SEMAP - Šifrovanie

Norbert Matuška

xmatuskan@stuba.sk

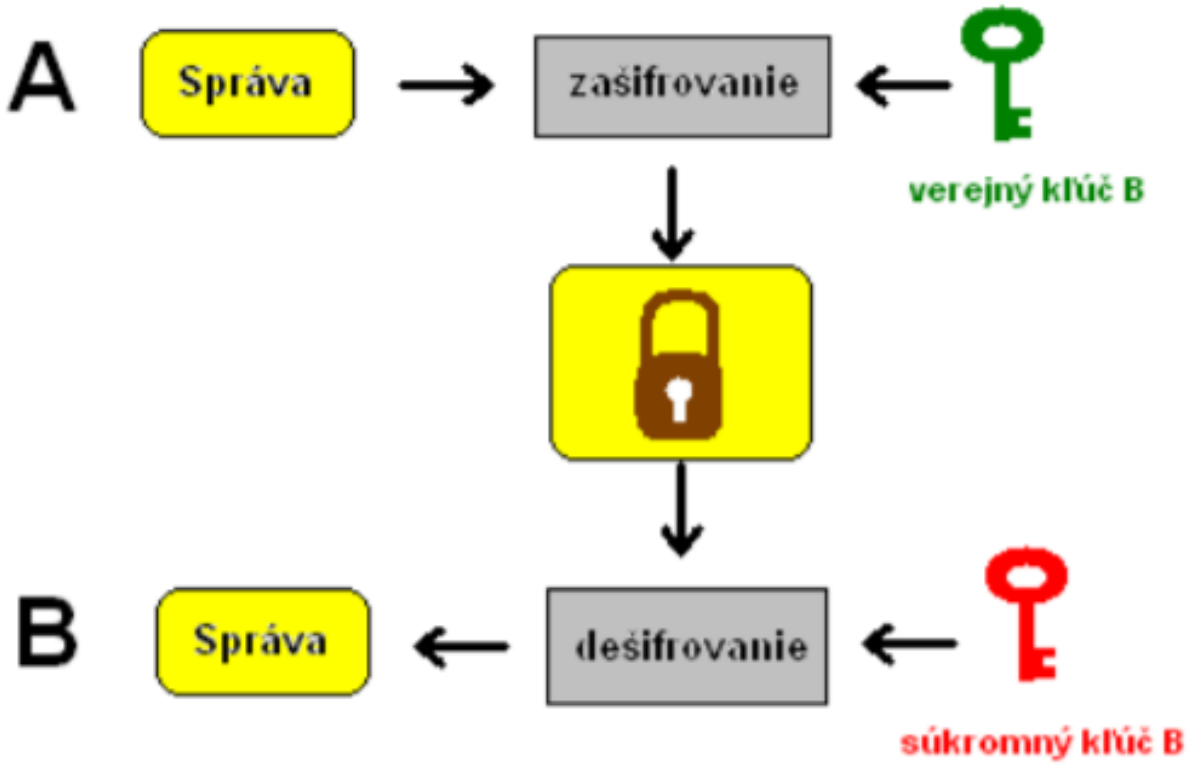
Úvod

Veľmi zjednodušene môžeme povedať, že šifrovanie je zapisovanie textu v takej podobe, aby mu nepovolaný čitateľ nemal šancu porozumieť. Najčastejšie sa šifrovanie v jeho rôznych podobách používa pri komunikácii: jeden človek má informáciu, ktorú potrebuje poslať inému; nechce však, aby si ju mohol cestou prečítať ktokoľvek ďalší.

Kryptografia: vedná oblasť zaoberajúca sa ukrývaním obsahu textu pred nepovolanými osobami.

Kryptoanalýza: vedná oblasť zaoberajúcu sa štúdiom metód, ktoré môže útočník použiť pri snahe získať zo zašifrovaného textu ukryté informácie.

Steganografia: vedná oblasť, ktorá sa zaoberá utajovaním samotného prenosu správ.[1]



Obrázkový príklad asymetrického šifrovania.[2]

[1] https://horvatha.edupage.org/files/Sifrovanie.pdf
[2] https://horvatha.edupage.org/files/Sifrovanie.pdf

Implementácia

Cézarova šifra

Cezárova šifra funguje na jednoduchom princípe posúvania sa v abecede. Napríklad si vyberieme kľúč 3. Takže každé písmeno stringu posunieme o hodnotu tri, čiže zo slova “Hello” nám vznikne slovo Khoor (ak používame anglickú abecedu). Implementácia takéhoto šifrovania je jednoduchá. Vytvoríme si krátky program na Cézarovu šifru:

```
#include <stdio.h>

int main()
{
    int i, x;
    char str[100];

    printf("\nZadajte string:\t");
    scanf("%[^\n]", str);

    printf("\nVyberte moznost\n");
    printf("1 = Zasifrovat string.\n");
    printf("2 = Desifrovat string.\n");
    scanf("%d", &x);


    switch(x)
    {
        case 1:
            for(i = 0; (i < 100 && str[i] != '\0'); i++)
                str[i] = str[i] + 3; //Pouzivame kluc 3, takze pridame
hodnotu 3 k ASCII hodnote

            printf("\nZasifrovany string: %s\n", str);
            break;

        case 2:
            for(i = 0; (i < 100 && str[i] != '\0'); i++)
                str[i] = str[i] - 3; //Pouzivame kluc 3, takze odoberem
hodnotu 3 k ASCII hodnote

            printf("\nDesifrovany string: %s\n", str);
            break;

        default:
            printf("\nError\n");
            break;
    }
    return 0;
}
```

Proces zašifrovania

Proces dešifrovania

Popísanie procesu je celkom jednoduché, program dostane vstupné slovo alebo text, načíta ho do pola str a následne pomocou for cyklu prejde cez všetky chary v poli a posunie ich cez ASCII tabuľku o 3 (alebo iný zvolený kľúč) dopredu alebo dozadu (podľa vybranej možnosti).

Takýto process môže byť využitý aj pre väčšie *súbory* s väčším obsahom znakov.

Bohužiaľ, tento typ šifrovania je veľmi jednoduchý na prelomenie pomocou počítačov. Avšak v minulosti bola takáto šifra relatívne silná. Ak však dešifrant vedel, aké slovo alebo slová sa v texte musia určite vyskytnúť, bola ľahko prelomiteľná.

Transpozičná šifra

Existuje viacero druhov transpozičného šifrovania. Toto šifrovanie spočíva v zamiešaní poradia textu podľa predom určených pravidiel aby následne dešifrovanie bolo možné. Medzi transpozičné šifry patrí takzvaná **Rail Fence šifra**, ktorá podľa kľúču, povedzme že tri, vytvorí 3 riadky v dĺžke zadanej správy. Povedzme, že vytvorená "tabuľka" má súradnice(pre zjednodušenie predstavenia si takejto šifry). Na súradnicu 0:0 (prvý riadok prvého stĺpca) pôjde prvé písmeno, na súradnicu 1:1 (druhý riadok druhého stĺpca) pôjde druhe písmeno atď. Ak narazíme na spodný riadok, ďalšie písmeno pôjde do nasledujúceho stĺpca ale o riadok vyššie. Na konci môžeme dostať nasledovnú sekvenciu: 0:0, 1:1, 2:2, 1:3, 0:4, 1:5, 2:6, 1:7, 0:8, ….

Prvé číslo sa vždy opakujú čísla 0,1,2 a potom naspať k 0: 0,1,2,1,0,1,2…

Druhé číslo ide vzostupne od 0,1,2,3,4,…

Nasledujúcim spôsobom môžeme zašifrovať správu "**Hello World**":

H				**o**				**r**		
	e		**l**				**o**		**l**	
		l				**W**				**d**

Nasledujúcim krokom je už iba čítať horizontálne a máme zašifrovanú správu: "**Horel ollWd**".

Tu je výborný príklad na **Rail Fence šifru**:[3]

[3] http://www.cprograms4future.com/p/encryption-rail-fence-cipher.html

```c
#include<stdio.h>
#include<string.h>
int main()
{
    int i,j,len,rails,count,code[100][1000];
    char str[1000];
    printf("Enter a Secret Message\n");
    gets(str);
    len=strlen(str);
    printf("Enter number of rails\n");
    scanf("%d",&rails);
    for(i=0;i<rails;i++)
    {
        for(j=0;j<len;j++)
        {
            code[i][j]=0;
        }
    }
    count=0;
    j=0;
    while(j<len)
    {
        if(count%2==0)
        {
            for(i=0;i<rails;i++)
            {
                code[i][j]=(int)str[j];
                j++;
            }

        }
        else
        {

            for(i=rails-2;i>0;i--)
            {
                code[i][j]=(int)str[j];
                j++;
            }
        }

        count++;
    }

    for(i=0;i<rails;i++)
    {
        for(j=0;j<len;j++)
        {
            if(code[i][j]!=0)
                printf("%c",code[i][j]);
        }

    }
    printf("\n");
    return 0;
}
```

Kód, ktorý je vyššie funguje tak, že prečíta string, ktorý načíta do pola a následne podľa špecifických podmienok ako sme si ukázali vyššie v tabulke poprehadzuje písmená a vypíše miesta v poli kde sa niečo nachádza.

Output:

```
Enter a Secret Message
Hello World
Enter number of rails
3
Horel ollWd
```

Ak by sme sa rozhodli vynechať poslednú podmienku pred printf, output je nasledovný:

```
Enter a Secret Message
Hello World
Enter number of rails
3
H   o   r   e l   o l   l   W   d
```

Tento spôsob šifrovania je na podobnej úrovni ako Cézarove šifrovanie, tým, že bezpečnosť oboch je veľmi malá ale Rail Fence vyžaduje mierne zložitejšie dešifrovanie ako Cézarove. Pri Cézarovom stačí odčítať podľa kľúču hodnotu o ktorú sa má abeceda posunúť.

Porovnanie s inými metódami

Vernamova šifra

Funguje na celkom "podobnom" princípe ako Cézarova: posúvanie písmeniek v abecede, lenže tu je rozdieľ v kľúči. Kľúčom je úplne náhodný číselný reťazec, ktorý sa môže opakovať ale aj nemusí.

Ukážeme si to na príklade:[4]

```
T O T O J E S T R A S N E T A J N A S P R A V A
3 3 7 5 3 3 7 5 3 3 7 5 3 3 7 5 3 3 7 5 3 3 7 5
-----------------------------------------------------
W R A T M H Z Y U D Z S H W H O Q D Z U U D C F
T + 3 = W; O + 3 = R; T + 7 = A; O + 5 = T ….
```

Takýto jednoduchý príklad šifry sa môže zmeniť na ťažko dešifrovateľnú ak – kľúč je rovnako dlhý ako string a kľúč je tvorený náhodným reťazcom čísiel.

[4] http://www.smatana.sk/texty/vernamova-sifra

Oproti predošlým šifrám vie byť táto šifra v princípe nerozlúštiteľná pri absencii hocijakej informácii o správe alebo o kľúči, kvôli náhodnosti zvolených čísiel v kľúči. Gilbert Vernam tvrdil, že si je istý, že jeho šifra je nerozluštitelná. S exaktným dôkazom ale prišiel až C. E. Shannon v roku 1949. Dôkaz je založený na tom, že náhodný posun v abecede sa rovná nahradeniu úplne náhodným písmenom.[5]

Zhrnutie

Ukázali sme, čo to vlastne šifrovanie je, ako to ilustračne funguje. Do väčšej hĺbky sme sa pozreli pri dvoch typoch šifrovania a taktiež sme si ukázali ako to môže byť implementované v jazyku C. Porovnali sme si ich s ďalším typom šifrovania, ktorý je pri správnych podmienkach virtuálne neprelomiteľný.

[5] https://cs.wikipedia.org/wiki/Vernamova_%C5%A1ifra