CS641: Level 7(Bonus)

28/04/2021

Cipher text :

606160E8000000008080820269656CE9E1EBEBEE69656CE9818A8B0669656CE9
010A090400000000818A8B0669656CE9616B69EC00000000606160E800000000

Encryption method :

Toy version of $SHA3$ which has only three step mappings: θ, π, χ, maximum length of the password is 16 characters.
Also we are given the code for implementation of this encryption. From that we know that the message is converted to bits and then filled in and array named "$state$" of size $[5, 5, 64]$. This array then undergoes 24 rounds of θ, π, χ functions. Let R be one round, A be the state array then:

$$R(A) = \chi(\pi(\theta(A)))E = R \circ R \circ \ldots \text{(24 times)} \cdots \circ R(A)$$

θ **function:** In state A we have 5 blocks each with 5 rows and 64 columns. For this function we need an array $c[5, 64]$ to save the XORs of all the columns in state.

$$c[i, k] = \bigoplus_{j=1}^{5} A[i, j, k]$$

$$\theta(A)[i, j, k] = A[i, j, k] \oplus (c[(i + 4)\%5, k] \oplus c[(i + 1)\%5, k])$$

π **function:**

$$\pi(A)[j, ((2 * i) + (3 * j))\%5, k] = A[i, j, k]$$

χ **function:**

$$\chi(A)[i, j, k] = A[i, j, k] \oplus (\neg A[i, (j + 1)\%5, k] \ \& \ A[i, (j + 2)\%5, k])$$

After the encryption, the first 512 bits of the state E are converted to there hex values 4 bits at a time printing 128 hex characters.

Inverse of the functions:
$\theta^{-1} function$: We know that XOR function inverts itself so we tried some iterations of θ function on the state A till we get A back for around 1000 random matrices and found that

$$A = \theta \circ \theta \circ \theta \circ \theta \circ \theta \circ \theta(A)\theta^{-1}(A) = \theta \circ \theta \circ \theta \circ \theta \circ \theta(A)$$

$\pi^{-1} function$:

$$\pi^{-1}(A)[i, j, k] = A[j, ((2 * i) + (3 * j))\%5, k]$$

$\chi^{-1} function$: For this we found that this function applies on columns of thet state A and the there is a

unique mapping for every 5 bit column value. So, we map this and get the inverse value for all columns to get the inverse of this.

00000	:	00000
00001	:	00101
00010	:	01010
00011	:	01011
00100	:	10100
00101	:	10001
00110	:	10110
00111	:	10111
01000	:	01001
01001	:	01100
01010	:	00011
01011	:	00010
01100	:	01101
01101	:	01000
01110	:	01111
01111	:	01110
10000	:	10010
10001	:	10101
10010	:	11000
10011	:	11011
10100	:	00110
10101	:	00001
10110	:	00100
10111	:	00111
11000	:	11010
11001	:	11101
11010	:	10000
11011	:	10011
11100	:	11110
11101	:	11001
11110	:	11100
11111	:	11111

For every column x in A we get the inverse $\pi^{-1}(x)$ and store it in the corresponding place.

Decryption:

First we tried encrypting different strings of different lengths and observed the pattern of state array E. Also we observed that the pattern of chunk of zeros appearing in the given encrypted text is the same as that of strings of length 12. After that we observe the similarity pattern of first 128 hex characters and and rest of the state E when converted to hex and observed that the chunks 16 characters (64 bits) is being copied at places, like for string "lovestarfish" we have:

Encrypted text :S[8] = '6269E46000000000', '0106020AECE268E4', '676FEE6BECE268E4', '05060A0BECE268E4', '0400080100000000', '05060A0BECE268E4', '6669EC6100000000', '6269E46000000000'

E as hex :T[25]= '6269E46000000000', '0106020AECE268E4', '676FEE6BECE268E4', '05060A0BECE268E4', '0400080100000000', '05060A0BECE268E4', '6669EC6100000000', '6269E46000000000', '0000000000000000', '636FE66AECE268E4', '0000000000000000', '6269E46000000000', '636FE66AECE268E4', '0400080100000000', '0000000000000000', '05060A0BECE268E4', '05060A0BECE268E4', '0400080100000000', '0000000000000000', '0106020AECE268E4', '0106020AECE268E4', '676FEE6BECE268E4', '676FEE6BECE268E4', '676FEE6BECE268E4', '0400080100000000'

Also T[10] = '636FE66AECE268E4' = '6269E46000000000' + '0106020AECE268E4' = T[1] + T[2]

So we transformed the encrypted text in the format similar to transforming S to T, so the encrypted text becomes :

$E_{hex}[25]$: '606160E800000000', '8080820269656CE9', 'E1EBEBEE69656CE9', '818A8B0669656CE9', '010A0904000000
'818A8B0669656CE9', '616B69EC00000000', '606160E800000000', '0000000000000000', 'E0E1E2EA69656CE9',
'0000000000000000', '606160E800000000', 'E0E1E2EA69656CE9', '010A090400000000', '0000000000000000',
'818A8B0669656CE9', '818A8B0669656CE9', '010A090400000000', '0000000000000000', '8080820269656CE9',
'8080820269656CE9', 'E1EBEBEE69656CE9', 'E1EBEBEE69656CE9', 'E1EBEBEE69656CE9', '010A090400000000'

We also observed that the state E is transformed to E_{hex} by reversing the order of chunks of 4 bits, So using that to convert E_{hex} to E we have

$$
E = \begin{matrix}
01100000011010000110000001110001000000000000000000000000000000000 \\
00010000000100000001010000000100011010010110101001100011011111001 \\
01111000011111010111110101110111011010010110101001100011011111001 \\
00011000000101010001110100000110011010010110101001100011011111001 \\
00001000000001010000100100000010000000000000000000000000000000000 \\
\\
00011000000101010001110100000110011010010110101001100011011111001 \\
01101000011011010110100101110011000000000000000000000000000000000 \\
01100000011010000110000001110001000000000000000000000000000000000 \\
00000000000000000000000000000000000000000000000000000000000000000 \\
01110000011110000111010001110101011010010110101001100011011111001 \\
\\
00000000000000000000000000000000000000000000000000000000000000000 \\
01100000011010000110000001110001000000000000000000000000000000000 \\
01110000011110000111010001110101011010010110101001100011011111001 \\
00001000000001010000100100000010000000000000000000000000000000000 \\
00000000000000000000000000000000000000000000000000000000000000000 \\
\\
00011000000101010001110100000110011010010110101001100011011111001 \\
00011000000101010001110100000110011010010110101001100011011111001 \\
00001000000001010000100100000010000000000000000000000000000000000 \\
00000000000000000000000000000000000000000000000000000000000000000 \\
00010000000100000001010000000100011010010110101001100011011111001 \\
\\
00010000000100000001010000000100011010010110101001100011011111001 \\
01111000011111010111110101110111011010010110101001100011011111001 \\
01111000011111010111110101110111011010010110101001100011011111001 \\
01111000011111010111110101110111011010010110101001100011011111001 \\
00001000000001010000100100000010000000000000000000000000000000000
\end{matrix}
$$

Now to decrypt E we used inverse functions of the encryption functions for 24 rounds to get A(the original message or password):

$$A = [(\theta^{-1} \circ \pi^{-1} \circ \chi^{-1}) \dots \textbf{(24 times)}](E)$$

After this we get first two rows of

$$A$$

as the input password and rest rows consist of zeros:

$$A[1:2] = \begin{matrix} 01110000011110000111010001110101011010010110101001100011011111001 \\ 01101000011011010110100101110011000000000000000000000000000000000 \end{matrix}$$

Taking 8 bits chunks of this and converting them to ASCII we get:

$$Password = \text{'pxtuijcyhmis'}$$