2. Domača naloga

Matija Kerkoč

April 17, 2019

Opisi funkcij so podani v Python datoteki, kjer so funkcije definirane, v dokumentu bomo opisali ideje, ki se skrivajo za njimi.

Geffejev generator - osnovni

Ideja rešitve je sledeča:

1. s pomočjo uganjene besede poiskati začetni del ključa s katerim je bilo besedilo zašifrirano,
2. z analizo ujemanja poiskati začetna zaporedja registrov LFSR1 in LFSR2,
3. s pregledom vseh ključev za register LFSR2 poiskati še začetno zaporedje za drugi register,
4. LFSR1, LFSR2 in LFSR3 zložiti v Geffejev generator ter odšifrirati kriptogram.

1. Najprej uganjeno besedo pretvorimo v dvojiški zapis s pomočjo slovarja. Sedaj vemo, kako se je šifriralo prvih 60 bitov, zato lahko s preprosto operacijo "XOR" takoj najdemo vsaj prvi del ključa z, saj velja $c_i = b_i \bigoplus z_i$ oz. $b_i = c_i \bigoplus z_i$ za vsak znak v prvih 60 bitih. Sedaj lahko na tem začetnem delu naredimo primerjanje ujemanja izhodov.

2. Vemo namreč, dase bosta izhoda LFSR1 ter LFSR3 ujemala z izhodom celotnega Geffejevega generatorja z v približno 75%. Zato lahko pregledamo znake, ter ugotovimo delež ujemanja ter v seznam shranimo tista začetna zaporedja, ki imajo delež ujemanja z ključem z okoli 0.75.

Opomba: delež ujemanja ne bo točno 0.75, zato je smiselno vzeti nekaj najboljših ključev, ter nato poiskati pravo odšifrirano besedilo.

3. S pregledom vseh izhodov registra LFSR2 poiščemo kandidate za besedila.

4. Pogledamo, ali kandidat vsebuje besedo "$CRYPTOGRAPHY$" in ga v tem primeru vrnemo.

Geffejev generator - napredni

Osnovna ideja ter funkcije za razbijanje šifre so iste, kar se spremeni, je pristop k problemu:

1. uganiti moramo nekaj besed, ki se bodo verjetno pojavile v besedilu,
2. za vsako izmed besed moramo pregledati celoten kriptogram, saj je lahko beseda na poljubnem mestu,
3. naredimo analizo ujemanja,
4. poiščemo pravo besedilo.

1. Problem je že z ugibanjem pravih besed, saj ne vemo kakšno besedilo imamo, prav tako pa morajo biti besede dovolj dolge, da lahko na tistem delu kriptograma izvedemo analizo ujemanja.

2. Za vsako izmed besed imamo možnost, da leži na katerem koli položaju v besedilu. Tukaj pridelamo največjo časovno zahtevnost, saj moramo za vsako besedo pregledati pregledati vse možne položaje.

3. Največji problem drugega dela je, kakšen interval napake nastaviti pri analizi ujemanja. Če izberemo premajhen interval lahko ne dobimo rešitve, če pa določimo prevelik interval bo funkcija računala izredno veliko časa. Če želimo ostati na varni strani in nas zanima le razbitje kriptograma (in ne čas, ki ga za to porabimo) potem lahko nastavimo precej velik interval odstopanja od 75%.

4. Funkcija je tukaj narejena nekoliko pametneje kot v prvem delu, saj za vsako možno odkodirano besedilo takoj vrne, če je pravilno. Torej za vsako odšifrirano besedilo pogleda ali le to vsebuje uganjeno besedno zvezo, ter ga v primeru, da jo, takoj vrne.

Opomba: Rešitev drugega deluje "dobro" za besede srednje dolžine ($10 - 20$ črk). Pri kratkih ($3 - 4$ črke) imamo namreč premalo bitov za primerjavo, pri krajših ($5 - 10$ črk) pa je ocena deleža ujemanja resda boljša, vendar nastane problem zaradi velikega števila ustreznih začetnih zaporedij registrov LFSR1 in LFSR3. Zaradi tega je verjetnost, da bomo začetno besedilo odšifrirali resda večja, vendar bo program deloval počasneje. Če pa za začetek vzamemo zelo dolge nize črk (20+ črk) program spet deluje "dobro", saj z daljšanjem uganjenih besednih zvez deleži ujemanja konvergirajo proti 0.75. Vendar pa tako dolgega zaporedja črk v nekem kriptiranem besedilu skoraj gotovo ne bomo uganili.

Odšifrirano besedilo iz naloge:

CRYPTOGRAPHYPRIORTOTHEMODERNAGEWASEFFECTIVELY
SYNONYMOUSWITHENCRYPTIONTHECONVERSIONOF
INFORMATIONFROMAREADABLESTATETOAPPARENT
NONSENSETHEORIGINATOROFANENCRYPTEDMESSAGE
ALICESHAREDTHEDECODINGTECHNIQUENEEDEDTORECOVER
THEORIGINALINFORMATIONONLYWITHINTENDEDRECIPIENTS
BOBTHEREBYPRECLUDINGUNWANTEDPERSONSEVEFROM
DOINGTHESAMETHECRYPTOGRAPHYLITERATUREOFTENUSES
ALICEAFORTHESENDERBOBBFORTHEINTENDEDRECIPIENT
ANDEVEEAVESDROPPERFORTHEADVERSARYSINCETHE
DEVELOPMENTOFROTORCIPHERMACHINESINWORLDWARI
ANDTHEADVENTOFCOMPUTERSINWORLDWARII
THEMETHODSUSEDTOCARRYOUTCRYPTOLOGYHAVE
BECOMEINCREASINGLYCOMPLEXANDITSAPPLICATION
MOREWIDESPREAD

Zaključek

Drugi primer je veliko daljši od prvega. Ponavadi najlažje uganemo začetni del besedila (včasih tudi končni), sploh če poznamo za kakšno vrsto besedila gre. Sporočila se ponavadi začenjajo z besedami "POZDRAVLJENI", zaključujejo pa z besedno zvezo "LEP POZDRAV IME PRIIMEK". V splošnem lahko uganemo tudi začetne znake nekaterih strukturiranih besedil (HTML značke ali kakšne posebne znake na začetku besedil). V primeru, da tega podatka nimamo se moramo z uganjeno besedno zvezo zapeljati čez celoten seznam ter predvidevati, da se lahko besedna zveza začne na kateremkoli izmed mest v množici $\{0, 5, 10, \ldots, \text{dolžina}(kriptogram) - 5 * \text{dolzina}(besednazveza)\}$. Program bo zato porabil ogromno časa, da bo izračunal vsa možna besedila. Vendar z dovolj dobrim računalnikom ter dovolj časa do rešitve pridemo.