Group K

Independent Mathematical Contractors, Inc.
136 TMCB
Provo UT, 84602
September 11, 2020

OCRA Creative Recursive Acronyms, Inc.
385 Primality Way
Provo, UT 84604

Dear Mr. Andrews,

Thank you for your interest in our team's cryptographic capabilities. We understand the serious implications that our contribution has for OCRAI and aim to effectively preserve the integrity of internal communications. As experts in cryptography, we have developed a unique cryptographic scheme that can be used to encode plaintext messages into ciphertext. This ciphertext is unreadable to a typical observer and difficult for a malignant attacker to decode. Decrypting the ciphertext requires knowledge of a special key and application of the appropriate decoding scheme. Without this key, encrypted messages cannot be understood. This encryption scheme is also rather efficient, requiring no more than five times the storage space of the original, unencrypted message. With the help of our method, OCRAI will be able to maintain safe and secure internal communications without worry for accidental or malicious leakage of confidential details.

Our encryption scheme begins with a key. This key is essential to encoding messages to be sent and decoding received messages. Since this system will be implemented in the phone app that OCRAI is developing, we can embed the key in the app itself. The advantage of this embedding is twofold. First of all, the key need not be known to anyone other than the developers who program it into the app; all other employees of OCRAI can enjoy use of our cryptographic method to send and receive encrypted messages without knowledge of the key. Secondly, this also means that the key does not have to be communicated over any channels beyond the internals of the app itself, which is a compiled software whose source code is automatically obfuscated.

The key is a special number that happens to be the product of exactly three prime numbers. According to the fundamental theorem of algebra, every positive integer greater than or equal to 2 is the product of a unique collection of prime numbers. This means that three pieces of information (one contained in each prime number) can be stored in a single key. There are two constraints on the allowed keys: first, the smallest prime factor of the key must be less than or equal to thirty-nine. Second, the next-to-smallest prime factor of the key must not be two or thirteen. These are very light constraints that still allow a large number of possible keys. Some valid example keys are 322 = 2·7·23, 2431 = 11·13·17, and 10759 = 7·29·53. Note that due to the non-triviality of factoring large numbers, it is recommended to use keys smaller than 100,000. This does not significantly limit the encryption capabilities of our system. Once the key has been determined, and its factors are known, we assign to the three factors the names *a*, *b*, and *c*. The numbers *b* and *c* denote two parameters in an affine cipher, which we will describe in the following paragraph.

An affine cipher gives a rule for how to transform plaintext messages into unreadable ciphertext. We begin by converting each letter in the alphabet the corresponding number of its order in a zero-indexed system (i.e. A = 0, B = 1, C = 2, …, Z = 25). We call this number *r*. Once this has been done for each letter in the message, we multiply the number by *b* and then add *c* to the result, which we call *x*. After computing *x*, we take the equivalent number mod 26 (that is, take the remainder of *x* after dividing by 26):

$$x \equiv br + c \pmod{26}.$$

Once this has been done, we know that *x* is some number between 0 and 25, inclusive. A typical affine cipher would simply convert *x* back to the corresponding letter. We take a different approach to utilize the advantage of a longer ciphertext than plaintext. We first multiply *x* by *a*, giving us

$$y = ax\,.$$

After *y* has been computed, it is a number with between one and three digits. We can thus represent it with three letters, one corresponding to each digit. For example, if *y* = 309, we would add the string “DAJ” to the cipher. Separation between letters are represented by a single letter whose assigned number is greater than or equal to ten, i.e. and letter between K and Z, and actual spaces between words are represented by two of such letters. These letters can be randomly decided, which significantly increases the difficulty of decryption through typical methods such as frequency analysis. For example, if two two-letter words resulted in *y* values of 135 and 30 for the first word and 255 and 90 for the second word, the corresponding ciphertext may look like “BDFLDAXQCFFYJF”. Note that the letters “L”, “X”, “Q”, and “Y” in that ciphertext could be replaced by any other letters between K and Z.

Now, since each individual character in the plaintext becomes up to four characters (as *y* is made of three digits; the extra digit comes from the separation letter between individual characters). Thus for a 160-character message, the corresponding ciphertext will be no longer than 640 characters.

To decrypt the message, we need to know the key. Knowing the key, we factor it into its three prime factors, assigning to *a*, *b*, and *c* the values of the smallest, middle, and largest factors. We then go through the message and convert each letter to its corresponding number. Any sequence of two numbers greater than nine is converted to a space, and any single number greater than nine is discarded. We then divide each number by *a* to recover *x*. From here, decoding the message is identical to decoding any affine cipher. We simply need to solve the equations

$$x_i \equiv br_i + c \pmod{26}$$

or

$$x_i - c \equiv br_i \pmod{26}$$

for each r_i, where i denotes the index of the letter. The fact that b is a prime number that is neither 2 or 13 guarantees that we can "divide" by b in these equations by multiplying each side by the number d for which $db \equiv 1 \pmod{26}$. This gives a solution

$$r_i \equiv d(x_i - c) \pmod{26}.$$

Since we know d, c, and x_i, this gives us a sequence of numbers between 0 and 26 that, when mapped back to the corresponding letter in the alphabet, reveals the original plaintext.

We conclude this by presenting a short example. We first select the key 10759 = 7·29·53, so a = 7, b = 29, and c = 53. If our plaintext message is "HELLO THERE", we first convert these characters to the number sequence $R = \{7, 4, 11, 11, 14, (), 19, 7, 4, 17, 4\}$. Each of these numbers represents a distinct r_i. We have noted the presence of a space with (). We then apply the affine transformation by computing $x_i \equiv br_i + c \pmod{26}$ for each of these numbers. This gives us a resulting sequence $X = \{22, 13, 8, 8, 17, (), 6, 22, 13, 0, 13\}$. Next, we multiply each number in this sequence by a to get the sequence $Y = \{154, 91, 56, 56, 119, (), 154, 91, 0, 91\}$. Finally, we replace each digit with its corresponding letter, insert a random letter from K to Z between each letter, and replace the space with two such letters. One possible resulting ciphertext would be "BFEPJBYFGMFGNBBJQWECVBFEYJBSASJB". To decode this ciphertext, we remove all letters between K and Z, inserting a space if we find two in a row: "BFE/JB/FG/FG/BBJ EC/BFE/JB/A/JB". We have inserted a slash between individual letters for clarity, but these will not be kept in the final decrypted message. Replacing each letter with its corresponding numbers and interpreting the result as the digits of up to 3 digit numbers, we recover the sequence Y above. We then divide each of these numbers by a to recover the sequence X. We now need to calculate d by finding the solution to $db \equiv 1 \pmod{26}$. This can be solved using Euclid's algorithm. Once d has been found, we multiply each element in X by d to recover R. Changing R back to the corresponding letters and keeping note of the space, we recover the original message "HELLO THERE".

The security of the internal communications of OCRAI can be effectively established and maintained through our solution. With our system, messages can be securely encrypted, sent, and decrypted in a way that electronic interception will yield a ciphertext that is impossible to decode without knowledge of the key. Having the key implemented in messaging software preserves its privacy, which in turn ensures the effectiveness of our scheme. Since the method is based on a simple affine cipher, it does not require a lot of computational power to encode or decode messages, yet its differences from the cipher ensure that the message cannot be decrypted without knowledge of the key. We firmly believe that the cryptographic system that we have developed and explained will be of great benefit to the advancement of OCRAI.

Sincerely,

[redacted]

Co-founders
IMC