

CS427

HW 1

Zongyan Lin.

1.

1.4 No. Bitwise-AND does not a good encryption method.

Compare with Bitwise-KoR.

The AND operator has 25% create 0
75% create 1

75% Create 1.

$$\begin{array}{r} 0 \& 0 = 0 \\ 0 \& 1 = 0 \\ 1 \& 0 = 0 \\ 1 \& 1 = 1 \end{array}$$

But Bitwise-XOR has equal possibility to generate either 0 or 1 uniformly.

So, we say otp using XOR operation is special.

And AND-operation does not a good choice

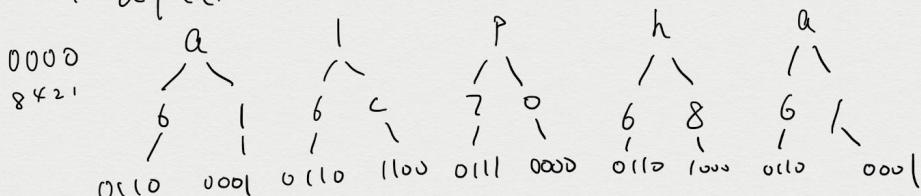
for encryption.

$$\begin{array}{rcl} 0 \oplus 0 & = & 0 \\ 0 \oplus 1 & = & 1 \\ 1 \oplus 0 & = & 1 \\ 1 \oplus 1 & = & 0 \end{array}$$

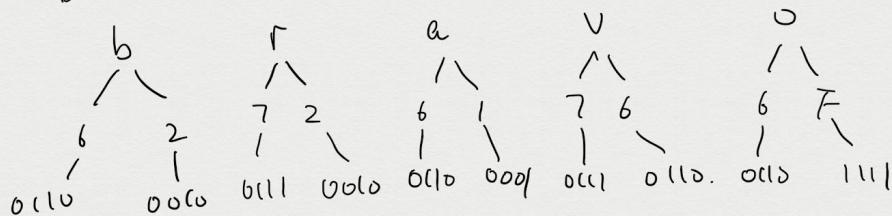
$$\begin{aligned}
 & \text{C}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \\
 & \oplus \quad \text{C}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \\
 & \hline
 & \text{C}_1 \oplus \text{C}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}
 \end{aligned}$$

Case 7

m. = alpha..



M₂ = bravo.



$$\text{So. } m_1 \oplus m_2 = \begin{array}{c} 0110 \quad 0001 \quad 0110 \quad 1100 \quad 0111 \quad 0000 \quad 0110 \quad 1000 \quad 0110 \quad 0001 \\ \oplus \quad 0110 \quad 0010 \quad 0111 \quad 0010 \quad 0110 \quad 0000 \quad 0111 \quad 0110 \quad 0110 \quad 1111 \\ \hline 0000 \quad 0011 \quad 0001 \quad 1110 \quad 0001 \quad 0001 \quad 0001 \quad 1110 \quad 0000 \quad 1110 \end{array}$$

$\therefore m_1 \oplus m_2 = c_1 \oplus c_2 \text{ when } \begin{cases} m_1 = \text{alpha} \\ m_2 = \text{bravo} \end{cases}$

Case 2.

$m_1 = \text{delta}$

$$\begin{array}{ccccc} d & e & f & t & a \\ / \backslash & / \backslash & / \backslash & / \backslash & / \backslash \\ 6 & 4 & 6 & 5 & 6 \\ | & | & | & | & | \\ 0110 & 0100 & 0110 & 0101 & 0110 \end{array}$$

$$\begin{array}{ccccc} & & & & \\ 7 & 4 & & & 6 \\ | & | & & | & | \\ 0111 & 0100 & & 0110 & 0001 \end{array}$$

$m_2 = \text{gamma}$.

$$\begin{array}{ccccc} g & a & m & m & a \\ / \backslash & / \backslash & / \backslash & / \backslash & / \backslash \\ 6 & 7 & 6 & 1 & 6 \\ | & | & | & | & | \\ 0110 & 0111 & 0110 & 0001 & 0110 \end{array}$$

$$\begin{array}{ccccc} & & & & \\ 6 & 1 & & 6 & 1 \\ | & | & & | & | \\ 0110 & 1101 & & 0110 & 0001 \end{array}$$

$$\text{So. } m_1 \oplus m_2 = \begin{array}{c} 0110 \quad 0100 \quad 0110 \quad 0101 \quad 0110 \quad 1100 \quad 0111 \quad 0100 \quad 0110 \quad 0001 \\ \oplus \quad 0110 \quad 0111 \quad 0110 \quad 0001 \quad 0110 \quad 1101 \quad 0110 \quad 1101 \quad 0110 \quad 0001 \\ \hline 0000 \quad 0011 \quad 0000 \quad 0100 \quad 0000 \quad 0001 \quad 0000 \quad 1101 \quad 0000 \quad 0000 \end{array}$$

$c_1 \oplus c_2 = 0000\ 0011\ 0001\ 1110\ 0001\ 0001\ 0001\ 1110\ 0000\ 1110$
 $\therefore m_1 \oplus m_2 \neq c_1 \oplus c_2$ when $m_1 = \text{delta}$
 $m_2 = \text{gamma}.$

Hence. The plaintext alpha and bravo. take the same value by xor gate calculation.

And plaintext delta and gamma doesn't.

So The correct encryption messages are.

$$m_1 = \text{alpha}, \quad m_2 = \text{bravo}.$$

Based on this, the encryption key.

$$\begin{aligned}
 k = m_1 \oplus c_1 &= 0110\ 0001\ 0110\ 1100\ 0110\ 0000\ 0110\ 1000\ 0110\ 0000\ 1 \\
 &\oplus 1111\ 1001\ 0111\ 1001\ 1100\ 1100\ 0001\ 0111\ 1000\ 0110 \\
 \text{key} \rightarrow & 1001\ 1000\ 0001\ 0101\ 1011\ 1100\ 0111\ 1111\ 1110\ 0111
 \end{aligned}$$

$2.8. \quad K = \{1, \dots, 9\}$. KeyGen: $K \leftarrow \{1, \dots, 9\}$. Enc(k, m):
 $M = \{1, \dots, 9\}$. return k return $(k \times m) \% 10$.
 $C = \mathbb{Z}_{10}$.

Based on two One-time Secrecy libraries:

we say.

Σ
 \mathbb{Z}_{ots-L}
 $\text{EAVESDROP}(m_L, m_R \in \Sigma, M)$.
 $K \leftarrow \{1, \dots, 9\}$.
 $C \leftarrow (K \times m_L) \% 10$
return C .

Σ
 \mathbb{Z}_{ots-R}
 $\text{EAVESDROP}(m_L, m_R \in \Sigma, M)$.
 $K \leftarrow \{1, \dots, 9\}$.
 $C \leftarrow (K \times m_R) \% 10$
return C .

And we construct a program A.

A.
 $C \leftarrow \text{EAVESDROP}(1, 2 \in \Sigma, M)$.
return $C = ?$

Based on the key and message. we get possibility table:

m_L	1	2	3	4	5	6	7	8	9
1	11	22	33	44	55	66	77	88	99
2	22	44	66	88	100	122	144	166	188
3	33	66	99	122	155	188	211	244	277
4	44	88	122	166	200	244	288	322	366
5	55	100	155	200	255	300	355	400	455
6	66	122	188	244	300	366	422	488	544
7	77	144	211	288	355	422	499	566	633
8	88	166	244	322	400	488	566	644	722
9	99	188	277	366	455	544	633	722	811

$$\text{So, } \Pr[A \diamondot \mathcal{L}_{\text{ots-L}}^\Sigma \Rightarrow 1] = 1/9.$$

$$\Pr[A \diamondot \mathcal{L}_{\text{ots-R}}^\Sigma \Rightarrow 1] = 0/9.$$

$$\text{Which } \Pr[A \diamondot \mathcal{L}_{\text{ots-L}}^\Sigma \Rightarrow 1] \neq$$

$$\Pr[A \diamondot \mathcal{L}_{\text{ots-R}}^\Sigma \Rightarrow 1].$$

So, for pick message $m_L=1, m_R=2$

which comes from secure scheme.

They demonstrate different possibilities

by constructing program A when linked to the two libraries from

ots,

This is contradicted with the definition of ots. which the ciphertext has same distribution possibilities for any input from secure scheme.

So, we conclude this encryption algorithm does not satisfy the definition of one-time secrecy.