

IT BIZTONSÁG (VIHIAC01)
HÁZI FELADAT

Kriptográfia

Szerző:
BUTTYÁN Levente



2022. február 22.

Tartalomjegyzék

1. Általános információk	2
2. Feladatok	3
2.1. A 2-négyzet rejtjelező feltörése	3
2.2. Egy, megérett a meggy, kettő...	6

1. Általános információk

Ebben a házi feladat kiírásban két feladat található, melyek a kriptográfia témakörhöz kapcsolódnak. A feladatok megoldásához szükséges háttér információk a feladatok leírásában találhatók.

A feladatok megoldásának beadása a Moodle rendszeren keresztül, egy kvíz kitöltésével történik majd. A kvízben olyan kérdéseket teszünk fel, melyeket a feladatok megoldása ismeretében könnyen meg lehet válaszolni. A kvízt fogjuk pontozni, az lesz az ezen házi feladatra kapott pont. Maximum 10 pont szerezhető így. A megszerzett pontszámot azonban 20%-kal csökkentjük, ha a kvíz kitöltése (azaz a házi feladat megoldásának beadása) a határidő után történik.

A feladatokhoz megoldási ötleteket, tanácsokat is adunk. Ezek segítségével a feladatok bárki számára könnyen megoldhatók. Ezzel a célunk annak elérése, hogy foglalkozz a feladatokkal, és ne valakitől készen megkapott megoldásokat adj be. Ezzel ugyanis semmit nem lehet tanulni és nem is fair azokkal szemben, akik becsületesen dolgoznak.

2. Feladatok

2.1. A 2-négyzet rejtjelező feltörése

Ebben a feladatban a 2-négyzet rejtjelezőt kell feltörni. A 2-négyzet rejtjelező működésének leírása megtalálható például a következő Wikipédia oldalon:

https://en.wikipedia.org/wiki/Two-square_cipher

A feladat

Adott a következő rejtett szöveget, amit a (vertikális) 2-négyzet rejtjelezővel állítottak elő:

```
KTOUOFKERKVZSFARSWFYUCEABLUBEPCNA
RSEPNRTANCBUWHADOWFDUZMSAXPYUOJLC
DIGRPUFCABVMWFBUIOBEMNLLHMTADVRL
ICROUVNTTBABMTVMWFBUIOYUMYKKIGMT
AXONTAOPZMSURUCJMTADVRLICROUVNTTB
ABPULAVMZAXTDIGRPUJSFCZAOC SURVVS
GCOTHCFCPBDPYUOJOSKCLESEKSPOLJG
ADOUKCLESBDIGRPUJHOSKCDFIFYUZAXT
VMYPCRZKGRPUJXWFISISABKARRDEZKIU
VUPBNSPAABNC DW
```

A nyílt szövegről tudjuk, hogy angol nyelvű, és néhány szót ismerünk is belőle:

```
.....C
IRCUMSTANCES.....
.....
.....
.....
.....
.....CIPHERTEXT
INCRYPTOGRAPHY.....
.....
```

Azt is tudjuk, hogy a rejtjelező alsó négyzete (mátrixa) egyáltalán nincs megkeverve. Továbbá, szokás szerint, a Q betűt törölték az (angol) ABC-ből (hogy annak méretét 25-re redukálják), ezért az alsó négyzet így néz ki:

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	R	S	T	U
V	W	X	Y	Z

A feladat a felső négyzet megfejtése, melyről tudjuk, hogy egy kulcsszóval kezdődik, amit az ABC maradék (a kulcsszóban nem szereplő) betűi követnek sorrendben. Például, ha a kulcsszó KEYWORD lenne, akkor a felső négyzet így nézne ki:

K	E	Y	W	O
R	D	A	B	C
F	G	H	I	J
L	M	N	P	S
T	U	V	X	Z

Fejtsd meg a felső négyzetet, majd a négyzetek segítségével dekódold az adott rejtett szöveget! A feladat megoldásának beadásához szükséged lesz a felső négyzetben található kulcsszóra, a visszaállított nyílt szövegre, és a nyílt szöveg eredetileg ismert szavainak betűpárjaihoz tartozó kódolt betűpárookra (a megoldás során ezeket úgyis fel kell írnod).

Figyelmeztetés!

Ha önállóan szeretnél próbálkozni, akkor ne olvass tovább! Ha elakadsz megnézheted az alább javasolt lépéseket. Jó szórakozást!

Tanácsok:

1. Írd le a megadott CIPHERTEXT, IN, CRYPTOGRAPHY, és (C)IRCUMSTANCES nyílt szavak segítségével kideríthető betűpár leképezéseket! Például a CIPHERTEXT szóból a következő leképezések adódnak:

CI --> DF
PH --> IF
ER --> YU
TE --> ZA
XT --> XT

2. Az azonosított betűpár leképezések segítségével állítsd vissza és írd le a felső négyzet potenciálisan hiányos sorait! A sorok sorrendje itt még nem számít.
3. Vond össze az összevonható sorokat, és írd le az így visszaállított hiányos felső négyzetet!
4. Egészítsd ki a hiányos felső négyzetet, rendezd a sorait, és írd le a teljes visszaállított mátrixot!
5. Fejtsd vissza a nyílt szöveget a rejtett szövegből a megfejtett rejtjelező segítségével!

2.2. Egy, megérett a meggy, kettő...

Ebben a feladatban egy olyan rejtett szöveget kell feltörni, amit az AES rejtjelezővel állítottak elő, mégpedig CTR (számláló) módban. A megoldáshoz nem lesz szükség az AES rejtjelezőre magára. Viszont hasznos lehet egy ún. hex editor program, mint pl. a HxD (Windows), az iHex (MacOS), az Xxd vagy a Bless (Linux), amivel bináris fájlok tartalmát (pl. a feladatban adott rejtett szövegeket) is meg lehet nézni. Szükség lesz továbbá bináris string-ek XOR-olására, amihez egy desktop számológépet, egy on-line XOR tool-t¹, vagy bármilyen programozási környezetet (pl. egy Python interpretert) használhatsz. Végül, ha rájössz a megoldási módszerre, akkor érdemes lesz azt egy script-ben vagy programban megvalósítani, és nem kézzel végezni majd el a rejtett szöveg dekódolását; ehhez bármilyen script- vagy programnyelvet használhatsz, amit jól ismersz.

A feladat

Hozzájutottunk egy versenytárs laboratóriumból származó érzékeny dokumentum két verziójához:

```
LabProfile-v1.crypt
LabProfile-v1.1.crypt
```

Sajnos a fájlok² rejtjelezve vannak, de egy régi ismerős, aki pont a versenytársnál dolgozik, megszerezte a programot, amivel rejtjelezték őket:

```
aes_ctr.py
```

Belenézve a programba szomorúan tapasztaljuk, hogy a rejtjelező kulcs nincs belekódolva. De mégis szerencsénk van, mert észrevesszük, hogy a programozó nem volt elég körültekintő (vagy nem értett a kriptográfiához) és rosszul használta a CTR blokkrejtjelezési módot!

Ki tudod használni a hibát és vissza tudod fejteni a nyílt szöveget? Ha sikerül, megszerezheted a szövegben elrejtett FLAG-et (azaz a szöveg FLAG-gel jelölt sorát).

A feladat megoldásának beadásához szükséged lesz a visszaállított nyílt szövegre (benne a FLAG-gel jelölt sorral), és annak bináris változatára, amit pl. úgy láthatsz, ha a visszafejtett fájlt egy hex editorban nyitod meg.

¹Pl: <http://xor.pw/>

²A feladatban szereplő minden fájl egy csatolt mappában található.

Figyelmeztetés!

Ez egy nehéz feladat, ezért alább sok segítséget adunk a megoldáshoz. Ha önállóan szeretnél próbálkozni, akkor ne olvass tovább! Természetesen, ha elakadsz megnézheted az alább javasolt lépéseket. A megoldást pontokba szedtük, így nem szükséges végig olvasnod az egészet: ha valamelyik pont után úgy érzed, hogy onnan már be tudod fejezni a feladat megoldását, akkor fejezd be önállóan! Jó szórakozást!

Tanácsok:

1. Nézz bele a rejtjelezett fájllokba és nézd meg az adott rejtjelező programot!

Nyissuk meg a rejtjelező programot egy editorban! Láthatjuk benne, hogy az AES rejtjelezőt használja CTR módban. Nyissuk meg a két rejtjelezett fájlt is egy hex editor programmal és vizsgáljuk meg a tartalmukat. Észrevehetjük, hogy a két fájl eleje (az első 128 bájt) azonos. Ebből arra a következtetésre juthatunk, hogy a CTR módot nem inicializálták megfelelően, és a két fájl rejtjelezésekor ugyanazt kulcsot és ugyanazt a számláló sorozatot használták. Ellenőrizd az utóbbi hipotézist a rejtjelező program vizsgálatával!

2. Nézd meg a rejtjelezett fájllok méretét!

A fájllok méretéből az látszik, hogy a dokumentum 1-es verziója 16 bájttal hosszabb, mint az 1.1-es verzió. Ebből az az ötletünk támadhat, hogy a két rejtjelezett fájlhoz tartozó nyílt szövegek lényegében azonosak, és az 1.1-es verzió csak annyiban különbözik, az 1-es verziótól, hogy abból 16 bájtot töröltek az első 128 bájt után.

3. Gyárts hipotézist a megfigyeléseidből!

Foglaljuk össze, hogy mit tudunk: (1) a két fájlt valószínűleg ugyanazzal a kulcsfolyammal (ugyanabból a számláló sorozatból származó AES kimenettel) rejtjelezték és (2) a két nyílt szöveg valószínűleg majdnem azonos, csak az 1.1-es verzióból hiányzik egy blokk (16 bájt) a 8. blokk (128. bájt) után. Mikor tudjuk, hogy két rejtett szöveget ugyanazzal a számláló sorozattal állítottak elő CTR módban, akkor az első ötletünk mindig az, hogy XOR-oljuk össze a két rejtett szöveget, mert ha így teszünk, akkor a kulcsfolyam kiesik és a két nyílt szöveg XOR összegét kapjuk (azaz $(X \oplus K) \oplus (X' \oplus K) = X \oplus X'$). Ez felfedhet valamit a nyílt szövegekről. Próbáld ki!

4. Ellenőrizd a hipotézisedet!

Az első 128 bájtot hiába XOR-oljuk össze, csak 0-kat kapunk. Viszont amikor a fájllok 9. blokkjait XOR-oljuk össze, akkor valami érdekeset láthatunk:

```

      35 9C 82 30 0C 60 1E 07 3E F5 D4 34 37 6C EE F7
XOR 56 CE DB 43 55 13 1E 6B 7F B7 D8 34 55 39 AA B6
-----
      63 52 59 73 59 73 00 6c 41 42 0c 00 62 55 44 41

```

Ez a következő szöveg ASCII reprezentációja:

```
c R Y s Y s ? l A B ? ? b U D A
```

Ez úgy néz ki, mintha a „CrySyS Lab...” szöveget szóközökkel (hex 0x20 bájtokkal) XOR-olták volna össze, hiszen tudjuk, hogy a kis és nagy betűk ASCII kódjai között pont 0x20 a különbség!

Próbáljuk meg tehát végig XOR-olni a kapott szöveget a 0x20 0x20 0x20 ... sorozattal:

```

      63 52 59 73 59 73 00 6c 41 42 0c 00 62 55 44 41
XOR 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
-----
      43 72 79 53 79 53 20 4c 61 62 2c 20 42 75 64 61

```

Az eredmény a következő szöveg ASCII reprezentációja:

```
C r y S y S   L a b ,   B u d a
```

Nagyszerű! Ebből megtudtuk, hogy a 9. blokk az 1.1-es verzióban „CrySyS Lab, Buda”, míg az 1-es verzióban a 9. blokk a csupa szóköz. Nem csoda, hogy ezt kitörölték belőle.

5. Általánosítsd az észrevételeidet!

Kicsit formalizáljuk az eddigieket! Jelöljük a dokumentum 1-es verziójának blokkjait a következőképpen: $m[1..8], m[9], m[10], m[11], \dots$. Azt sejtjük, hogy akkor az 1.1-es verzió blokkjai a következők: $m[1..8], m[10], m[11], \dots$, mivel azt feltételeztük, hogy az 1.1-es verzió úgy keletkezett az 1-es verzióból, hogy abból a 8. blokk után egy blokkot töröltek. Amikor kiszámoltuk a 9. blokkok XOR összegét, akkor valójában az 1-es verzióból származó $m[9]$ -et XOR-oltuk össze az 1.1-es

verzióból származó $m[10]$ -zel, ebből kaptuk az $m[9] \oplus m[10]$ -et. Ennek megfigyelése révén kitaláltuk, hogy $m[9]$ egy csupa szöközőkből (0x20 bájtokból) álló blokk, és ebből aztán kijött, hogy $m[10]$ a „CrySyS Lab, Buda” szöveg. Ha most a 10. blokkokat XOR-oljuk össze, akkor az 1-es verzióból származó $m[10]$ és az 1.1-es verzióból származó $m[11]$ XOR összegét kapjuk, azaz $m[10] \oplus m[11]$ -et. Mivel már ismerjük $m[10]$ -et, ezért ebből ki tudjuk számolni $m[11]$ -et a következő módon:

```

      10th blk of v1: 57 D3 82 E3 B3 16 43 E4 26 84 54 F3 FE C3 49 2B
XOR 10th blk of v1.1: 64 C4 88 C4 C0 4F 37 C0 22 C6 14 B2 DE D9 5F 2B
                        -----
      m[10]+m[11]: 33 17 0A 27 73 59 74 24 04 42 40 41 20 1A 16 00
XOR      m[10]: 43 72 79 53 79 53 20 4C 61 62 2C 20 42 75 64 61
                        -----
      m[11]: 70 65 73 74 0A 0A 54 68 65 20 6C 61 62 6F 72 61

```

Az eredmény a következő szöveg ASCII reprezentációja:

```
p e s t      T h e      l a b o r a
```

Tehát jó úton járunk! Most XOR-oljuk össze a 11. blokkokat, ebből megkapjuk $m[11] \oplus m[12]$ -t, és mivel ismerjük $m[11]$ -et, ezért ki tudjuk számolni $m[12]$ -t. Innen már egyszerű látni a sémát, és befejezni a feladat megoldását...

6. Teljes megoldás

Legegyszerűbb, ha script-et vagy programot írsz a fenti módszer automatizált végrehajtására, ami dekódolja a rejtjelezett dokumentumot³.

³A nyílt szöveg első 8 blokkja, azaz 128 bájtja nem fejthető vissza, de ezzel nem kell foglalkozni.