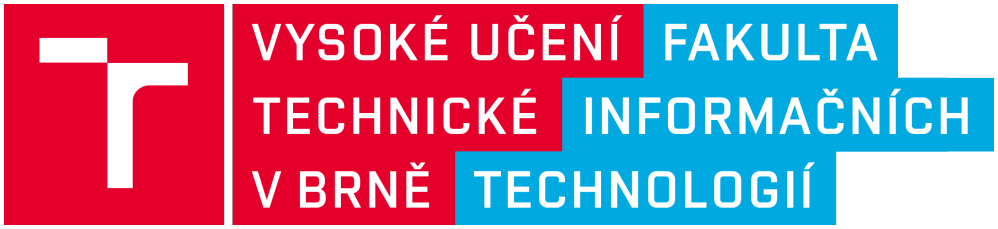

Kryptografie

Implementace a prolomení Afinní šifry

Tereza Burianová (xburia28)

19. března 2023

1 Struktura programu

Program obsahuje funkce pro šifrování (*encrypt*), dešifrování (*decrypt*), dešifrování bez klíče (*decryptWithoutKey*), dále také pomocné funkce pro kontrolu validity klíče A (*checkA*), kontrolu validity textu pomocí nejčastějších slov českého jazyka (*checkText*), kontrolu validity dešifrovaného textu a klíčů jako celku (*checkOutput*), samostatnou funkci pro výpočet multiplikativní inverze (*multInv*)[1] a funkce pro výpočet možných klíčů bez jejich znalosti (*calculateKeys, calculateKeysFirstEq, calculateKeysSecondEq*). Jednotlivé funkce jsou popsány dále v dokumentaci. Ve funkci *main* je provedeno parsování vstupu a několik základních kontrol validity vstupu. Dále byl také přiložen skript v jazyce Python, který umožnil automatizovat kontrolu programu na dodaných vstupech ve formátu csv, a z něhož byly získány procentuální hodnoty dále použity v této dokumentaci.

2 Šifrování a dešifrování

Základem funkcí pro šifrování a dešifrování textu je cyklus, který prochází jednotlivé znaky vstupního textu. Pokud je daný znak mezera, je zachován. Malá písmena abecedy jsou v šifrovací funkci transformována na velká písmena, dešifrovací funkce počítá pouze s písmeny velké abecedy. Velká písmena jsou transformována dle rovnic daných pravidly Afinní šifry, s korekcí jednotlivých částí výpočtu s ohledem na decimální ASCII hodnotu znaků. Další znaky jsou považovány za nevalidní a jsou ignorovány.

3 Dešifrování bez znalosti klíče

Dešifrování a zjištění klíče je provedeno pomocí frekvenční analýzy, konkrétně dle tří nejčastějších písmen. Nejprve jsou vypočítány četnosti jednotlivých písmen ve vstupním textu a vybrány tři nejčastější z nich. To je provedeno cyklením přes vstupní text, navýšením počtu daného písmene na odpovídajícím indexu v poli pro každý znak a následným dalším průchodem pro zjištění indexů s nejvyššími počty. V českém jazyce jsou tři nejčastější písmena **"e"** (index 4), **"a"** (index 0) a **"o"** (index 14). Tímto způsobem jsou získány tři rovnice:

$$input_1 = (4 * a + b) \mod 26 \quad (1)$$

$$input_2 = (0 * a + b) \mod 26 \quad (2)$$

$$input_3 = (14 * a + b) \mod 26 \quad (3)$$

Program se následně postupně pokusí spočítat klíče A a B nejprve sčítací metodou 1. a 3. rovnice (*calculateKeys*), následně dosazovací metodou 2. rovnice do 1. rovnice (*calculateKeysFirstEq*) a 2. rovnice do 3. rovnice (*calculateKeysSecondEq*). Po každém takovém kroku je provedena kontrola výsledného textu a další kroky jsou provedeny pouze, pokud nalezené klíče nebyly validní či výsledný text nebyl v českém jazyce, což lze určit porovnáním textu s několika nejčastějšími českými slovy. Tento přístup byl dle testování na poskytnutých vstupních textech úspěšný zhruba v 40 % případů.

Dle průzkumu neúspěšných pokusů se pořadí nejčastějších písmen v zašifrovaném textu a v českém jazyce ne vždy shodovala. Podobný postup byl tedy znovu proveden s vyměněnými písmeny na 1. a 2. místě a dále také na 2. a 3. místě. Takový postup zajistil 90% úspěšnost u poskytnutých vstupních textů.

[1] Kód inspirován pseudokódem z publikace Nechvatal, J.: PUBLIC-KEY CRYPTOGRAPHY.