LUCRAREA INDIVIDUALĂ (EXEMPLU)

Sistemul de criptare AFIN

Prenume1 Nume1 – Prenume2 Nume2 – Prenume3 Nume3 – . . . – n

Despre

În criptografie, sistemul AFIN nu este altceva decât o generalizare a sistemului CAESAR.

Descrierea

Procesul criptografic AFIN constă în utilizarea unui număr cheie în calitate de cheie de criptare, precum şi unui pas de parcurgere a alfabetului. Numărul cheie şi pasul pot fi doar în limita alfabetului latin, deci între 0 şi 25. Deci, metoda constă în rescrierea alfabetului latin permutat ciclic începând de la numărul cheie şi continuând conform pasului. Criptarea şi decriptarea se fac numai în baza numărului cheie şi a pasului. Dacă numărul cheie este 5 iar pasul este 3 vom obţine următoarea configurare prezentată mai jos.

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
F I L O R U X A D G J M P S V Y B E H K N Q T W Z C

Exemplu

$k=5, p=3$ P R I M A V A R A T A R Z I E Y E D P F Q F E F K D E C D R	$crypt(x)=ax+b \bmod 26$ $decrypt(y)=a^{-1}y+a^{-1}(26-b) \bmod 26$

Implementări

Pentru criptare/decriptare vom utiliza următoarele funcţii integrate în program în fişierul „afn.h”, care va arăta cam aşa.

```
void AFN_encrypt(char msg[])
{
    int i, j, k, key, step;
    char str[50];

    printf("skey: "); scanf("%i", &key);
    printf("step: "); scanf("%i", &step);

    for(i=0; i<strlen(msg); i++) {
        for(j=0; j<strlen(letters); j++)
```

```
            if(msg[i]==letters[j]) {
                k=(j*step+key)%strlen(letters);
                str[i]=letters[k];
            }
    }
    printf("encr: ");

    for(i=0; i<strlen(msg); i++)
        printf("%c", str[i]);
    printf("\n");
}

void AFN_decrypt(char msg[])
{
    int i, j, k, key, step, InvStep, InvKey;
    char str[50];

    printf("skey: "); scanf("%i", &key);
    printf("step: "); scanf("%i", &step);

    InvStep=ModMulInverse(step,strlen(letters));
    InvKey=(InvStep*(strlen(letters)-key))%strlen(letters);
    for(i=0; i<strlen(msg); i++) {
        for(j=0; j<strlen(letters); j++)
            if(letters[j]==msg[i]) {
                k=(InvStep*j+InvKey)%strlen(letters);
                str[i]=letters[k];
            }
    }
    printf("decr: ");

    for(i=0; i<strlen(msg); i++)
        printf("%c", str[i]);
    printf("\n");
}
```

Funcţiei de bază „main.h” care rulează totul i se va mai adăuga următoarele rânduri.

```
else if(strcmp(str1,"-Eafn")==0) AFN_encrypt(str2);
else if(strcmp(str1,"-Dafn")==0) AFN_decrypt(str2);
```

În continuare, vom prezenta funcţionalitatea aplicaţiei descrise anterior, pe care am numit-o „CRYPTO”. Prin intermediul acesteia vom efectua 3 exemple de criptare/decriptare a mesajelor „ioanjeleascov” şi „tatianaleca” utilizând diferite chei. Astfel, vom verifica corectitudinea executării operaţiilor al acestui modul al aplicaţiei. Demn de remarcat este faptul că în exemplul 3 este prezentată o eroare de calcul, ce denotă faptul că dacă pasul e mai mic de numărul 3 atunci eroarea este prezentă.

Exemplul 1

```
jo@laptop:~/Documents/work/crypto/final$ ./crypto -Eafn ioanjeleascov
skey: 5
step: 3
encr: dvfsgrmrfhlvq
jo@laptop:~/Documents/work/crypto/final$ ./crypto -Dafn dvfsgrmrfhlvq
skey: 5
step: 3
decr: ioanjeleascov
```

Exemplul 2

```
jo@laptop:~/Documents/work/crypto/final$ ./crypto -Eafn ioanjeleascov
skey: 2
step: 7
encr: gwcpnebecyqwt
jo@laptop:~/Documents/work/crypto/final$ ./crypto -Dafn gwcpnebecyqwt
skey: 2
step: 7
decr: ioanjeleascov
```

Exemplul 3

```
jo@laptop:~/Documents/work/crypto/final$ ./crypto -Eafn ioanjeleascov
skey: 4
step: 2
encr: ugeewmameoigu
jo@laptop:~/Documents/work/crypto/final$ ./crypto -Dafn ugeewmameoigu
skey: 4
step: 2
decr: aaaaaaaaaaaaa
```