**Title:**

Evaluating and understanding the Affine Cipher.

**Body:**

In this report, the technical and security aspects of the Affine Cipher will be reviewed, examined, and evaluated for a greater understanding. Strengths and weaknesses will be examined, evaluated and understood, so the correct decision on if it should be used within an institution such as DMU Internal Communications can be concluded.

**I. Introduction**

Encryption is an ancient process that converts information into a secret code preventing any unauthorized access. Its use has been dated back thousands of years to protect any sensitive information, however, they used very insecure methods of encryption that have flaws. Nowadays encryption methods consist of math equations/algorithms, the two main types being symmetric and asymmetric key encryption. Symmetric consists of the same key used for both encryption and decryption, whereas asymmetric uses two different keys.

The purpose of this report will be to review, examine, and understand the Affine Cipher, a classic type of encryption algorithm. This report will also help judgment be placed on if this type of encryption algorithm will be suitable for sufficient use within DMU Internal Communications. This will be done by highlighting any possible security implications which will be presented clearly and concisely.

**II. Technical & Mathematical Description of the Affine Cipher**

This part will describe the technical and mathematical parts of the Affine Cipher. The Affine Cipher is a monoalphabetic substitution cipher, which means that each letter in the alphabet is assigned a numeric equivalent, and then encrypted using simple mathematical formulas1. It is a standard substitution with a rule governing which letter encrypts to what, this leads to weaknesses which will further be explained.

Each letter is enciphered using the formula *“E(x) = (ax + b) mod 26”*, where *“x"* stands for the numerical value of the plaintext alphabet letter, and *“a”*and *“b”* stands for the key perimeters used for encryption. Here is an example of if the letter D (4th letter in the alphabet) was encrypted using the key perimeters *“a*”=3 and *“b”*=9.

We can use the formula described above using our numerical values;

*“x”*=4 (D), *“a”*=3, and *“b”*=9.

E(4) = (3\*4 + 9)

E(4) = 21

Therefore, the encrypted version of D using these key perimeters would result in the 21st letter of the alphabet, V.

**III. Cracking the Ciphertext**

The code used to crack the given ciphertext is quite simple but works using brute force.

It uses all possible values of *“a” in* the range 1 to 25 and “*b*” in the range 0 to 25, then using the *“mod\_inverse”* function *“a”* is checked for any modular inverses modulo *“m”* from *“m=26”*, if it’s not valid then it skips to the next value of *“a”.* In the affine\_crack function, different values of *“a” (1 to 25)* are used to see which one can decrypt it, if *“a=13”* is encountered it cannot be used as it doesn’t have a modular inverse; therefore, we would move onto *“a=14”.*

The ciphertext is then decrypted using the formula *“p\_num = (a\_inv \* (c\_num - b)) % m****”*.** *“p\_num”* is obtained by applying the inverse transformation to *“c\_num*” and both are the number range of the ciphertext (0, 25) where 0 represents *“a”.* The value of *“c\_num”* is obtained by subtracting the ASCII value of *“a”* from the ASCII value of the ciphertext letter. *“a\_inv”* represents the modular inverse of *“a”* modulo *“m”*, whilst *“b”* is the additive key, and *“m”* is the size of the alphabet.

Basically, the decryption formula applies the inverse affine transformation to each letter of the ciphertext and obtains the corresponding plaintext letter. When the correct decryption key is produced the plaintext message will make sense, any incorrect keys will produce incomprehensible results. Brute force uses all possible keys until one is found that is usable.

**IV. Evaluation of Affine Cipher for DMU Internal Communication**

There are a couple of advantages and disadvantages of using the Affine Cipher for DMU Internal Communications. As it is relatively easy to implement it will not take long for people to understand it properly, it can also be strengthened by using larger keys, however, there is a limit. It’s also stronger than some other types of encryptions such as the Caeser Cipher.

Unfortunately, as it uses a monoalphabetic substitution algorithm it is not very secure and can be cracked very easily, the English alphabet allows for only 12 options for the variable “a”, and since the alphabet is 26 letters long the value of “b” can only be 26. This means that there is only 312 (12 \* 26) keys possible3. It’s also susceptible to brute force attacks, which can be worse when smaller key sizes are used.

Overall, it does not seem to be in the best interest to use the Affine Cipher as the encryption method for DMU Internal Communications. As it would be storing sensitive data such as financial and personal data e.g. student ID information, it would be recommended not to be used as there are a lot of security risks that can be decrypted with somewhat simple methods.

References

[1]https://en.wikipedia.org/wiki/Affine\_cipher#:~:text=The%20affine%20cipher%20is%20a,converted%20back%20to%20a%20letter.

[2]https://en.wikipedia.org/wiki/Affine\_cipher#:~:text=The%20affine%20cipher%20is%20a,converted%20back%20to%20a%20letter.

[3] https://privacycanada.net/affine-cipher/