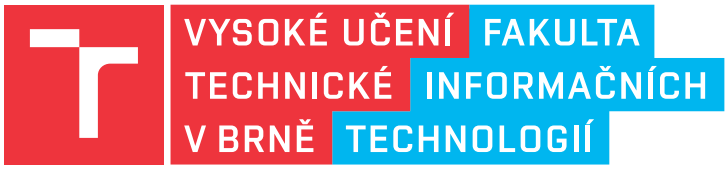

Dokumentácia projektu
Predmet: **KRY** 2018/19
Projekt: **Projekt č. 1**
Autor: **Michael Halinár** xhalin01

1 Rozbor

Úlohou bolo prelomiť neznámu synchrónnu prúdovú šifru, a získať tak tajomstvo. Prvým krokom je teda získanie keystreamu. Toho docielime XOR-ovaním zašifrovaného súboru a známeho plaintextu. Následne tento keystream použijeme na rozšifrovanie súboru ktorý obsahuje vytváranie keystreamu.V získanom súbore vidíme časť programu ktorý bol použitý na vytvorenie keystreamu. Úlohou tedaje vytvorenie inverzných funkcií k vytváraniu tohoto keystreamu.

2 Šifra

Keystream je vytváraný v niekoľkých krokoch. V cykle sa 128 krát použijú dve funkcie na modifikovanie vstupu. Prvá funkcia `x` zoberie na vstupe hodnotu ,z nej vytiahne `MSB` a `LSB`, následne vytvorí nový reťazec pre ktorý platí:

$$x_{new} = x[LSB] \cdot x \cdot x[MSB].$$

Druhá funkcia `y` vytvára nový reťazec, tak že podľa hotnoty troch po sebe idúcich bitoch, vytvorí nový bit, nachádzajúci sa na pozícii poľa SUB. Tento proces opakuje po celej dĺžke vstupu, pričom sa začína od LSB.

3 Dešifrovanie

Úlohou dešifrovania je teda vytvorenie inverzných funkcií k `x` a `y`. Inverzná funkcia k `x` je v podstate orezanie čísla o jeden bit z prava aj z ľava. Vytvoríme teda orezávanie tak že pozrieme na druhý bit z prava, ak je to 0, vieme že aj MBS bude 0, stačí nám teda číslo na vstupe posunúť bitovým shiftom doprava. Ak je na druhom bite z prava hodnota 1, vieme že aj na pozícii MSB bude 1, pomocnou funkciou nastavíme teda na pozícii MSB bit na 0, a celé číslo ešte posunieme bitovým shiftom doprava.

Inverznú funkciu k `y` vytvoríme tak že bit po bite prechádzame daný vstup. Vieme že bit 0 mohol byť vytvorený postupnosťami: `000,011,101,111`, bit 1 zase postupnosťami `001,010,100,10`. Zistíme si teda kandidátov pre jednotlivé bity a v cykle sa snažíme aby sa kandidáti prelínali. Ak sa prelínajú spojíme ich cez 2 posledné bity a 2 prvé bity, takto postupujeme pre všetky bity vo vstupe. Následne máme vytvorených kandidátov z ktorých mohol vzniknúť keystream. Každý tento kandidát pošleme do pôvodnej funkcie step, ak sa výsledok rovná vstupu, našli sme inverznú podobu keystreamu. Tento postup opakujeme 128 krát tak ako bol použitý pri šifrovaní. Výsledná sekvencia bitov ukrýva tajomstvo, v tomto prípade `KRY{xhalin01-5a722b77c3c4dc8}`.