Group Q

MATH 485 - CRYPTO PROJECT

[redacted]

1. Introduction

We are glad to inform you that Independent Math Contractors, Inc has completed the project as described in your previous letter. We have created an entirely new cipher for your exclusive use, to prevent any further embarrassment caused by careless employees. Rest assured we have had our top cryptological math experts work tirelessly on this cryptosystem since receiving your message. The system described herin will encumber any third party's attempt to understand what your internal communication is without the key. Below, we will give an overview of how the system works, and provide a short example of how to encrypt a message.

2. The Cipher

2.1. Theory. Consider the function $f_n : \mathbb{Z}_{26} \to \mathbb{Z}_{26}$ defined by $f_n(x) = x^n$. Since f_n is defined on $\mathbb{Z}_{26}$, we note that we are working (mod 26) in all computations. To obtain a decryptable cipher, we wish to choose values for n that give a bijective function. This will give us a unique mapping from the English alphabet to itself, allowing us to reverse the encryption process. After numerical experimentation, the only values n for which this function is bijective on $\mathbb{Z}_{26}$ are $n = 1, 5, 7, 11$. For $n \in \{1, 5, 7, 11\}$ we see that the values for f are given by

`alphabet`	`A`	`B`	`C`	`D`	`E`	`F`	`G`	`H`	`I`	`J`	`K`	`L`	`M`	`N`	`O`	`P`	`Q`	`R`	`S`	`T`	`U`	`V`	`W`	`X`	`Y`	`Z`
x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$f_1(x) = x^1 (\text{mod } 26)$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$f_5(x) = x^5 (\text{mod } 26)$	0	1	6	9	10	5	2	11	8	3	4	7	12	13	14	19	22	23	18	15	24	21	16	17	20	25
$f_7(x) = x^7 (\text{mod } 26)$	0	1	24	3	4	21	20	19	18	9	10	15	12	13	14	11	16	17	8	7	6	5	22	23	2	25
$f_{11}(x) = x^{11} (\text{mod } 26)$	0	1	20	9	10	21	24	15	18	3	4	19	12	13	14	7	22	23	8	11	2	5	16	17	6	25

We notice that these functions perform a series of transpositions (swapping two elements with each other) that are distinct (we don't swap 2 with 3 and then 3 with 6). For example, note that $f_5(2) = 6$ and $f_5(6) = 2$. So, this gives that $f_n^2(x) = (f_n \circ f_n)(x) = f_n(f_n(x)) = x$ for $n \in \{1, 5, 7, 11\}$. Hence $f_n = f_n^{-1}$. We also note that if we add a constant shift to these functions to get $g(x) = f_n(x) + s$ for some $s \in \mathbb{Z}_{26}$, then the inverse function is given by $g^{-1}(x) = f_n(x - s)$ since

$$(g \circ g^{-1})(x) = g(g^{-1}(x)) = g(f_n(x - s)) = f_n(f_n(x - s)) + s = x - s + s = x$$

and

$$(g^{-1} \circ g)(x) = g^{-1}(g(x)) = g^{-1}(f_n(x) + s) = f_n(f_n(x) + s - s) = f_n^2(x) = x$$

2.2. Key. Our cryptosystem requires a key to be used for encryption and decryption. The basic structure of the key is a tuple of a permutation of $\{1, 2, 3, 4\}$ and a number in the interval $[1, 25]$, concatenated together. So, the key is 5-6 characters long, where the first four digits are a permutation of $\{1, 2, 3, 4\}$ and the last 1-2 digits are an integer in the interval $[1, 25]$. 14321, 34222, 123413 are all valid keys.

This key is used to define the order of which encryption/decryption functions to use and an offset.

Suppose that the key has the form `key` $= d_1d_2d_3d_4d_5d_6$, where $d_1, d_2, d_3, d_4 \in [1, 4], d_5 \in [1, 9]$, and $d_6 \in [0, 5]$ are integers. Let s be the integer formed by concatenating the digits d_5 and d_6 together. Note that d_6 is optional and need not be included in the key. Then we choose our encryption functions to be an ordering of the functions $g_n(x) = f_n(x) + s$ with corresponding decryption functions $g_n^{-1}(x) = f_n(x - s)$ depending on the values of d_k for $k \in \{1, 2, 3, 4\}$. This ordering is explained more in section 2.4.

Since there are 4 ways to pick each of d_1, d_2, d_3, d_4 and 25 ways to pick the pair d_5d_6, we have that there are $4 \cdot 4 \cdot 4 \cdot 4 \cdot 25 = 4^4 \cdot 25 = 6400$ possible keys.

Date: September 2020.

2.3. **Inputs and Alphabet.** Once we have obtained our encryption/decryption functions using the key, we can encrypt each letter of the plaintext. Note that the only acceptable inputs to our cipher are a letter in the alphabet, without case. Spaces or other punctuation are skipped by this algorithm; they are not present in the output.

We are using a zero-based representation of the alphabet, meaning 0 corresponds to `a`, and 25 corresponds to `z`. This maps the alphabet into $\mathbb{Z}_{26}$.

2.4. **Transformation.** The basic equations used to encrypt are:

$$f_n(x) = x^n \tag{1}$$

$$g_n(x) = f_n(x) + s \tag{2}$$

Where x is the number to encrypt (according to the alphabet defined above) and s is the number at the end of the key (s described in section 2.1), and n is one of 1, 5, 7, or 11.

We determine the order of which $g_n(x)$ to use to encrypt a plaintext character in a cyclical pattern given by the first four characters of the key. The first four characters of the key define a simple mapping to which $g_n(x)$ to use.

- 1 in the key maps to $g_1(x)$
- 2 in the key maps to $g_5(x)$
- 3 in the key maps to $g_7(x)$
- 4 in the key maps to $g_{11}(x)$

In other words, if the first few characters of plaintext to encrypt are $c_1, c_2, c_3, c_4, c_5, c_6, \ldots$, and the key is 423111, the pattern of which $g_n(x)$ to use is $g_{11}(c_1)$, $g_5(c_2)$, $g_7(c_3)$, $g_1(c_4)$, $g_{11}(c_5)$, $g_5(c_6)$, $g_7(c_7)$, $g_1(c_8)$ repeating until the end of the message.

2.5. **Example.** As an example, suppose that we wish to encrypt the plaintext `CRYPTO` and we choose the key 132417.

For readability, we use the plaintext letters as input to the functions g_n rather than their representatives in $\mathbb{Z}_{26}$. Because of the pattern in the key, we use g_1 to encrypt the first plaintext letter, g_7 to encrypt the second plaintext letter, and so forth.

$$\begin{aligned}
g_1(\texttt{C}) &= f_1(\texttt{C}) + 17 = 19 = \texttt{T}\\
g_7(\texttt{R}) &= f_7(\texttt{R}) + 17 = 8 = \texttt{I}\\
g_5(\texttt{Y}) &= f_5(\texttt{Y}) + 17 = 11 = \texttt{L}\\
g_{11}(\texttt{P}) &= f_{11}(\texttt{P}) + 17 = 24 = \texttt{Y}\\
g_1(\texttt{T}) &= f_1(\texttt{T}) + 17 = 10 = \texttt{K}\\
g_7(\texttt{O}) &= f_7(\texttt{O}) + 17 = 5 = \texttt{F}
\end{aligned}$$

Hence the final encrypted message is `TILYKF`. If we were to decrypt this same message, all that is needed are the decryption functions given by $g_1^{-1}(x) = f_1(x-17), g_7^{-1}(x) = f_7(x-17), g_5^{-1}(x) = f_5(x-17)$, and $g_{11}^{-1}(x) = f_{11}(x-17)$. Using g_1^{-1} to decrypt the first ciphertext letter, g_7^{-1} to decrypt the second ciphertext letter, and so forth, we see that the plaintext letters are

$$\begin{aligned}
g_1^{-1}(\texttt{T}) &= f_1(\texttt{T} - 17) = 2 = \texttt{C}\\
g_7^{-1}(\texttt{I}) &= f_7(\texttt{I} - 17) = 17 = \texttt{R}\\
g_5^{-1}(\texttt{L}) &= f_5(\texttt{L} - 17) = 24 = \texttt{Y}\\
g_{11}^{-1}(\texttt{Y}) &= f_{11}(\texttt{Y} - 17) = 15 = \texttt{P}\\
g_1^{-1}(\texttt{K}) &= f_1(\texttt{K} - 17) = 19 = \texttt{T}\\
g_7^{-1}(\texttt{F}) &= f_7(\texttt{F} - 17) = 14 = \texttt{O}
\end{aligned}$$

Thus the decrypted message is the original plaintext we started with, namely `CRYPTO`.

3. Conclusion

In this report we have demonstrated a cipher that will allow OCRA Creative Recursive Acronyms, Inc to add a layer of privacy to their employee's messages, hampering and inadvertent leaks of confidential proprietary information. The cipher accepts a key of length 5 or 6 characters and a message of any length, and uses modular exponentiation to hide the plaintext. The keys and output used by this cipher have an easy-to-read representation that can be typed on any keyboard.

As demonstrated, this cipher will obfuscate any misplaced sensitive company information. Any third party who sees sensitive company information or trade secrets in a leaked employee's message will need to know the cipher key used to encrypt it. This will reduce the chance of any more embarrassment to the company or harm to your future revenue streams.