Group C

Encryption Method Report

Math 485

[redacted]

September 2015

Dear OCRAI:

IMC is a global leader in the creation of cryptosystems to create secure communication methods and we are glad to assist you. As your organization has learned in an unfortunate matter, without a secure cryptosystem to communicate, it is very easy for information to be captured or misplaced. Our team has created a custom protocol with which to secure your information so that it can easily be encrypted, transmitted or stored, and then decrypted as necessary. Initially, we felt that we might attempt a system using a variation of a cipher on the English alphabet but we then developed the method described further in this letter that allows for information to be encrypted with only a doubling of the message size and using a 10 digit key. We feel that this system will be most secure and useful as OCRAI conducts daily business operations.

In Kerckhoffs' La Cryptographie Militaire, he discusses that the security of any cryptographic system should be based on the key and not on the obscurity of the encryption algorithm. (Kerckhoffs 1883) This was one of the main ideas that guided our creative process of inventing an encryption algorithm.

Our idea is a simple one that simply involves a process of scrambling the order of the plaintext, mixed with random dummy text. Below we outline the algorithm involved in encryption, after which an example of encryption will be provided.

Encryption

1. Create a plaintext message which has 140 characters or fewer.
2. Randomly generate a 10-digit key using alphanumeric characters (0-9).
3. To ensure security, the plaintext message must be padded to reach a character length of 150. This padding is accomplished by adding randomly generated alphabetic characters after the plaintext and QQ to denote the end of your plaintext portion and the beginning of the padding text.
4. Randomly generate dummy text that is equivalent in length to your plaintext message.

5. Assign each letter of your plain, and dummy, text a number (0-25) based on its position in the alphabet.
6. Add the numerical values of your dummy text to the numerical values plaintext to obtain a new, obfuscated, plaintext document.
7. Insert the dummy text in the even positions of your plaintext, thus every character of the original message is now separated by a character of the dummy text. Note: the first character of this obfuscated text should be a character of your plaintext, not the dummy text.
8. The text is then broken into 10 character blocks. This is another reason why padding text needs to be added after our plaintext message, to reach a multiple of 10 for the length of the entire text to encrypt.
9. The first n characters of each block are then moved from the front of the block to the back of the block. n is determined by the integer value of the first character of the key.
10. The first character of the entire text will then be moved from the front to the back. Effectively shifting the text by one character to enhance security and ensure the the 10 character blocks are not the same for the second scrambling.
11. After this first character shift, separate the text again into 10 character blocks and repeat steps 7 and 8, however this time using the second integer value of the key to determine the number of characters that are moved from the front of the 10 digit block, to the back of the same block.
12. Continue this process until all of the values of the key have been used in scrambling the blocks

Example

1. Plaintext: “Hello I am currently studying math” length = 29, remove spaces.
2. Key: 4837291021
3. Dummy text: “aptent taciti sociosqu seesguita” length = 29, remove spaces.
4. Add QQ and padding text at the end of your plain text to reach 150 characters, thereby increasing security.
    a. Therefore your plaintext would now be “...studying mathQQajdfinaklisd…”
    b. for simplicity and readability, the padding has not been included in this example.

5. Plaintext value, using (0-25) for (a-z) values:

“7-4-11-11-14-8-0-12-2-20-17-17-4-13-19-11-24-18-19-20-3-24-8-13-8-12-0-19-7...”

Dummy text value:

“0-15-19-4-13-19-19-0-2-8-19-8-18-14-2-8-12-18-16-20-18-4-4-18-6-20-8-19-0...”

5. Added for new plaintext value. plain(1) + dummy(1) = new(1) - … - plain(n)+dummy(n) = new(n):

“7-19-30-15-27-27-19-12-4-28-36-25-22-37-21-19-38-36-35-40-21-28-12-31-12-32-8-38-7...”

6. Intersperse new plaintext and dummy text values. p(1), d(1), p(2), d(2), … , p(n), d(n):

“7-0-19-15-30-19-15-4-27-13-27-19-19-19-12-0-4-2-28-8-36-19-25-8-22-18-37-14-21-2-19-8-38-12-36-18-35-16-40-20-21-18-28-4-12-4-31-18-12-6-32-20-8-8-38-19-7-0...”

7. Take the ever 10 characters of the text and separate them into blocks (in this step the QQ has been added to reach a multiple of 10, this would already be included when encrypting a full message interspersed evenly with the dummy text):

“7-0-19-15-30-19-15-4-27-13-27-19-19-19-12-0-4-2-28-8-36-19-25-8-22-18-37-14-21-2-19-8-38-12-36-18-35-16-40-20-21-18-28-4-12-4-31-18-12-6-32-20-8-8-38-19-7-0-16-16...”

8. The first n characters are then moved from the front to the back:

a. for simplicity, the rotation of only one block is shown.

n = 4 (see first value in key)

“7-0-19-15-30-19-15-4-27-13” => “30-19-15-4-27-13-7-0-19-15”

9. This leaves us with the following ciphertext after the first key value (n) has been used to shift ALL of the blocks in the message:

“30-19-15-4-27-13-7-0-19-15-12-0-4-2-28-8-27-19-19-19-22-18-37-14-21-2-36-19-25-8-36-18-35-16-40-20-19-8-38-12-12-4-31-18-12-6-21-18-28-4-38-19-7-0-16-16-32-20-8-8...”

a. We then move the first character to the back of the ciphertext to ensure that the blocks are all different on our next scrambling:

“30-19-15-4-27-13-7-0-19-15-12-0-4-2-28-8-27-19-19-19-22-18-37-14-21-2-36-19-25-8-36-18-35-16-40-20-19-8-38-12-12-4-31-18-12-6-21-18-28-4-38-19-7-0-16-16-32-20-8-8...”

=>

“19-15-4-27-13-7-0-19-15-12-0-4-2-28-8-27-19-19-19-22-18-37-14-21-2-36-19-25-8-36-18-35-16-40-20-19-8-38-12-12-4-31-18-12-6-21-18-28-4-38-19-7-0-16-16-32-20-8-8-30...”

10. We then repeat this process with the second numerical value of the key:

“19-15-4-27-13-7-0-19-15-12-0-4-2-28-8-27-19-19-19-22-18-37-14-21-2-36-19-25-8-36-18-35-16-40-20-19-8-38-12-12-4-31-18-12-6-21-18-28-4-38-19-7-0-16-16-32-20-8-8-30...”

a. (second key value is 8)

“19-15-4-27-13-7-0-19-15-12” => “15-12-19-15-4-27-13-7-0-19”

b. The new ciphertext is then shifted:

“15-12-19-15-4-27-13-7-0-19-19-22-0-4-2-28-8-27-19-19-8-36-18-37-14-21-2-36-19-25-12-12-18-35-16-40-20-19-8-38-4-38-4-31-18-12-6-21-18-28-8-30-19-7-0-16-16-32-20-8...”

=>

“12-19-15-4-27-13-7-0-19-19-22-0-4-2-28-8-27-19-19-8-36-18-37-14-21-2-36-19-25-12-12-18-35-16-40-20-19-8-38-4-38-4-31-18-12-6-21-18-28-8-30-19-7-0-16-16-32-20-8-15...”

11. This process is continued until the entire key has been iterated through once.

Dummy characters need to be added at the end of your plaintext before interspersing it with the dummy text, causing your plain text to be 150 characters in length consistently. After the dummy text has been added to the values of your plaintext, and interspersed, you will have a ciphertext of 300 characters, no matter the initial message size. In addition to this, QQ will be added after the end of your plain text, before the padding text, to ensure easy decryption and additional security.

In conclusion, this method of encryption allows us to rely wholly on the key as our life line of security. While the algorithm is quite obscure, the key ensures that the message is practically impossible to decrypt without knowledge of the key. We feel that this makes the encryption method simple and secure, allowing it to be easily implemented and used regularly.

Sincerely,

IMC

Citations:

Kerckhoffs, A. (1883). *La Cryptographie Militaire.*