CSCI361 - Cryptography & Secure Application

One-time Pad

ONE-TIME PAD

One-time-pad is known to provide the *perfect security*. What does the term *perfect security* mean?

It refers to the security provided by one-time pad. Since one-time-pad produces random output that bears no statistical relationship to the plaintext, and since the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code, and therefore it is said to be perfectly secured.

One-Time Pad

One possible implementation of “One-time pad” is to generate a random key sequence of the same length as message and encrypt the message using the cipher $C_i = M_i + K_i \ (mod\ 26)$, where K_i are random key characters and $M = K = C = \mathbb{Z}_{26}$. One of the recommendations for a proper use of “One-time pad” and to ensure perfect secrecy is to never reuse the same key for encryption of two different messages. Your task is to explain how only the knowledge of two different ciphertext sequences $C = C_1 C_2 \ldots C_n$ and $C' = C_1' C_2' \ldots C_n'$, obtained by applying the same secret key, can compromise the security of One-time pad.

ONE-TIME PAD

Suppose the two messages are encrypted using the same key as follow:

$$C_i = M_i + K_i (mod\ 26) \quad \ldots\ldots eq1,$$
$$C_i' = M_i' + K_i (mod\ 26) \quad \ldots.\ldots eq2$$

By solving the two equations, we have $M_i - M_i' = C_i - C_i' (mod\ 26)$ for $i = 1,2, \ldots, n$. From here, we can see that if we know C_i and C_i' we can find $M_i\ and\ M_i'$.

Alternatively, we can also apply known plaintext attack.

One-Time Pad

Alice and Bob wish to communicate using the one-time pad encryption system. Alice wishes to encrypt the following message:

i pass csci361

and send it to Bob. Prior to sending the communication, Alice and Bob agreed on a one-time pad key 2, 4, 6, 9, 1, 23, 7, 1, 5, 17, 1, 3, 10, and 11. These keys are numerical equivalent of alphabetical symbols that they agree upon. The numerical equivalent of the alphabetic characters and symbols are as follow:

One-Time Pad

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9
18	19	20	21	22	13	24	25	26	27	28	29	30	31	32	33	34	35

Space
36

ONE-TIME PAD

1. Encrypt the message using the one-time pad key as discussed during lecture and tutorial.

ONE-TIME PAD

Encryption:

- Convert the message (plaintext) to its numeric equivalent as follow:

i		**p**	**a**	**s**	**s**		**c**	**s**	**c**	**i**	**3**	**6**	**1**
8	36	15	0	18	18	36	2	18	2	8	29	32	27

- Encrypt the numeric equivalent message by adding the corresponding number (character by character) to the key and reduce mod 37 as follow:

ONE-TIME PAD

Plain text:	**8**	**36**	**15**	**0**	**18**	**18**	**36**	**2**	**18**	**2**	**8**	**29**	**32**	**27**
Key:	2	4	6	9	1	23	7	1	5	17	1	3	10	11
Plaintext + key (mod 37):	10	3	21	9	19	4	6	3	23	19	9	32	5	1

For example:

- 8 + 2 (mod 37) = 10 mod 37
- 36 + 4 (mod 37) = 40 mod 37 = 3 mod 37
- Etc.

One-Time Pad

- Next we convert the numeric encrypted message into alphabetic ciphertext, and we have

Numeric encrypted text:	10	3	21	9	19	4	6	3	23	19	9	32	5	1
Ciphertext:	k	d	v	j	t	e	g	d	x	t	j	6	f	b

One-Time Pad

2. Charlie intercepts the ciphertext in (1), and he also discovers some of the number of the one-time pad key – 2, 4, ?, ?, ?, ?, 7, 1, 5, 17, 1, 3, 10, 11. Charlie makes some guesses, and come out with the four missing numbers for the one-time pad key as follow:

2, 4, 14, 9, 0, 0, 7, 1, 5, 17, 1, 3, 10, 11

Decrypt the ciphertext in (1) using the key Charlie has guessed.

ONE-TIME PAD

- First, convert the ciphertext to its numeric equivalent as follow:

Ciphertext:	k	d	v	j	t	e	g	d	x	t	j	6	f	b
Numeric encrypted text:	10	3	21	9	19	4	6	3	23	19	9	32	5	1

- Decrypt the numeric equivalent message by subtracting the corresponding number from the key and reduce mod 37 as follow:

Numeric encrypted text:	**10**	**3**	**21**	**9**	**19**	**4**	**6**	**3**	**23**	**19**	**9**	**32**	**5**	**1**
Key:	2	4	14	9	0	0	7	1	5	17	1	3	10	11
Encrypted text – key (mod 37):	8	36	7	0	19	4	36	2	18	2	8	29	32	27

ONE-TIME PAD

- Next, we convert the numeric message into alphabetic plaintext, and we have:

Encrypted text – key (mod 37):	**8**	**36**	**7**	**0**	**19**	**4**	**36**	**2**	**18**	**2**	**8**	**29**	**32**	**27**
Plaintext:	i		h	a	t	e		c	s	c	i	3	6	1

ONE-TIME PAD

3. What implication or deduction that can be drawn from the encryption and decryption processes in (1) and (2) with regards to the strengths of an one-time pad cipher?

From the encryption and decryption processes done above, we can conclude that different one-time pad key may lead to two different intelligible messages when decrypting. One-time-pad produces random output that bears no statistical relationship to the plaintext. This means that a brute-force attack on one-time pad will never work because many different intelligible messages can be obtained. For example, i hate csci361, i like csci361, i miss csci361, i fail csci361, i love csci361, etc. This conclusion justify that one-time pad cipher is perfectly secure, provided it is used correctly; that is, the one-time pad key is random, and the key is never repeated once used.

ONE-TIME PAD

If one-time pad cipher is perfectly secure, why not using one-time pad cipher then? What are the problems / issues with one-time pad cipher?

1. There is the practical problem of making large quantities of random keys. Supplying truly random characters in large quantities is a significant task.
2. The second problem is the distribution of this huge key. For every message to be sent, a key of equal length is needed by both sender and receiver. Hence, a mammoth key distribution problem exists.