

SurajSG23 /
Cryptography-Assignment

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Cryptography-Assignment / Affine Cipher



SurajSG23 Create Affine Cipher

9f9befe · 9 hours ago



88 lines (74 loc) · 2.71 KB

Code

Blame



Raw



```
1  import java.util.Scanner;
2
3  public class AffineCipher {
4
5      // Function to perform modular inverse
6      public static int modInverse(int a, int m) {
7          a = a % m;
8          for (int x = 1; x < m; x++) {
9              if ((a * x) % m == 1) {
10                 return x;
11             }
12         }
13         return -1;
14     }
15
16     // Encrypt the plaintext using Affine Cipher
17     public static String encrypt(String text, int a, int b) {
18         StringBuilder cipherText = new StringBuilder();
19         text = text.toLowerCase();
20         for (int i = 0; i < text.length(); i++) {
21             char c = text.charAt(i);
22             if (c >= 'a' && c <= 'z') {
23                 int x = c - 'a';
24                 char encryptedChar = (char) (((a * x + b) % 26) + 'a');
25                 cipherText.append(encryptedChar);
26             } else {
27                 cipherText.append(c); // Keep non-alphabetic characters unchanged
28             }
29         }
30         return cipherText.toString();
31     }
32
33     // Decrypt the ciphertext using Affine Cipher
34     public static String decrypt(String text, int a, int b) {
35         StringBuilder plainText = new StringBuilder();
36         int a_inv = modInverse(a, 26); // Find modular inverse of a
37     }
```

```
38     if (a_inv == -1) {
39         return "Inverse of 'a' does not exist.";
40     }
41
42     text = text.toLowerCase();
43     for (int i = 0; i < text.length(); i++) {
44         char c = text.charAt(i);
45         if (c >= 'a' && c <= 'z') {
46             int y = c - 'a';
47             char decryptedChar = (char) (((a_inv * (y - b + 26)) % 26) + 'a');
48             plainText.append(decryptedChar);
49         } else {
50             plainText.append(c); // Keep non-alphabetic characters unchanged
51         }
52     }
53     return plainText.toString();
54 }
55
56 public static void main(String[] args) {
57     Scanner scanner = new Scanner(System.in);
58
59     System.out.println("Affine Cipher");
60
61     // Get inputs for encryption
62     System.out.print("Enter the plaintext: ");
63     String plaintext = scanner.nextLine();
64     System.out.print("Enter the multiplier (a): ");
65     int a = scanner.nextInt();
66     System.out.print("Enter the shift (b): ");
67     int b = scanner.nextInt();
68
69     // Encrypt the plaintext
70     String encryptedText = encrypt(plaintext, a, b);
71     System.out.println("Encrypted Text: " + encryptedText);
72
73     // Decrypt the ciphertext
74     String decryptedText = decrypt(encryptedText, a, b);
75     System.out.println("Decrypted Text: " + decryptedText);
76
77     scanner.close();
78 }
79 }
80
81 //Output
82 Affine Cipher
83 Enter the plaintext: hello i am Suraj
84 Enter the multiplier (a): 1
85 Enter the shift (b): 4
86 Encrypted Text: lipps m eq wyven
87 Decrypted Text: hello i am suraj
```