Group B

M.A.R an Independent Mathematical Contractor Inc.

Report On Work Performed For OCRAI

Note: This report contains the work done and MAR's solution for the problem given by OCRAI. All work has been done according to the instructions provided by OCRAI. Therefore, this document contains only the information about our company's solution to the specific problem and it should not be applied to any other problems OCRAI might have.

OCRAI's Problem

We, MAR an Independent Mathematical Contractor, received a request for our services from the OCRAI's Vice President of Security on 1 September 2020. On September 3rd, we received a detailed description of the problem that our company has been contracted to solve. On September 8th, we sent OCRAI a confirmation for its request with the information of all the team members that will be working on the problem. We accepted the request with the conditions of complying with all the instructions given by OCRAI and delivering our work by the 11th of September at 5:00 PM. In the same, OCRAI promised full payment (all points) for our service after an inspection of the quality of our work.

OCRAI's problem consists of the use of insecure messages to exchange sensitive company information such as trade secrets. This has caused multiple severe security breaches. The problem originated when OCRAI's employees erroneously used smartphones to communicate and send messages to incorrect recipients. These issues not only represent a threat to OCRAI's profits and future revenue streams, but create a bad image for the company. For this reason, OCRAI requested our services to provide a method for keeping OCRAI data secure.

Our team will provide a nontrivial system of encrypting plaintext English messages. OCRAI will use our system of encryption to encrypt all text messages before they are sent, and decrypt any received messages. OCRAI requested that our solution to its problem should take standard text messages (160 characters) and output ciphertext that can be sent in at most 5 text messages (800 characters). It was also requested that our encryption key should be of limited length due to the limited processing power of mobile devices. Additionally, it was also stated that the ciphertext should be in a form that can be typed with any QWERTY design; this will allow OCRAI's engineers to use our cryptosystem to create a mobile application for its employees. Thus, any messages sent to unauthorized recipients will be unreadable without our encryption system.

How MAR's System of Encryption Works

MAR encryption works by employing the fundamental theorem of arithmetic. Each letter is assigned a unique value between 1-26 as follows:

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

Encryption

First we generate a random string of characters between 5-10 characters. This string serves as the key. The key will be broken up into sections of x letters with the key word repeating as many times as needed to fully encrypt the message. For the following examples, we will split up the key into sections of 4 letters (x=4). This number (x) can be changed as needed. In order to keep the integrity of the key, x must be smaller than the total number of characters in the key and should not be below 4. The following chart shows an example of how the keys will be broken up. For illustration purposes, we will use "tests".

KEY	test	stes	tste	stst	ests	test	stes	tste	stst	ests
Plaintext Character Index	0	1	2	3	4	5	6	7	8	9

As illustrated above, the first character (indexed as 0) of the plaintext will be encrypted using the key "test". The second character (indexed as 1) of the plaintext will be encrypted using the key "stes". Each of these keys can be changed into numerical representations by using the originally stated substitution (A=1, B=2, etc). The following chart shows how the above listed keys would be listed numerically.

KEY	test	stes	tste	stst	ests
Numerically	20, 5, 19, 20	19, 20, 5, 19	20, 19, 20, 5	19, 20, 19, 20	5,19,20,19

This numerical key, that will differ with each letter of the code, will be referred to as the number key. Due to the fundamental theorem of arithmetic, each letter's assigned number has a unique prime factorization. For example, the prime factorization for twenty-two(22) or "V" is 2*11 or 2*11*1*1. Each letter is assigned an x-digit prime factorization based on its

corresponding number. If a number's prime factorization is less than x digits, we fill in with ones (1) until the product reaches x digits. For example two(2) or "B" is 2*1*1*1. We treat one (1) as a series of four ones (1*1*1*1). It is essential that the prime factorization is arranged in order from smallest to largest with the exception of the ones(1). Ones(1) should be at the end of the prime factorization. We will refer to this number as the PTPF (plain text prime factorization)

Each plaintext character is encrypted into x cipher-text characters. Spaces are omitted to further secure and protect this encryption method. The x-character ciphertext is generated using the PTPF and the number key. Each letter in the key is shifted by each number in the plaintext character's prime factorization. The following formula illustrates this:

Let:
x=number of characters in the key (4 in this case)
p=a prime number in the plaintext letter's prime factorization
t=the value of the letter in the number key
PTPF = $p_1p_2...p_x$
Number Key = $t_1t_2...t_x$

A single plaintext character will be encoded into the following set of x numbers

$$((t_1+p_1) - 1) \bmod(26)) + 1, ((t_2+p_2) - 1) \bmod(26) +1, \ldots ((t_x+p_x) - 1) \bmod(26) +1$$

Each number in the above will then be converted to its corresponding letter and the commas will be removed.

This may be best illustrated in the following example for encoding "B" using the string "tests". For this encoding we will assume that "B" is the first letter of the plain text and therefore uses "test" as its code.

Note: B = 2 = 2*1*1*1

	T	E	S	T
Number Key (TEST in this scenario)	20	5	19	20
Corresponding Prime Factor of Plaintext Character	2	1	1	1
(t_x+p_x) - 1 mod(26) +1	22	6	20	21
Ciphertext Character	V	F	T	U

Thus, plaintext B becomes VFTU

Decryption

Decryption requires the key and the ciphertext. To illustrate this, we will decrypt the ciphertext UYFT with the key "stes". This assumes that UYTF are the ciphertext characters for the second plaintext letter. We will reverse the above process.

Let:

x=number of characters in the key (4 in this case)
y=the ciphertext character's number
t=the value of the letter in the number key
Key = $t_1t_2...t_x$

	S	T	E	S
Ciphertext Character	U	Y	F	T
Corresponding numerical value	21	25	6	20
Number Key (STES in this scenario)	19	20	5	19
$(y_x - t_x) - 1 \bmod(26) + 1$	2	5	1	1
Value of Plaintext Primes	2	5	1	1

Since 2*5*1*1 = 10 and 10=J, Ciphertext UYFT decrypts to plaintext J

Note: By convention $(y_x - t_x) - 1 \bmod(26)$ should be between 0 and 25. Add 26 as necessary to make $(y_x - t_x) - 1 \bmod(26)$ between 0 and 25.

Conclusion

Mar's encryption works by employing a fundamental theorem of arithmetic. An original key, a random string of 5-10 characters, will be broken up into sections of four letters with the key word repeating as many times as needed. Each section becomes a key. Each of these keys can be changed into numerical representations by using MAR's index of the English alphabet which is referred to as the number key. Each letter's assigned number has a unique prime factorization. The prime factorization is arranged in order from smallest to largest. If a number's prime factorization is less than four digits, we fill in with ones at the end of the prime factorization arrangement until the elements of the arrangement reaches the number of letters in the original key, this is what we call PTPF. Each plaintext character is encrypted into four cipher-text characters. Spaces are omitted to further secure and protect this encryption method. The four-character ciphertext is generated using the PTPF and the number key. Each letter in the key is shifted by each number in the plaintext character's prime factorization. Finally, the decryption requires the key and the ciphertext and it consists of reversing the process of encryption. This nontrivial method of encrypting plaintext English messages will prevent OCRAI from experiencing the same major security breach again.