ZH – Kriptográfia és biztonság

2021-05-10

Megjegyzés: Open book zh. Max pont 50, elérendő minimum 20. Az utolsó feladatból két típusból választható. Beadni (canvasba) feltölteni az elkészült programokat (akár egy munkalapot/python notebookot), illetve egyéb fájlokat lehet (pl. beszkennelt/fotózott papír akár). Ahol lehet, röviden kommenteljünk/indokoljunk. Az időkorlát 90 perc.

1. Egy titkosítási algoritmusban a kulcs csak (decimális) jegyekből áll, és annyit tudunk róla, hogy a a kulcs hossza egy 10 és 20 közötti prímszám. Mekkora a kulcstér? Milyen hosszú bináris kulcs biztosítja ugyanezt (vagy nagyobb) biztonságot? [6 pont]

2. Legyen $f(k) = ((k+5)^3)\%113$. Egy olyan shift cipher változatot használunk, ahol a k-adik karaktert f(k) pozícióval tolunk el (csak kisbetűkből áll a szöveg: a-z)

   Implementáljuk, és titkosítsuk a következőket:

   (a) a neptun-kódunk,

   (b) a neptun-kódunk 10-szer megismételve egymás után,

   (c) egy 100 hosszú, csupa "a" betűből álló szöveg.

   Írjunk egy olyan visszafejtő algoritmust, mely az alábbi problémára kínál megoldást. Kaptunk egy ciphertext-részletet, aminek az első néhány (legfeljebb 20) karaktere hiányzik. Annyit tudunk még, hogy a hozzá tartozó plaintextben sokszor szerepel egymás után legalább 4-5 egyforma karakter. Próbáljuk meg kitalálni, hogy hány karakter veszett el az elejéről. Használjuk a fenti csupa ,,a"-s szöveget tesztként.

   [12 pont]

3. Kitaláltunk egy ,,nyelvet":

   (a) Minden üzenet csak az angol ábécé nagybetűiből áll (A-Z).

   (b) Ha van egy C betű, utána mindig jön még 6 darab C betű.

   (c) Van valami ismeretlen 7 hosszú, csak magánhangzóból álló sorozat, ami gyakran ismétlődik, átlagosan 30 karakterből legalább egyszer szerepel ez a sorozat.

   Készítsünk egy olyan generátort, ami ilyen nyelvbeli szöveget állít elő.

   Vigenere-rel titkosítunk, a kulcs legfeljebb 7 hosszú. Hogyan tudnánk megtámadni a titkosítást egy legalább 1000 hosszú szövegen? Próbáljuk ki a saját magunk által generált inputon.

   [14 pont]

4. Keressük meg a 10^6 után következő első prímszámot, jelöljük p-vel. Legyen H az első 10 pozitív prímszám halmaza. A H minden x, y elempárjára keressünk olyan m pozitív egészet, mellyek $x^m\%p == y$ (vagy matekosabban: $x^m \equiv y \pmod{p}$), vagy mondjuk meg, hogy nincs ilyen m.

   [6 pont]

"5A" Titokmegosztási feladatot szeretnénk megoldani, egyszerű XOR (one-time-pad-et utánzó) módszerrel. A titok egy 64 bites 0-1 sorozat. A megosztásban résztvevő emberek pont 26-an vannak, és a kezdőbetűik A, B, ... Z. A titkot akkor tudják rekonstruálni, ha

   (a) Minden magánhangzó összeáll, VAGY

   (b) Minden mássalhangzó összeáll.

   Készítsünk randomizált függvényt, ami adott titok bemenethez elkészíti a 26 titokrészt. Plusz 5 pontért: helyreállító függvény készítése

   [12 pont]

"5B" Titokmegosztási feladatot szeretnénk megoldani, egyszerű XOR (one-time-pad-et utánzó) módszerrel. A titok egy 64 bites 0-1 sorozat. A megosztásban résztvevő emberek pont 26-an vannak, és a kezdőbetűik A, B, ... Z. A titkot akkor tudják rekonstruálni, ha

   (a) legalább egy magánhangzó és egy mássalhangzó összeáll VAGY

   (b) az X nevű és még legalább egy valaki összeáll.

   Készítsünk randomizált függvényt, ami adott titok bemenethez elkészíti a 26 titokrészt. Plusz 5 pontért: helyreállító függvény készítése

   [12 pont]