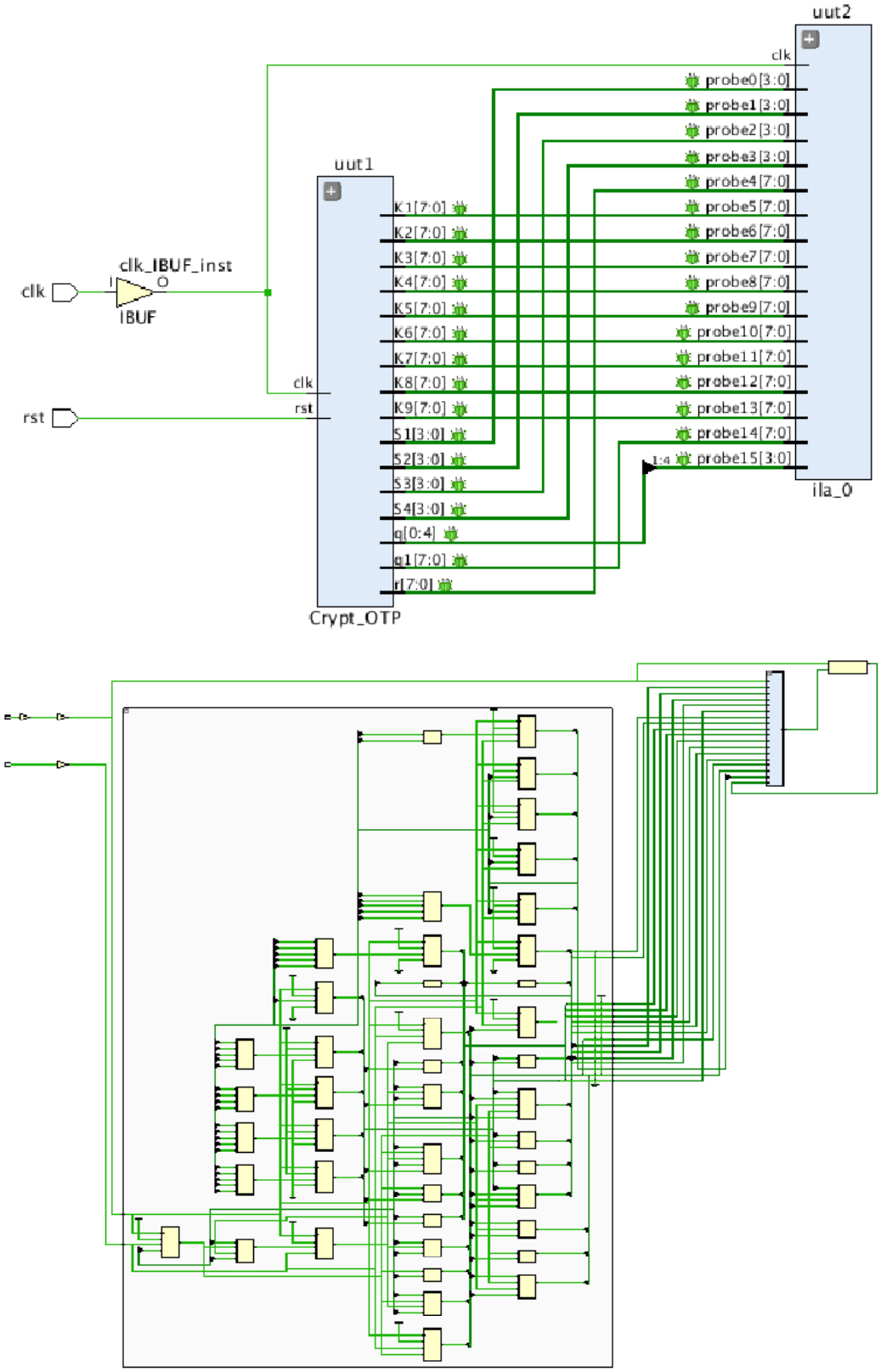

Figure 17: RTL Schematic of Proposed OTPES Algorithm with Integrated Logic Analyzer

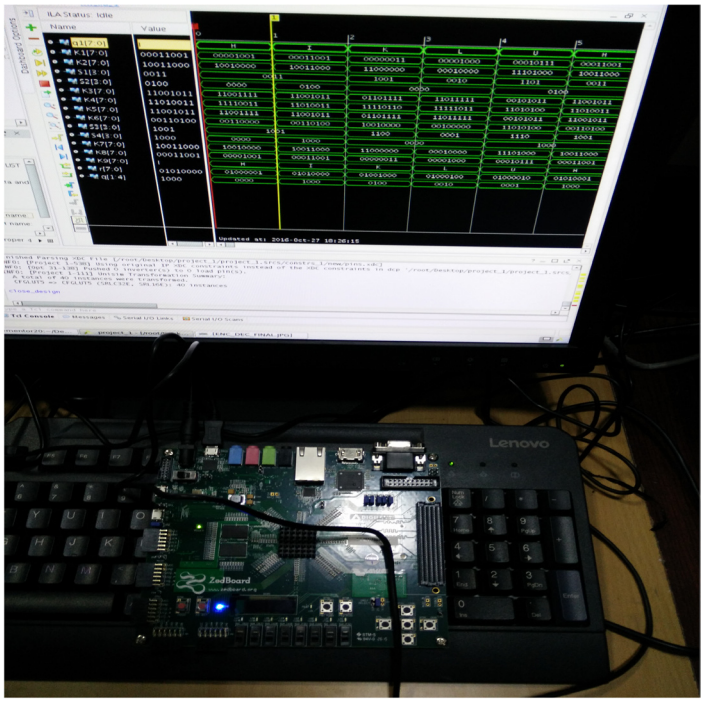

Figure 18: Physical View of OTP Algorithm Implementation on Zed Board

Logic Utilization	*Used*	*Available*	*Utilization*
LUT	23	53200	0.04
Flip-Flops	17	106400	0.02
IO	2	200	1
BUFG	1	32	3.13

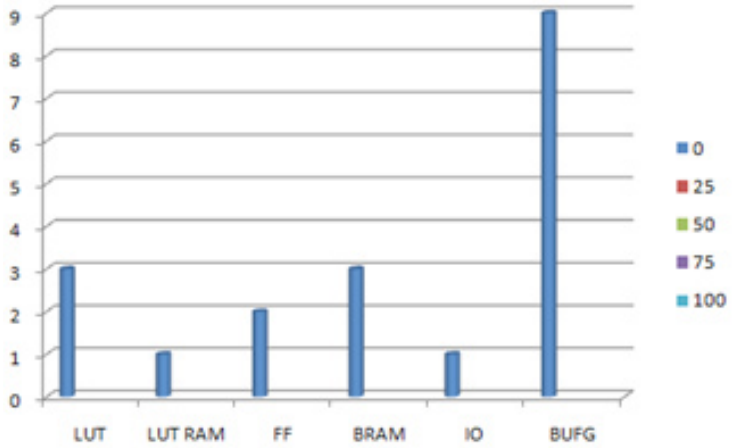

Figure 19: Resources Utilized for the proposed OTPES algorithm for Virtex FPGA XC7z020clg484-1

5. CONCLUSION

Despite the fact that Internet is the medium which is widely used for e-banking and transactions, research shows it is insecure. It is vulnerable to several threats like phishing, spoofing and hacking. Attackers are using advanced techniques to access confidential data by breaking algorithms used for financial transactions. Research proves

that OTP SMS is also under threat due to lack of advancement. Several cryptography techniques which are used to hide the data use different protocols. One Time Pad is a cryptographic algorithm which encrypts the message multiplexed with a random key for each byte. Cracking possibility of the algorithm is less due to random nature of key generation mechanisms by imposing few DNA Properties. Security level is enhanced in the proposed algorithm, by performing some permutations on key mechanisms along with message to produce cipher. It also addresses the five principles of secure services, and hence proves to be reliable over other protocols.

REFERENCES

[1] Ashraf Aboshoshaet. at.: Multi-Channel User Authentication Protocol based on Encrypted Hidden OTP. *International Journal of Computer Science and Information Securitty* Vol. 13.No. 6, 14–19 (2015).

[2] G.C.C.F. Pereira, M.A.S. Santos, B.T. de Oliveira, M.A. Simplicio, P.S.L.M. Barreto, C.B. Margi, et. al., "SMSCrypto: A lightweight cryptographic framework for secure SMS transmission", *The Journal of Systems and Software*, Vol. 86, pp. 698-706, 2013

[3] K.K. Brajesh, "An Approach For User Authentication One Time Password (Numeric And Graphical) Scheme", *Journal of Global Research in Computer Science*, Vol. 3, pp. 54-57, 2012.

[4] Ch. santhoshreddy "Poly-alphabetic symmetric key algorithm using randomized prime numbers "international journal of scientific and research publications volume-2, issue,9, September 2012, ISSN 2250 3153.

[5] G. Sainath, S. Shashank, S. Pruthvi, V. Siddhartha and V. G. Suryakanth, "Passblot: A Highly Scalable Graphical One Time Password System", *International Journal of Network Securit& Its Applications (IJNSA)*, Vol. 4, No. 2, pp. 201-206, March 2012.

[6] U. Maurer, "Constructive cryptography-A new paradigm for security definitions and proofs", *Theory of Security and Applications Ser. Lecture Notes in Computer Science*, Vol. 6993, pp. 33-56, 2012.

[7] Sharad Patil, Dr. Ajay Kumar "Modified One Time Pad Data Security Scheme: Random Key Generation Approach " *International Journal of Computer and Security* Volume 3 issue 2 March/April 2009 Malaysia.

[8] J. Zhou, O.C. Au, and P.H.W. Wong, "Adaptive chosen-ciphertext attack on secure arithmetic coding," *IEEE Transactions on Signal Processing*, Vol. 57, No. 5, pp. 1825–1838, 2009.

[9] Yun Huang, Zheng Huang, Haoran Zhao, Xuejia Lai "A new One-time Password Method", *International Conference on Electronic Engineering and Computer Science in 2013.*

[10] D. M'Raihi, M. Bellare, F. Hoornaert, and D. Naccache, "HOTP: An HMAC based one-time password algorithm, RFC 4226", Dec. 2005.

[11] Zhou Lu, HuaZhang Yu: One time password generating method and apparatus, US8184872, May 2012.

[12] RSA Laboratories - OTP-PKCS #11: PKCS #11 mechanisms for One-Time Password tokens, Dec. 2005, http://www.rsa.com/rsalabs/node.asp?id=2818.

[13] Alfred Menezes, Paul van Oorschot and Scott Vanstone: Handbook of Applied Cryptography, CRC Press, Boca Raton (1997).

[14] Richard E. Newman, Piyush Harsh, and Prashant Jayaraman, "Security analysis of and proposal for image-based authentication", CISE Dept., University of Florida, Gainesville FL 32611-6120{nemo, pharsh, pjayaram}@cise.ufl.edu.

[15] N. Haller, C. Metz, P. Nesser, M. Straw: A One-Time Password System, Feb. 1998, http://www.ietf.org/rfc/rfc2289.txt.

[16] Teoh, A.B.J., Ngo, D.C.L, Goh, A. 2004. BioHashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recognition. Vol. 37(11), pp. 2245-2255.

[17] Lumini Alessandra and Loris Nanni. 2006. Empirical tests on BioHashing. Neurocomputing. Vol. 69, pp. 2390-2395.

[18] Anil K. Jain, KarthikNandakumar and Abhishek Nagar. 2008. Biometric Template Se.