iJETRM
International Journal of Engineering Technology Research & Management

WHY NOT XNOR?

Mithralekha.M[1],
Murugan. A [2]
[1] M.PhilResearch Scholar, PG and Research Department of Computer Science , Dr. Ambedkar Government Arts College, Affiliated by University of Madras, Chennai, Tamil Nadu, India.
[2] Associate Professor, PG and Research Department of Computer Science, Dr. Ambedkar Government Arts College, Affiliated by University of Madras, Chennai, Tamil Nadu, India.

ABSTRACT
From 1994 ever since, L. Adleman determined a feasible use of DNA as a form of computation which fixed the seven-point Hamiltonian path problem. Scientists have found tough practical and theoretical barriers in the area of DNA computation. As Adleman and others in this field have come to visualize, there may never be a computer made from DNA, sooner that directly resembles today's silicon-based microelectronics. But exploring in amalgamation like biology and computer science will show us an unexpected and efficiently usefulinfluence much better than the separate ones. Along with promising combinational field of science, this paper described about a contemporary idea and design of DNA cryptographic algorithm based on XNOR logic gate.

Keywords:
DNA Computation, Cryptography algorithms, Logical operation, XOR, XNOR.

1. INTRODUCTION

DNA computation is referred as carrying out computations using biological molecules, in place of traditional silicon chips. The idea that individual molecules perhaps could be used for computation dates back to 1959, when American physicist Richard Feynman conferred his ideas on *Nano* technology. However DNA computing was not materially accomplished until 1994, when American computer scientist Leonard Adleman revealed how molecules perhaps be used to answer a computational problem[3]. In DNA computing information is depicted using four character genetic alphabets. They are A [Adenine], G [Guanine], C [Cytosine], T [Thymine] on behalf of the binary alphabets. These four characters are Nucleobases also known as nitrogenous bases or often simply bases

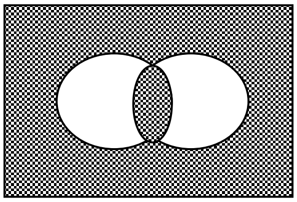
An XNOR gate or Exclusive NOR gate is a digital logic gate with 2 or more inputs and one output that performs logical equality. The output of the XNOR gate is true when both its input were either true or false. If one of its inputs is input is true and another is false the output of the XNOR gate is false.

Table 1: *Truth Table of XNOR gate*

A	B	O/P
F	F	T
F	T	F
T	F	F
T	T	T

iJETRM
International Journal of Engineering Technology Research & Management

Figure 1: *Venn diagram of XNOR gate*

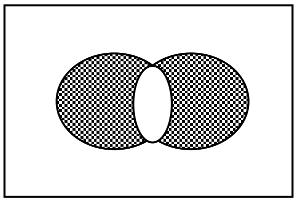


An XOR gate or Exclusive OR gate is a digital logic gate that returns output as true if one, but not both of its inputs are true and otherwise returns output as false.

Table 2: *Truth Table of XOR gate*

A	B	O/P
F	F	F
F	T	T
T	F	T
T	T	F

Figure 2: *Venn diagram of XOR gate*



The XOR Boolean operation is an involuntary function that is if anyone applies XOR once over they get the original plaintext back.

XOR (K, XOR (K, X)) = X

Where X is the plaintext and K is the key. The internal XOR is encryption and the external XOR is the decryption, i.e., the rigorous XOR operation can be used for both encryption as well as decryption[2]. The XOR logical operation is used in encryption of data and is solid to crack even by brute-force method, i.e., generating random encryption keys to match with the correct one.

.

2. RELATED WORK

Even though XOR isn't the only involuntary function or operation in logical operation it is the only one used. This paper discuss about the other important and equally involuntary Boolean operation that is XNOR as a replacement to XOR in the field of cryptography.The XOR operation is used in almost all the common algorithms like DES, Triple DES, AES, IDEA, Twofish, Serpent, RC4, RC5, RC6, etc. This paper made a first attempt as, where the XOR has been utilized in the former noted algorithms So that it can be replaced by XNOR.

Within the Data Encryption Standard, (DES) block cipher itself, the XOR operation is used at 2 different places. First, on the input of S-boxes, XORing 48 bits per round: 48 bitssubkey and 48 bits from the output of expansion. The 48 bits result schemes the eight 6 bits input of the S-boxes. Second, on the output of the S-boxes

iJETRM
International Journal of Engineering Technology Research & Management

XORing 32 bits per round: 32 bits from the former bisection of the state and 32 bits formed by the eight 4 bits output of the S-boxes. The 32 bits result schemes the new bisection of the state.

The Triple DES algorithm also uses XOR obviously since, the operation of Triple DES is based on DES Algorithm and DES uses XOR in the process. In Advanced Encryption Standard, (AES) algorithm the round key will be XORed with the intermediate result and consequently permuted and substituted [5]. International Data Encryption Algorithm, (IDEA) uses XOR operation as one its three main functions which are XOR, addition and multiplication [10]. The Twofish algorithm operates on a block of plaintext residing of 128 bits. There are three main steps in Twofish algorithm, they are i) the block gets divided into four parts ii) then XOR operation will be performed between bit inputs within the key iii) followed by 16 times Feistel network is used to process the input bits [12].

The Serpent algorithm which consists of round function uses XOR for key mixing, 32 parallel utilizations of the same 4*4 S-box, and a linear conversion. Except in the last round, at which point another key mixing XOR replaces the linear conversion [6].

The Rivest Cipher 4 also known as Ron's Code 4 (RC4) uses XOR operation between data bytes and pseudo random byte streams generated from the key. Rivest Cipher 5 (RC5) algorithm which has multiple round function includes bitwise XOR in each round function. The Rivest Cipher 6 (RC6) which can be viewed as a interweaving of two parallel RC5 encryption process followed by an additional extra multiplication operation also uses XOR in its round function [8]. In spite of the fact that almost all symmetric cryptographic algorithms uses XOR operation the asymmetric cryptographic algorithms doesn't uses them. Since asymmetric cryptographic algorithms likeRivest – Shamir – Adleman algorithm,(RSA) and ElGamalalgorithmare based on number theory[9].

Most of the symmetric cryptosystems uses XOR operation for additional strength [11]. XOR can be used to generate pseudo random numbers in hardware; it is also used in error detection, RAID data protection, etc.

The following table 3, gives the various algorithms related to cryptography.

Table 3: *Work related to cryptographic algorithm*

ALGORITHM	**YEAR OF PUBLICATION**	**PRESENCE OF XOR**	**BROKEN USING**
DES	1975	YES	Brute-Force Attack
TRIPLE DES	1998	YES	Sweet32 Attack
AES	1998	YES	Not cracked
IDEA	1991	YES	Meet-in-the-middle Attack
TWOFISH	1998	YES	Not cracked
SERPENT	1998	YES	Meet-in-the-middle Attack
RC4	1987	YES	Fluhrer, Mantin& Shamir Attack
RC5	1994	YES	Differential Attack
RC6	1998	YES	Not cracked

iJETRM
International Journal of Engineering Technology Research & Management

3. ENCRYPTION ALGORITHM USING XNOR

In this paper, a new algorithm is proposed with the contemporary use of XNOR gate.The new algorithm for enciphering textprogresses as follows:

Algorithm :Encrypt_XNOR
Input: Textual message within specific length for encryption.
Output: Encrypted message in nucleo bases for sharing.

```
begin
        letActualInput is the original textual message
        letActualKey is the original 8bit key
        letActualOutput is the enciphered textual message
        function Type[] XNOR_Encry
        begin
        //Conversion of textual message to binary
                begin
                //read string and bit
                        Str1 □  data, bits □  bit;
                        ActualInput = = str1.tocharArray();
                for each i ranging from 0 to max_ length_ of _ActualInput of string do;
                tmpystring□ □ □ integer.tobinary string _of_ ActualInput;
                        tmpyint□  tmpystring.length();
                end for
                if (tmpyint! = = bits) then
                        tmpyint□  bits– tmpyint;
                        if (tmpyint = = bits) then
                                result□ result + tmpystring ;
                        end if
                else
                        if (tmpyint > 0) then
                        for each j ranging from 0 to tmpyint  do;
                                result  □ result + 0;
                        end for
                        result  □ result + tmpystring;
                        end if
                else
                result□ result + tmpystring;
                        end if
                        return binary;
        end
                NewInput□NewInput + result;
                TmpyOutput  = XNOR(NewInput , ActualKey);
                // XNOR  =((NewInput . ActualKey)' + NewInput .ActualKey)
                //Conversion of binary output to enciphered text
        begin
                v□  NewInput;
                        for each n ranging from 1 to max_length of NewInput do;
                                charp□  NewInput.charAt(n);
                                charq□  NewInput.charAt(n + 1);
```

iJETRM
International Journal of Engineering Technology Research & Management

```
                        string v □p + q;
                            if (t = 00) then v = A;
                                if (t = 01) then v = T;
                                    if (t = 10) then v = G;
                                        if (t = 11) then v = T;
                            end if
                    end for
                ActualOutput □ ActualOutput + v;
                return ActualOutput;
        end
    end
```

In the algorithm for encryption using XNOR gate the original message is retrieved from the sender and stored in ActualInput. The ActualKey is the 8-bit key that both the sender and receiver had shared in this symmetric algorithm for encryption.

4. DECRYPTION ALGORITHM USING XNOR

The deciphering of text takes place in the reverse order of enciphering algorithm. Just like this new algorithm the former mentioned algorithm can also be implemented and strengthened using XNOR logical operation.The new algorithm for deciphering text progresses as follows:

Algorithm :Decrypt_XNOR
Input: Encrypted message in nucleo bases
Output: Decrypted original textual message

```
    begin
        letNucleoInput is the original encrypted message
        letActualKey is the original 8bit key
        letActualOutput is the deciphered textual message
        function Type[] XNOR_Decry
            begin
             Read NucleoInput □data
            v = NucleoInput
                for each n ranging from 1 to max_length of NucleoInput do;
                    if (v = A) then t[i] = 00;
                        if (v = T) then t[i] = 01;
                            if (v = C) then t[i] = 10;
                            if (v = G) then t[i]= 11;
                                    t[i]□ □ □   t[i+1]
                end for
             TmpyInput □ □ TmpyInput + t[i]
            end
            begin
                z= XNOR(TmpyInput, ActualKey)
            // XNOR  =((TmpyInput . ActualKey)' + TmpyInput .ActualKey)
            for each j ranging from 0 to max_ length_ of _z string do;
                tmpystring□ □ (char)Integer.parseInt string_z;
                 ActualOutput □ ActualOutput + tmpystring.length;
                end for
            end
```

iJETRM
International Journal of Engineering Technology Research & Management

26. end

In the algorithm for decryption using XNOR gate the encrypted message is stored in NucleoInput. The ActualKey is the 8bit key that both the sender and receiver had shared in this symmetric algorithm for decryption. The resultant original message is stored in the ActualOutput.

5. CONVERSION USING A SAMPLE

The algorithm Encry_XNOR and Decry_XNOR takes the following steps in order to secure the text.
STEP 1: In the Encry_XNOR algorithm let us assume the given input is “ POTENTIAL ALGORITHM ”.
STEP 2: The above plain text message is the converted into its corresponding binary value. The converted plaintext will turn out to be like the following:
0101000001001111010101000100010101001110010101000100100101000001010011000010000001000001010011000100011101001111010100010100100101010100010010000100110 1
STEP 3: The above binary digits are then XNORed together with the key value 00110101.
STEP 4: The XNORed result will look like this:
1001101010000101100111101001111111000010010011110100000111000101110000110111010101000101110000
1101000110110000101100110111000001110011110100000101000001 11
STEP 5: The XNORed result is then converted into the nucleo base value that has been specified in the loop that is
A = 00
T = 01
C = 10
G = 11
STEP 6: After STEP 5, the enciphered textual message will be the result.
CTCCCATTCTGCCTGGCATACTGCCAAGCACGCATCGCCCCACGCATCCAGTCATTCTCGCAAGCTGCC
AACCATG
STEP7: From the Decry_XNOR with the input as
CTCCCATTCTGCCTGGCATACTGCCAAGCACGCATCGCCCCACGCATCCAGTCATTCTCGCAAGCTGCC
AACCATG
we can calculate its plaintext.
STEP 8: The enciphered text is converted into binary data by using the nucleo base value from STEP 5. The result of this process will appear like the following:
1001101010000101100111101001111111000010010011110100000111000101110000110111010101000101110000
110100011011000010110011011100000111001111010000010100000111
STEP 9: The above result is then XNORed with the same key value as encryption, 00110101.
This will give the result as
0101000001001111010101000100010101001110010101000100100101000001010011000010000001000001010011
10001000111010011110101000101001001010101000100100001001101
STEP 10: The above result is then converted into text by making use of the ASCII values.
The output of this process is the original text that is “ POTENTIAL ALGORITHM ”
Like the above the algorithm is tested for 80 words and decrypted contents are verified. The algorithm given a 100 percent accuracy.

6. CONCLUSION

As the saying goes, a chain is as strong as the weakest link. The securities of the digital information rely totally on the exposed vulnerabilities. Having an element of surprise that is XNOR instead of XOR within the algorithm will provide defense for the user against attacks. This new method of implementing XNOR operation instead of XOR in other algorithms will give rise to new set of cryptographic systems.

iJETRM
International Journal of Engineering Technology Research & Management

7. REFERNCES

[1] Adleman, L., "*Molecular computation of solutions to combinatorial problems*". Science, vol.266, pp.1021 - 1029, 1994.
[2] Michael Lewin, "*All about XOR*". Overload Journal, vol.109, June 2012.
[3] Martyn Amos, "*https://www.britannica.com/search?query=dna+computing*".
[4] "*https://www.electronics-tutorials.ws/logic/logic_8.html*".
[5] Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani, "*New Comparative Study between DES, 3DES and AES within Nine Factors*". Journal of computing, vol.2, issue 3, March 2010.
[6] Eli Biham, Ross Anderson, and Lars Knudsen, "*Serpent: A New Block Cipher Proposal*". Springer-Verlag Berlin Heidelberg,LNCS 1372, pp. 222, 1998.
[7] "*https://www.technologyreview.com/2000/05/01/236306/dna-computing/*"
[8] Sourabh Chandra, Siddhartha Bhattacharyya, SmitaPaira, SkSafikulAlam, "*A Study and Analysis on Symmetric Cryptography*". International Conference on Science, Engineering and Management Research,978-1-4799-7613, 2014.
[9] *https://crypto.stackexchange.com/questions/33065/is-all-of-encryption-based-on-xor*.
[10] Xuejia Lai and James L. Massey, "*A Proposal for a New Block Encryption Standard*".Eurocrypt,pp. 389-404, 1990.
[11] Don Coppersmith, Chris Holloway, Stephen M Matyas and NevZunic, "*The data Encryption Standard*" Information Security Technical Report, vol. 2, pp. 22-24, 1997.
[12] *"https://en.wikipedia.org/".*