Dokumenácia k 1. projektu z KRY

Dešifrovanie synchrónnej prúdovej šifry

Matej Marušák, Apríl 2018

1 Úvod

V tejto správe je popísané lámanie neznámej synchrónnej prúdovej šifry. V prvej kapitole je popísaný obecný princíp získania informácií o použitej šifre, kapitola 2. sa venuje riešeniu danej šifry ručne a kapitola 3. popisuje akým spôsobom je možné danú šifru prelomiť za pomoci SAT-solveru.

2 Získavanie informácií o šifre

Prvý útok, ktorý bol použitý je `known-plaintext attack`. Vykonaním funkcie XOR na súbory `bis.txt` a `bis.txt.enc` bol získaný `keystram`. Jeho dĺžka je len 512B, teda nie je možné odšifrovať všetky súbory ale len prvých 512B. Toto množstvo je dostatočné na to, aby bolo možné zistiť spôsob vytvárania `keystram`-u zo súboru `super_cipher.py.enc`. Opäť stačí vykonať XOR na daný súbor a keystream, ktorý sme predtým získali z *bis* súborov. Hlavným krokom je N-násobná transformácia inicializačného vektora funkciou `step`, ktorá vykonáva substitúciu a rotáciu aktuálneho kľúča.

3 Ručné riešenie

Ručné riešenie implementuje reverznú funkciu k funkcii `step`. Samotná funkcia sa skladá z dvoch krokov:

- Rotácia
- Substitúcia

Oba kroky sú reverzované osobitne a to nasledovne:

- Rotácia - vykoná sa rotácia do opačného smeru
- Substitúcia - Substitúcia sa vykonáva pre každý bit. Vstupom substitúcie sú aktuálny bit a dva predchádzajúce bity. Spolu teda tri bity indexujúce čísla od 0 do 7. Tento index sa používa ako index do poľa binárnych hodnôt. Indexovaným prvkom je nahradený bit aktuálnej pozície. Reverzácia tohoto kroku je zložitejšia, nakoľko najvyššie dva bity nie sú známe a je teda potrebné ich uhádnuť. Jedná sa len o 4 možnosti a ako najlepšie riešenie sa javí skúšať všetky možnosti, až kým vzniknutá šifra nie je korektne dešifrovaná. Na overenie validity sa dá použiť fakt že spodné dva bity sa musia rovnať vrchným dvom bitom.

4 Riešenie za pomoci SAT-solveru

Riešenie za pomoci SAT solvera je veľmi podobné ručnému riešeniu. Taktiež sa jedná o reverzáciu funkcie `step`. Rotácia sa revertuje rovnako ako pri ručnom riešení, len substitúcia sa rieši za pomoci SAT solvera. Riešenie sa skladá z troch krokov:

- Prevod aktuálneho kľúča na formulu v boolovej algebre - Pre každý bit keystreamu je vytvorená jedna premenná. Následne je vytvorená formula v konjuktívnej normálnej

forme, kde každý člen je podformula popisujúca jeden bit a to nasledovne - Ak je daný bit nulový, tak existujú 4 možnosti ako môžu vyzerať dva predchádzajúce bity (indexácia, viď ručné riešenie). Jedná sa teda a 4 podformule spojené logickým OR. Daný bit je vždy negovaný a predchádzajúce dva sú negované ak sa jedná o 0. Teda napríklad binárny kód 110 popisuje bit 0 (0 je aktuálna pozícia, označme ju v0) podformula by vyzerala nasledovne $v2|v1|\neg v0$. Pre bit s hodnotou 1 sa všetko len zneguje.

- Hľadanie ohodnotenia premenných - Na tento krok je použitá knižnica `satispy` pre Python3. Ako vstup je formula v boolovej algebre a výstupom je buď `False` a teda formula je neriešiteľná, alebo `True` aj s ohodnoteným premenných, pri ktorých je formula ohodnotiteľná.
- Konverzia ohodnotených premenných na kľúč - Výsledný kľúč je vytvorený bit po bite a to tak, že i-ty bit je nastavený na 1 ak je i-ta premenná ohodnotená ako pravdivá, inak je nastavený na 0.

5 Záver

V správe boli prezentované dva prístupy k hľadaniu inicializačného kľúča z keystramu. Jedná sa o ručné riešenie, v ktorom je priamo implementovaná reverzácia šifrovacieho algoritmu a riešenie za pomoci SAT-solveru. Obe tieto metódy pracujú očakávane a vždy nájdu inicializačný kľúč. Ručné riešenie je zložitejšie na naimplementovanie, avšak je výrazne rýchlejšie oproti SAT riešeniu, ktoré beží dlho, ale jeho implementácia je jednoduchá.