Group I

Introduction

We have received your request to build a nontrivial cryptographic algorithm that can convert up to 160 characters of plaintext into up to 800 characters of ciphertext, for use in employee communications. Our algorithm uses as its encryption key an integer between 1 and 27, and our algorithm converts each character of plaintext into one character of ciphertext.

Encryption Overview

To encode any specific character in the plaintext, we take the integer value corresponding to the last ciphertext integer and multiply that integer by the encryption key. We take the newly created integer and add the position of the character in the text and the value of the plaintext character. Finally, we take the modulus of that sum and 27, and end up with an integer between 0 and 26, which corresponds to the value of a letter in the alphabet. That corresponding letter is used as the ciphertext character.

So, the steps in order are:

$$cipherValue = lastCipherValue * encryptionKey$$

$$cipherValue = cipherValue + plaintextPosition + plaintextValue$$

$$finalCipherValue = cipherValue \bmod 27$$

To compute the value of a modulo b, compute the following, with a, b, q, and r being integers:

$a \bmod b = r$, where:

$r = a - bq$, and $0 \leq r < |b|$

The value of a character is its position in the alphabet, starting indexed from 0, with space taking on the value of 26.

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	
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	26

The last ciphertext value is assumed to be 0 for the first character encrypted and is updated after each new character is encoded from left to right. The position of the character in the text is assumed to be indexed starting from 0.

For an example, let us assume that we are encoding the second character of a message, which is the plaintext character ‘w’, our encryption key is 25, and the ciphertext character value for the first character of the message was an ‘e’, which has a value of 4. Because this is the second character of the message, it is in position 1, and because it is a ‘w’, it has a value of 22.

So, in the above steps, lastCipherValue is 4, encryptionKey is 25, plaintextPosition is 1, and plaintextValue is 22.

$$cipherValue = 4 * 25 = 100$$

$$cipherValue = 100 + 1 + 22 = 122$$

$$finalCipherValue = 122 \, mod \, 27 = 15$$

By following the steps, we discover that we would get a value of 15, so we check our alphabet and find that 'p' has a value of 15. So, the ciphertext assigned is the letter 'p'.

Decryption Overview

To decrypt any specific character, we take the alphabet value of the previous ciphered character, and multiply it by the encryption key, just as we did in the encryption, and then add the ciphered characters position. We then take that sum mod 27, and call this number the decryption value. We take the alphabet value of the current ciphered character and subtract the decryption value, before adding 27. We then take that sum mod 27, and end up with an integer between 0 and 26, which corresponds to the value of a letter in the alphabet. That corresponding letter is used as the plaintext character.

So, the steps in order are:

$$decryptionValue = lastCipherValue * encryptionKey$$

$$decryptionValue = decryptionValue + plaintextPosition$$

$$decryptionValue = decryptionValue \, mod \, 27$$

$$cipherValue = currentCipherValue - decryptionValue + 27$$

$$finalDecipheredValue = cipherValue \, mod \, 27$$

For an example, let us use the same values as we did for the encryption example. We have an encryption key of 25, we are on the second character of the ciphered message, so the position is 1. The current ciphered character is a 'p' and so has a value of 15. The first character of the ciphered message is a 'e', and so has a value of 4.

$$decryptionValue = 4 * 25 = 100$$

$$decryptionValue = 100 + 1 = 101$$

$$decryptionValue = 101 \, mod \, 27 = 20$$

$$cipherValue = 15 - 20 + 27 = 22$$

$$finalDecipheredValue = 22 \, mod \, 27 = 22$$

By following the steps, we discover that we would get a value of 22, so we check our alphabet and find that ‘w’ has a value of 22. So, the plaintext is the letter ‘w’.

Conclusion

In conclusion, we have presented above our non-trivial cryptographic method to encrypt the data in your employee communications. By converting each plaintext character into a singular ciphertext character, we have kept the length of communications to a max of 160 characters a message, and with a key size of two digits, we believe our method will be of minimal impact on employee devices.

We hope this method meets your specifications, and that you will have a great day.

Sincerely,

[redacted]

Postscript: This report has some parts simplified for ease of explanation and reuse in your application. For better security, position could be assigned starting from a number other than 0, giving us another key to work with, and making the code harder to break.