International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 913-920 June - Dec 2018 www.csjournals.com

XOR Cipher Based Cryptography and Authentication with Hardware Chip

[1]Nitin Gupta, [2]Prof (Dr) Sarvottam Dixit
[1]Research Scholar, [2]Professor
Department of Computer Science &Engineering, Mewar University Chittorgarh, (Rajasthan),INDIA

Abstract: In the context on network security, the fundamental building block is user authentication. User accountability and access control are dependent on user authentication. Authentication is an indispensable part of Cryptography, which is an unconditionally secure key distribution technique based on the laws of nature. The paper presents the hardware chip design and simulation of XOR encryption and decryption method called XOR cipher technique. The chip design is done in Xilinx 14.2 software and function simulation in Modelsim 10.0 software using VHDL Programming. The chip is verified on Virtex 5- FieldProgrammable Gate Array (FPGA) for 8-bit, 16-bit, 32-bit, 64-bit and 128-bit block cipher operation.

Keywords: Cryptographic Encryption and Decryption, XOR Cipher, VHDL Programming.

1. Introduction

The authentication is the process that assurances about the identification and confirmation of a user's identity. It is one of the sources of information assurance in the context of computer system. The main important features of a secured computer system are availability, confidentiality, integrity, authentication and nonrepudiation. The process of authentication is happening when a user is trying to access the data and information. Essentially, it is required to provide the access rights and identity by the user. When the system is login the user must enter the username and password for authentication process. The login processes must be allocated to each user. There are different methods to authenticate the system such as smart cards, RFID, fingerprint, human eyes. The better form of authentication is biometrics, which depends on the user's existence and biological makeup (i.e., fingerprints, retina). The technology provides the more secured system against hackers to breakdown the computer systems or stealing the information. The security of the system is the primary concern and cryptography play very important role for encryption and decryption the data in the system with authentication.

2. Authentication and Cryptography

Authentication [1, 2] is the process of verifying that someone or something are who they claim to be before they are granted access to protected resources. Cryptography allows people to carry over the confidence found in the physical world to the electronic world, thus allowing people to do business electronically, without worries of deceit and deception. Everyday hundreds of people interact electronically, whether it is through E-mail, e-commerce (business conducted over the internet), ATM machines, or cellular phone. Cryptography makes [3, 4] secure web sites and electronic safe transmissions possible. The perpetual increase of information transmitted

electronically has led to an increased relies on cryptography. Authentication is a simple function where one party presents a set of credentials to a system. If the credentials match a given set on the system, the system returns a value that represents authorization; otherwise it does not. The purpose of authentication is to verify that the specific information presented represents a request to be authentic from a specified entity The Public Key Infrastructure (PKI) based authentication techniques follow the digital records and certificates to verify a user's identity. In Cryptography encryption [5] the original message or data is called plain text which is encoded with key, called cipher text and transmitted over a channel. Description [6] is the reverse process, in which the plain text is decoded from the cipher text. With the help of secret key and cipher text it produces the original plain text. Cryptography involves encryption and decryption with the sharing of same key at both end or the different key on both ends. There are mainly two types of encryption algorithms called symmetric and asymmetric algorithm. Symmetric key algorithm is also called a private key algorithm and symmetric key algorithm is called public key. The model of cryptography [7, 10] is shown fig.1 in which plaintext *(T)* is encrypted with key value (*Key*) and transmitted cipher text is *B = E [key, T]*, the same text is extracted with decryption algorithm *T = D [Key, B]*, and same key (*Key*).

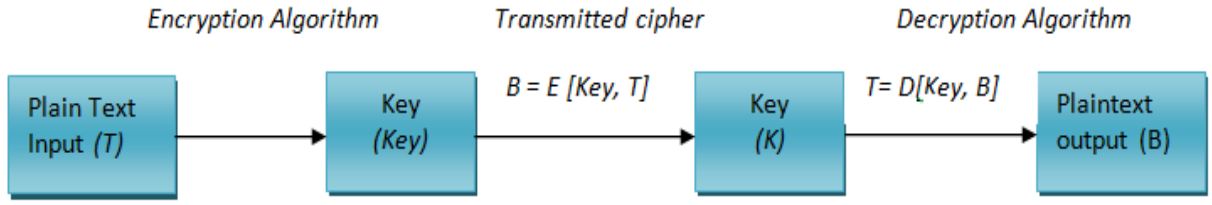


Fig. 1 Encryption and decryption

3. XOR Encryption and Decryption (XORED) Method

The XOR encryption and decryption technique [8] is based on the XOR operation of the plain text with the key value to get the cipher in encryption end. The cipher text is again XORed with the key value receiving end to decrypt the same text which was sent at transmitting end. It is possible in case of symmetric cryptography in which the encryption and decryption both keys are same. The size of the key is random and plain text size is also same as key size. The XOR operation is also called modulo-2 addition. The XOR operation is following the additive properties

$$X \oplus 0 = X$$
$$X \oplus X = 0$$
$$(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z)$$
$$(X \oplus Y) \oplus Z = X \oplus 0 = X$$

The operator is the $\oplus$ exclusive OR (XOR) operation and it has the behavior given as.

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

The XOR cipher method is shown in Fig.2

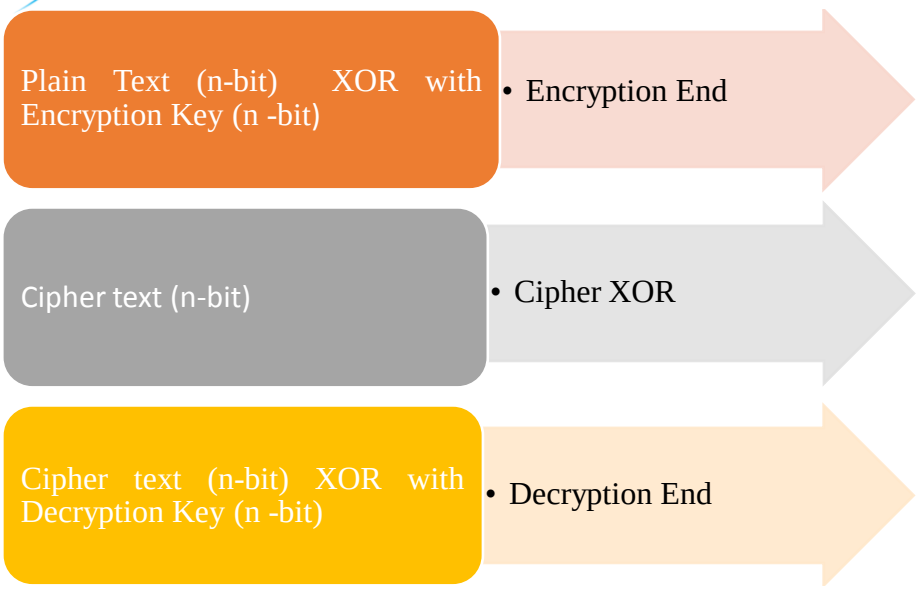


Fig. 2 XOR Cipher

Encryption Process: The encryption process has the following steps
Step-1:Read the complete plain text byte by byte or read each 8-bit ASCII character from the full string 'n' bit of plain text from LSB.
Step-2:Based on the size of plain text, apply the encryption key value of 'n' bit.
Step-3:Accomplish the bit wise XOR operation on plain text and encryption key value.
Step-4:Read the corresponding binary value as cipher text**.**
Step-5:Perform the step-1 to step-4 operation till End of File (EoF) is completed.
Decryption Process: The encryption process has the following steps
Step-1:Read the 8-bit ASCII value of each character for 'n' bit cipher text encoded at transmitting end.
Step-2:Based on the size of plain text and cipher text, apply the decryption key value of 'n' bit.
Step-3:Accomplish the bit wise XOR operation on cipher text and decryption key value
Step-4:The decoded binary value at the receiving end is original plain text of ' n bit' which is read against 8-bit ASCII value as original character till End of File (EoF) is completed. The decoded text is the original text sent on transmitting end.

4. Results and Discussions

The chip design of the XOR cipher is done in Xilinx ISE 14.2 software. The RTL view of the chip is shown in fig.3 and its internal schematic in fig.4. the chip has clk, reset, Input_plain_text<127:0>, passowrd_enryption<127:0> and passowrd_deryption<127:0> and selction_inputs as the inputs and Cipher_XOR<127:0> and Output_plain_text<127:0> as the outputs. Input_plain_text<127:0>, and Output_plain_text<127:0> as input and output plantext data as textual text of the encryption an decryption end. The size of the text can vary of 'N' bit. The passowrd_enryption<127:0> and passowrd_deryption<127:0> are encryption key and decryption key as password. Clock is 1 bit used to provide rising clock pulse to work digital logic at 50 % duty cycle. Reset pin is used to reset the logic circuitry and synchronized with clock pulse. Selection_input is 1-bit input in chip to decide chip operation in encryption mode only or in encryption/ decryption. Cipher_XOR<127:0> is the XOR cipher achieved at

transmitting end after XOR encryption. The Modelsim simulation waveform are given in fig.5 and corresponding simulated values are listed in table 1.

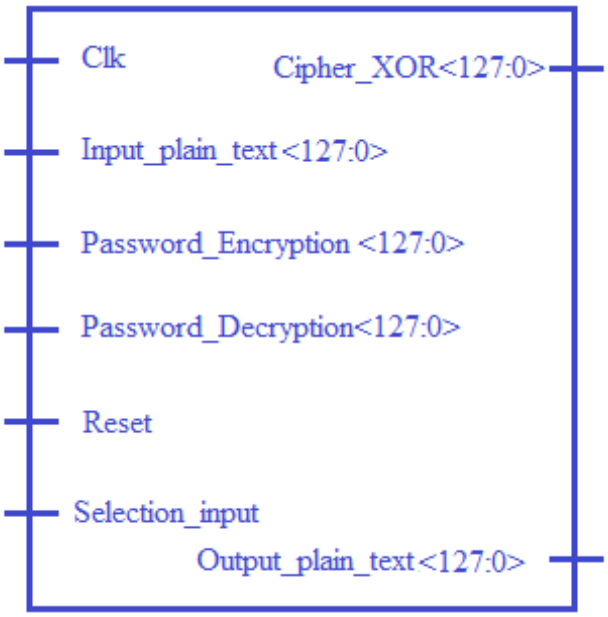


Fig.3. RTL view of chip

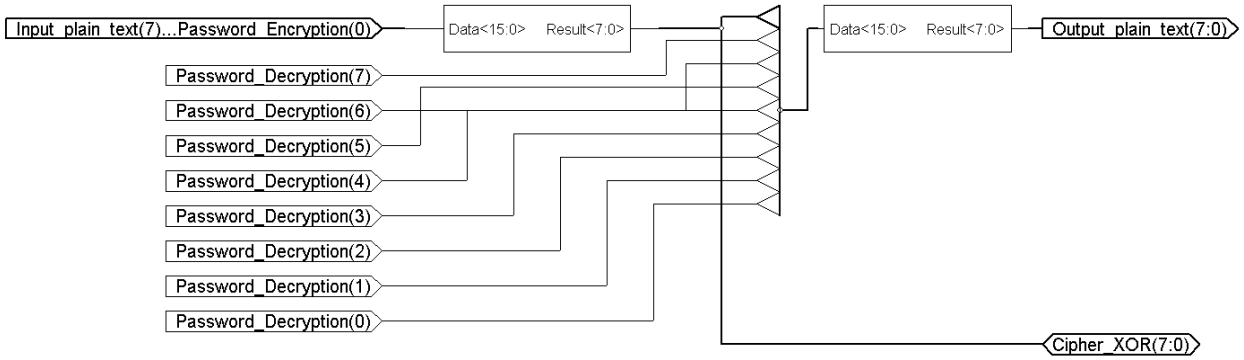
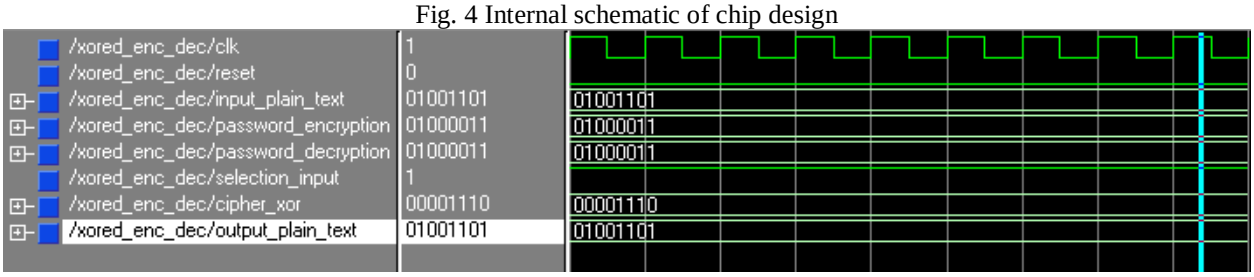
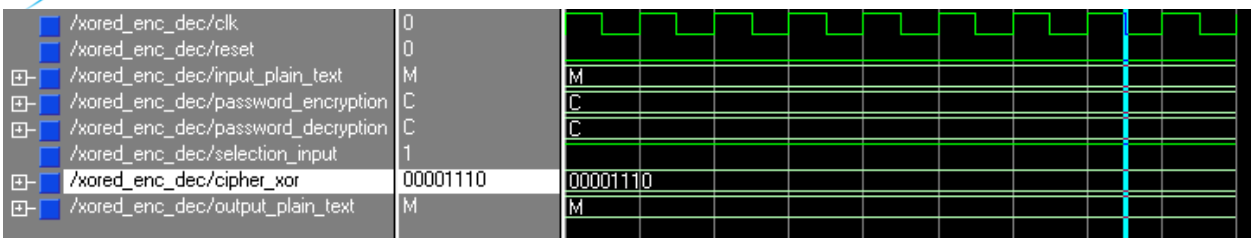


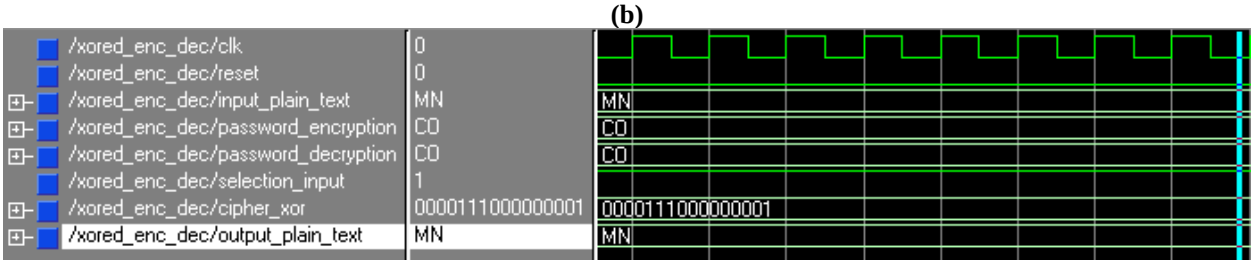
Fig. 4 Internal schematic of chip design



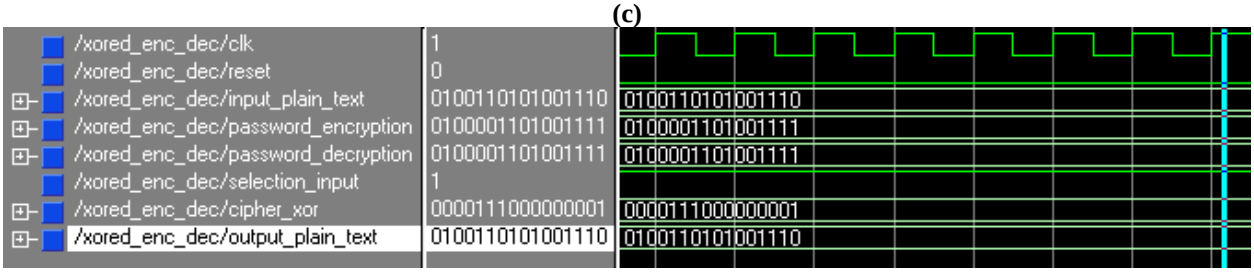
(a)


/xored_enc_dec/clk 0
/xored_enc_dec/reset 0
/xored_enc_dec/input_plain_text M
/xored_enc_dec/password_encryption C
/xored_enc_dec/password_decryption C
/xored_enc_dec/selection_input 1
/xored_enc_dec/cipher_xor 00001110
/xored_enc_dec/output_plain_text M

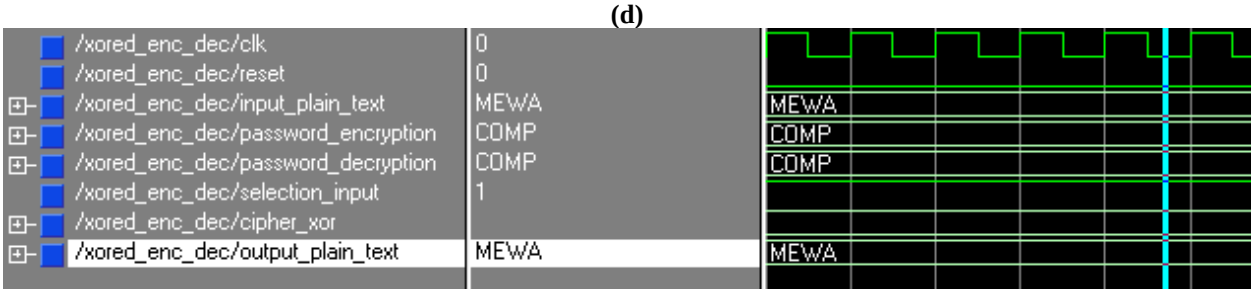
(b)


/xored_enc_dec/clk 0
/xored_enc_dec/reset 0
/xored_enc_dec/input_plain_text MN
/xored_enc_dec/password_encryption CO
/xored_enc_dec/password_decryption CO
/xored_enc_dec/selection_input 1
/xored_enc_dec/cipher_xor 0000111000000001
/xored_enc_dec/output_plain_text MN

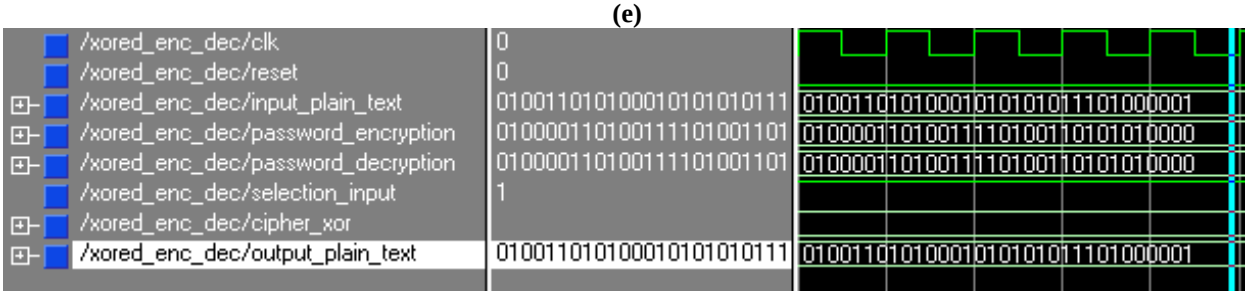
(c)


/xored_enc_dec/clk 1
/xored_enc_dec/reset 0
/xored_enc_dec/input_plain_text 0100110101001110
/xored_enc_dec/password_encryption 0100001101001111
/xored_enc_dec/password_decryption 0100001101001111
/xored_enc_dec/selection_input 1
/xored_enc_dec/cipher_xor 0000111000000001
/xored_enc_dec/output_plain_text 0100110101001110

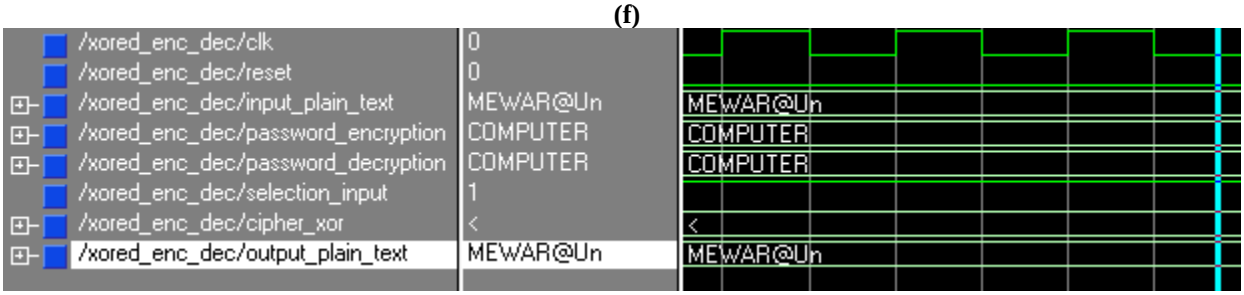
(d)


/xored_enc_dec/clk 0
/xored_enc_dec/reset 0
/xored_enc_dec/input_plain_text MEWA
/xored_enc_dec/password_encryption COMP
/xored_enc_dec/password_decryption COMP
/xored_enc_dec/selection_input 1
/xored_enc_dec/cipher_xor
/xored_enc_dec/output_plain_text MEWA

(e)

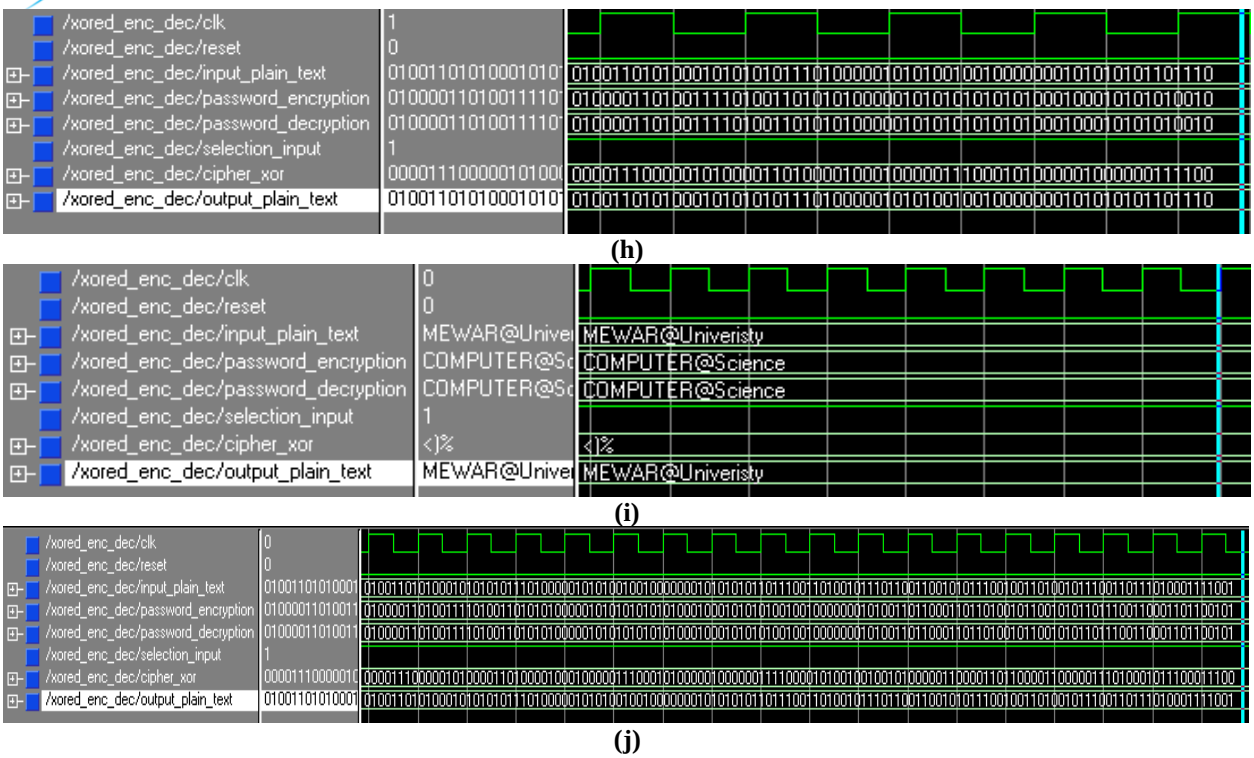

/xored_enc_dec/clk 0
/xored_enc_dec/reset 0
/xored_enc_dec/input_plain_text 010011010100010101010111
/xored_enc_dec/password_encryption 010000110100111101001101
/xored_enc_dec/password_decryption 010000110100111101001101
/xored_enc_dec/selection_input 1
/xored_enc_dec/cipher_xor
/xored_enc_dec/output_plain_text 010011010100010101010111

(f)


/xored_enc_dec/clk 0
/xored_enc_dec/reset 0
/xored_enc_dec/input_plain_text MEWAR@Un
/xored_enc_dec/password_encryption COMPUTER
/xored_enc_dec/password_decryption COMPUTER
/xored_enc_dec/selection_input 1
/xored_enc_dec/cipher_xor <
/xored_enc_dec/output_plain_text MEWAR@Un

(g)

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 913-920 June - Dec 2018 www.csjournals.com



(h)

(i)

(j)

Fig.5. Simulation Modelsim Waveform (a) 8-bit ASCII (b) 8-bit binary (c) 16-bit ASCII (d) 16-bit binary) (e) 32-bit ASCII (f) 32-bit binary (g) 64-bit ASCII (h) 64-bit binary (i) 128-bit ASCII (j) 128-bit binary

Table 1 Simulated Output Values

Test	Values
Case-1 (8 bit)	Input_plain_text = "01001101" in binary, 'M' in ASCII Password_encryption = "01000011" in binary, 'C' in ASCII Selection_input = '0' for encryption and Selection_input = '1' for = ' 1' for decryption Cipher_text = "00001110" in binary Decryption_key = ""01000011" in binary, 'C' in ASCII Output_plain_text= "01001101" in binary, 'M' in ASCII
Case-2 (16 bit)	Input_plain_text = "01001101 01000101" in binary, 'ME' in ASCII Password_encryption = "01000011 01001111" in binary, 'CO' in ASCII Selection_input = '0' for encryption and Selection_input = '1' for = ' 1' for decryption Cipher_text = "00001110 00001010" in binary Password_decryption= "01000011 01001111" in binary, 'CO' in ASCII Output_plain_text= "01001101 01000101" in binary, 'ME' in ASCII
Case-3 (32 bit)	Input_plain_text = "01001101 01000101 01100001 01010010" in binary, 'MEWA' in ASCII Password_encryption = "01000011 01001111 01001101 01010000" in binary, 'COMP' in ASCII Selection_input = '0' for encryption and Selection_input = '1' for = ' 1' for decryption Cipher_text = "00001110 00001010 00011010 00010001" in binary Password_decryption ="01000011 01001111 01001101 01010000" in binary, 'COMP' in ASCII Output_plain_text = "01001101 01000101 01100001 01010010" in binary, 'MEWA' in ASCII
Case-4 (64 bit)	Input_plain_text = "01001101 01000101 01100001 01010010 01010010 01000000 01010101 0110 1110" in binary, 'MEWAR@Un' in ASCII Password_encryption = "01000011 01001111 01001101 01010000 01010101 01010100 01000101 01010010" in binary, 'COMPUTER' in ASCII Selection_input = '0' for encryption and Selection_input = '1' for = ' 1' for decryption Cipher_text = "00001110 00001010 00011010 00010001 00000111 00011110 101000000 00111100" in binary Password_decryption = "01000011 01001111 01001101 01010000 01010101 01010100 01000101

	01010010" in binary, 'COMPUTER' in ASCII Output_plain_text = "01001101 01000101 01100001 01010010 01010010 01000000 01010101 0110 1110" in binary, 'MEWAR@Un' in ASCII
Case-5 (128 bit)	Input_plain_text = "01001101 01000101 01100001 01010010 01010010 01000000 01010101 0110 1110 01101001 01110110 01100101 01110010 01101001 01110011 0111100 01111001" in binary, 'MEWAR@University' in ASCII Password_encryption = "01000011 01001111 01001101 01010000 01010101 01010100 01000101 01010010 01000000 01010011 01100011 01101001 01100101 01101110 01100011 01100101" in binary, 'COMPUTER@Science' in ASCII Selection_input = '0' for encryption and Selection_input = '1' for= '1' for decryption Cipher_text = "00001110 00001010 00011010 00010001 00000111 00011110 101000000 00111100 00101001 00100110 00000110 0011011 00001100 00011101 00010111 0001 1100" in binary Password_decryption = "01000011 01001111 01001101 01010000 01010101 01010100 01000101 01010010 01000000 01010011 01100011 01101001 01100101 01101110 01100011 01100101" in binary, 'COMPUTER@Science' in ASCII Output_plain_text = "01001101 01000101 01100001 01010010 01010010 01000000 01010101 0110 1110 01101001 01110110 01100101 01110010 01101001 01110011 0111100 01111001" in binary, 'MEWAR@University' in ASCII

Table 2 and Table 3 presents the hardware and timing parameters summary for the developed design. The hardware summary includes No of slices,flip flops, LUTs, IOBs and No of gated clocks (GCLKs) used in the implementation of design. Timing details provides the information of delay, minimum period value, maximum frequency value, minimum input arrival time before clock and maximum output required time after clock. Total memory utilization value required to complete the design. The target device is: xc5vlx20t-2-ff323 synthesized with Virtex-5 FPGA.

Table 2Hardware utilization summary for 128-bit encryption/decryption logic

Parameters	**Utilization (128 bit)**
Number of Slices	252 out of 12480 2%
Number of Slice Flip Flops	342 out of 12480 3%
Number of 4 input LUTs	130 out of 12480 1%
No. of bounded I/OBs	43 out of 172 25 %
Number of GCLKs (Gated Clk)	1 out of 32 3%
Total memory usage (kB)	252390 Kb

Table 3 Timing summary for 128-bit encryption and decryption logic

Timing Parameter	**Utilization (128 bit)**
Frequency (Max)	315.00 MHz
Period (Min)	1.829 ns
Time before clk (Min)	2.341 ns
Time after clock (Max)	3.807ns
Combination delay (ns)	5.636 ns
Speed Grade	-5

5. Conclusions

Password based authentication schemes are the most widely used techniques for remote user authentication. Password based remote user authentication schemes are used to check the validity of a login request made by a remote user. Cryptography is not only protecting the data from hackers or alteration, but also applied for user authentication. The XOR cipher method is designed successfully in Xilinx 14.2 software and Modelsim for functional simulation. The design provides the best results for 8-bit to 128-bit data encryption and decryption with same size of encryption and decryption password. The synthesized results are verified successfully on xc5vlx20t-2-ff323 on Virtex-5 FPGA. The hardware utilization and timing summary are optimal value. In future we are planning to verify the results for large size of block size and key size.

References

1. Awasthi, A. K., & Lal, S. (2003). A remote user authentication scheme using smart cards with forward secrecy. *IEEE Transactions on Consumer Electronics*, *49*(4), 1246-1248.
2. Chen, L., Wei, F., & Ma, C. (2015). A secure user authentication scheme against smart-card loss attack for wireless sensor networks using symmetric key techniques. *International Journal of Distributed Sensor Networks*, *11*(4), 704502.Hindawi Publishing Corporation Volume 2015, Article ID 704502, 10 pages http://dx.doi.org/10.1155/2015/704502.
3. Gupta, N., & Kumar, M. (2015). Comparative Study of Different Authentication And Identification Algorithms In Secured Cryptography. *International Journal of Engineering Sciences & Research Technology*.
4. Gupta, N., & Kumar, M. (2015). Authentication with AES cryptographic encryption chip design and simulation International Journal of Engineering Sciences & Management, Vol. 5(1), 70-81
5. Kouser, Z., Singhal, M., & Joshi, A. M. (2016, December). FPGA implementation of advanced Encryption Standard algorithm. In *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)* (pp. 1-5). IEEE.
6. Kumar, A., Singhal, S., &Kuchhal, P. (2012). Network on chip for 3D mesh structure with enhanced security algorithm in HDL environment. *International Journal of Computer Applications*, *59*(17), 6-12.
7. Rao, M. R., & Sharma, R. K. (2017, July). FPGA implementation of combined AES-128. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.
8. Rachmawati, D., Budiman, M. A., &Aulia, I. (2018, March). Super-Encryption Implementation Using Monoalphabetic Algorithm and XOR Algorithm for Data Security. In *Journal of Physics: Conference Series* (Vol. 979, No. 1, p. 012033). IOP Publishing.
9. Rahim, R., Napitupulu, D., Nurdiyanto, H., Sari, U. F., Rizky, F., Nofriansyah, D., ... &Ihwani, M. (2018, September). Pixel image steganography using EOF method and modular multiplication block cipher algorithm. In *IOP Conference Series: Materials Science and Engineering* (Vol. 420, No. 1, p. 012084). IOP Publishing.
10. Stallings, W. (2006). *Cryptography and Network Security, 4/E*. Pearson Education India.