1 Purpose

The field of cryptography is filled with techniques for protecting data. The importance of securing data is becoming more important as computers are being used for storing personal information. In this project we will look at a simple technique for encrypting and decrypting data. Even though this technique is not used anymore and easily broken, it should be a good starting point to show how classical cryptography systems were used.

2 Background

The idea behind cryptography is that party $a$ wants to send a message (plaintext) to another party $b$. Party $a$ will encrypt the message with a method that is prearranged with party $b$. The encryption method will use a key (or password) to turn the plaintext message into the encrypted message, called ciphertext. When party $b$ receives the encrypted message they will change it back into plaintext using the key.

For our project we will make some conventions to simplify the process:

- plaintext and ciphertext will be converted to all lower case letters
- The letters of the alphabet are assigned numbers as follows:

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

  | q | r | s | t | u | v | w | x | y | z |
  | 16| 17| 18| 19| 20| 21| 22| 23| 24| 25|

- Space and punctuation are omitted. This is necessary to prevent easy attacks to the system.

We have seen in class the simple shift cipher which shifts each character in the string by a certain number of characters. The shift ciphers may be generalized and slightly strengthened as follows. Choose two integers $\alpha$ and $\beta$, with $\gcd(\alpha, 26) = 1$, and consider the function (called an affine function)

$$x \rightarrow \alpha x + \beta \pmod{26}$$

For example, let $\alpha = 9$ and $\beta = 2$, so we are working with $9x + 2$. Take a plaintext letter such as $h(= 7)$. It is encrypted to $9 \cdot 7 + 2 = 65 \equiv 13 \pmod{26}$, which is the letter $N$. Using the same function, we obtain
The key for the affine cipher are the values $\alpha$ and $\beta$. Since the condition $\gcd(\alpha,26) = 1$, must be met we will need to write a $\gcd$ function to compute the greatest common divisor. Here is the pseudo code for a simple algorithm for computing the $\gcd$ by Euclid’s method:

$$\text{GCD}(m,n)$$
while n > 0
    r = m mod n
    m = n
    n = r
return m

3 Requirements

For this project you will need to implement a program in C++ which encrypts messages using the affine cipher. The following must be present in your project:

- One source code file named crypto.cpp
- At least one for loop
- Use cin (or getline) for entering the key ($\alpha,\beta$) and text
- Two functions:
  - int $\gcd(\text{int } m, \text{int } n)$
  - string affine_encrypt( string plaintext, const int a, const int b )
- Use of the modular (%) operator
- Program prints out the encrypted text using the given key
- Program should use the stripify_string function that was given in class to convert the input text.

The program output should look something like:

Enter a,b: 9 2
Enter text: affine
Encrypted text: cvvwpm

4 Grading

The following table shows the scoring that will be used for this project:

<table>
<thead>
<tr>
<th>Area</th>
<th>Max Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>35</td>
</tr>
<tr>
<td>Requirements</td>
<td>35</td>
</tr>
<tr>
<td>Comments</td>
<td>15</td>
</tr>
<tr>
<td>Neatness</td>
<td>15</td>
</tr>
</tbody>
</table>