02 (03) September, 2024

Substitution Techniques

- **One-Time Pad (based on Vernam Cipher)**

Transposition Techniques

- **Rail Fence Cipher**
- **Row Transposition Cipher**

(mostly taken from William Stallings, *Cryptography and Network Security: Principles and Practices*, 7ed.)

Substitution Techniques

One-Time Pad:

The One-Time Pad is based on the Vernam cipher. Within the Vernam cipher, we apply an XOR on the letters of the *plaintext*, one by one, with the letters of the *key*. The key does not necessarily have to be of the same length as the plaintext, so if this is the case, we simply repeat the key again and again, until the key is of the same length as the plaintext.

The plaintext "OAK", when used with key "SON", will lead to the ciphertext "COH". Let us see how this happens:

If we map each letter to its corresponding decimal value, A corresponds to 0, B to 1, and so on, until finally Z maps to 25. So, let us examine "O", the first letter of our plaintext, and "S", the first letter of our key.

Letter	O	S
Dec	14	18
Binary	01110	10010

The XOR of 01110 (i.e., "O") and 10010 (i.e., "S") is: 11100 (we have to apply modulo 26, so that the answer remains within the range of 0-25. 11100 equals 28 (decimal), 28 mod 26 equals 2. Thus, our ciphertext is "C" (C maps to decimal 2).

So, for OAK and SON, the ciphertext comes to COH.

The One-Time Pad requires that the key is generated randomly, that the key is of the same length as the plaintext, and that for every new plaintext, we create a new key.

The One-Time Pad provides ***perfect secrecy***, perfect secrecy indicates that the ciphertext reveals no information about the corresponding plaintext other than its length. Naturally, it is very difficult to break, however it is not easy to distribute a key of the same size of the plaintext in a secure manner.

Transposition Techniques

Rail Fence Cipher:

A transposition cipher is one in which plaintext symbols are rearranged (i.e., transposed or permuted) to produce ciphertext. The method of transposition may be either mathematical or typographical in nature (source).

With the Rail Fence Cipher, we assume some depth level x, and we simply write the plaintext in a zig-zag (or diagonal) form across x rows. Finally, to get the ciphertext, we read the text in rows.

Example:

Plaintext: *neso academy is the best*
Depth: 2
Assuming we want to remove the whitespaces.

n		s		a		a		e		y		s		h		b		s	
	e		o		c		d		m		i		t		e		e		t

CIPHERTEXT: *NSAAEYSHBEOCDMITEET*

Another Example:

Plaintext: *thank you very much*
Depth: 3
Assuming we want to remove the whitespaces.

t				k				v				m			
	h		n		y		u		e		y		u		h
		a				o				r				c	

CIPHERTEXT: *TKVMHNYUEYUHAORC*

Cryptanalysis:

How can we go about the decryption of a ciphertext received, when we know that the encryption method was Rail Fence? Naturally, we need to know the depth level. If we know the depth level and the length of the ciphertext, we can build a matrix (the number of rows will equal the depth level, whereas the number of columns will equal the length of the ciphertext.

So if we know that the depth level is 3, and the number of characters in the ciphertext is 16, we can build a 3*16 matrix. Next, we can start filling in the initial letters, into the possible slots of the first row, assuming that the original plaintext was written in zig-zag (diagonal manner).

t				k				v				m			
	*		*		*		*		*		*		*		*
		*				*				*				*	

After consuming the first four letters (*TKVM*), we can see there is no space left in row-1, so now the next letters must be placed in row-2, i.e., letters *HNYUE…*

t				k				v				m			
	h		n		y		u		e		y		u		h
		*				*				*				*	

After consuming the next eight letters (*HNYUEYUH*), we can see there is no space left in row-2, so now the next letters must be placed in row-3, i.e., letters *AORC*

t				k				v				m			
	h		n		y		u		e		y		u		h
		a				o				r				c	

After reading the existing text in zig-zag (or diagonal), we can clearly see the original plaintext "THANKYOUVERYMUCH".

Row Transposition Cipher:

With the row transposition cipher, we will use a key to encrypt a plaintext. The way we do this is by first writing out the plaintext in separate rows. We draw a matrix of N columns, where N is the length of our key. Let us assume that the plaintext is "winter is coming", and the key is "hack". The length of "hack" is 4, therefore we will build a matrix of 4 columns.

But we will transform our key into numeric format, by rearranging the letters into the correct lexicographic order. For example, HACK is 1234, H maps to 1 (first position), A maps to 2, C maps to 3, and K maps to 4. However, their lexicographic order is 2314, i.e., ACHK. Now we will use 2314 (our key) a bit later.

So, after filling the four column matrix, we have (note, we are maintaining whitespaces, this is optional):

W	I	N	T
E	R	_	I
S	_	C	O
M	I	N	G

I have used underline for representing whitespaces. Now we will write down all the text column by column, and in order of column 2, 3, 1, 4 (i.e., according to our key). So, this leads to:

IR_I   N_CN   WESM   TIOG → together, this becomes IR_IN_CNWESMTIOG

This is our final ciphertext.

Try using Row Transposition Cipher (rowcipher.netlify.app) to see if plaintext "This life is short", with key, quick, maps to SES_IF_T__H_TLIOHISR.