1. **Mivel foglalkozik a kriptográfia?**

A kriptográfia klasszikus értelemben a titkosítás tudománya, titkosítási rendszerek (kriptorendszerek) felépítésével és biztonsági elemzésével (kriptoanalízisével) foglalkozik. Modern értelemben azonban a kriptográfia fogalma magába foglal olyan aktuális témaköröket is, mint a digitális aláírások, hitelesítések, hash függvények stb.

2. **Mi a különbség a kriptográfia és a kriptoanalizis között?**

A kriptográfia azoknak a matematikai eljárásoknak, algoritmusoknak, biztonsági rendszabályoknak kutatását, alkalmazását jelenti, amelyek elsődleges célja az információnak illetéktelenek elől elrejtése. A kriptoanalízis viszont a kriptográfiai rendszerek elemzésével és azok feltörésének kutatásával foglalkozik.

3. **Mondjon két szteganografiai módszert!**

Szteganográfia(adatrejtés, data hiding): ,,A szteganográfia üzenetek elrejtése, tipikusan az üzenetnél nagyobb adathalmazban úgy, hogy az üzenetátadás ténye is rejtve marad a külső megfigyelő számára.

- rabszolga fejbőrére írva (hátránya meg kell várni, míg kinő a haja)

- képben a színeket leíró bájtok alacsony helyi érté ű bitjeiben (szemmel nem látható)

- szórt spektrumú adásban (fehér zajként észleli a külső megfigyelő)

4. **Mik az informatikai biztonság alapvető céljai/szolgáltatásai?**

Az informatikai biztonság alapvető célja, hogy az információkat titkosítva tudjunk továbbítani valamint tárolni, oly módon, hogy biztosítva legyünk affelol, hogy ezek az információk nem lesznek megsemmisítve vagy eltulajdonítva.

5. **Mit jelent a bizalmasság (confidentiality)?**

A bizalmasság azt jeleni, hogy az információkat csak azok érhetik el, akik arra jogosultak. Vagyis a titkos információk nem kerülnek illetéktelen kezekbe.

6. **Mit jelent a hitelesség (integrity)?**

A hitelesség (authenticity) azt jelenti, hogy az üzeneteknek tudjuk az eredetet, tartalmat, küldési idejét, valamint, hogy kitől jött az üzenet, mivel ezek hitelesítve vannak kommunikáció szereplői által. A hitelesség magába foglalja a sértetlenséget (integrity) is, ami azt jeleni, hogy az adatok nem módosíthatóak.

7. **Mit jelent a rendelkezésre állás (availability)?**

A rendelkezésre állás azt jeleni, hogy az adatokhoz hozza tudunk férni bizonyos feltételek mellett ha szükségünk van rájuk.

8. **Mit jelent a letagadhatatlanság (non-repudiation)?**

A letagadhatatlanság azt jelenti, hogy utólag nem módosíthatóak az adatok. Valamint, hogy utólag egyik fél sem tagadhatja le a cselekedeteit vagy kötelességvállalását. A letagadhatatlanság alkalmazásakor ez ilyen vitákat egy megbízható harmadik fel (trusted third party) helyesen el tudja dönteni. (Pl: elektronikus aláírás)

9. **Mondjon három példát a kriptográfia alkalmazási területeire!**

A kriptográfiát több területen is alkalmazzák, például: üzenetek titkosításához (encryption), banki tranzakcióknál, elektronikus aláírás, elektronikus kereskedelem (vevő, bank, bolt – mindenki csak a rá tartozó információkat láthatja).

10. **Mondjon három olyan területet, mely az informatikai biztonság körébe beletartozik, a kriptográfiáéba viszont nem!**

Az antivirusok, a szoftware hibák kivédése, valamint a programozási hibák kivédése az informatikai biztonsághoz sorolhatóak, viszont nem tatoznak a kriptográfia körébe.

11,12. A küldő fél a nyílt szöveget (plaintext) szeretné eljuttatni egy másik félhez.Titkosítja (encrypt) a szöveget,így megkapja a titkosított változatát (ciphertext),amely értelmezhetetlen a kulcs (key) nélkül. A fogadó fél megfejti (decrypt) a kódolt szöveget ez után elolvashatja az eredeti üzenetet. Egy harmadik fél feltöri (break) az üzenetet,ha kriptóanalízissel megfejti a nyílvános csatornán küldött kódolt szöveget.

13, 14, 15.A szimmetrikus kulcsú rendszerek esetén a kódoláshoz és dekódoláshoz használt kulcsok azonosak vagy megkapható az egyik a másikból. Nyilvános kulcsú rendszerek esetén a kulcsok nem azonosak. A kódoláshoz használt kulcs nyílvános,a dekódoláshoz használt nem. Szimmetrikus kulcsú rendszerek esetén nem lehet nyílvános kulcs.

16. Miért van szükség nyilvános kulcsú kriptográfiára?

A szimmetrikus kulcsu kriptografiaban a kulcs cseréjével kapcsolatosan még egy másik probléma is felmerül: n egyén közti titkosított kommunikáció lebonyolításához összesen n(n−1)/2 kulcs szükséges (egy 5 csomópontot tartalmazó hálózatban is már 5•4/2 = 10 kulcsra van szükség – 2.3 ábra). Ha a kulcsok el˝oállítása és célba juttatása költséges, ez nagy többletkiadáshoz vezet, ezenkívül a nagyszámú kulcscsere miatt számottev˝oen megn˝o annak a valószín˝usége is, hogy a támadó fél kezére jut egy vagy több kulcs.
-Ezt a problemat oldja meg a nyilvanos kulcsu kriptografia."

"Azert van ra szukseg, mert ha Bobnak tobb ember is akar irni, akkor mindenkivel meg kell, hogy ossza a kulcsot, ami noveli az eselyet annak, hogy a kulcs rossz kezekbe kerul. Bob megtehetne azt is, hogy minden egyes partnerevel kulon megyezik kulonbozo titkos kulcsokban, de ez nagyon idoigenyes es lehet, hogy mindenkivel nem is tud biztonsagban kulcsot cserelni. A nyilvanos kulcsu kriptografia erre a kellemetlensegre kinal okos megoldast, hiszen Bob partnereinek elegendo csak Bob nyilvanos kulcsat ismerniuk ahhoz, hogy titkositott uzenetet kuldjenek neki. Mindegyik kodolt uzenet dekodolasahoz Bob ugyanazt a sajat titkos kulcsot hasznalja, amit csak o ismer."

17. Mi a teljes kipróbálás (kimerítő kulcskeresés vagy angolul brute-force)? Mondjon rá példát!

Bármely kriptorendszer titkos kulcsait próbálgatással el lehet elvileg találni. Ehhez azonban (a legrosszabb esetben) szisztematikusan az összes lehetséges kulcsot végig kell próbálni. Ezt nevezzük kimerítő kulcskeresésnek vagy angolul brute-forcennak.

Példa Caesar-kód: Mivel a lehetséges kulcsok száma kicsi, kipróbálhatjuk az összes lehetséges kulcsot. Elkészítünk az illető nyelv gyakori szavaiból egy listát. Az a kulcs lesz jó, mellyel a titkosított szöveget dekódolva a legtöbb listabeli szó beazonosítható.

18. Mik a Kerchoff követelmények?

1. Ha elméletileg nem is, a rendszernek gyakorlatilag feltörhetetlennek kell lennie.
2. A módszerrel titkosított üzenetek biztonsága ne függjön magának a módszernek a titkosságától: a módszer leírását ugyanis az ellenség is megszerezheti.
3. A kulcs könnyen megjegyezhető, továbbítható és változtatható kell, hogy legyen (anélkül, hogy azt papírra kellene írni).
4. A kódolt üzenetnek olyan formája legyen, hogy azt táviraton továbbítani lehessen.
5. A rendszer által használt segédeszközök hordozhatóak kell, hogy legyenek, és a kódolás/dekódolás műveletét egyetlen személynek is el kell tudnia végezni.
6. A módszer használata legyen egyszerű (ne kelljen túl sok szabályt, lépést megjegyezni), szellemileg ne terhelje túl használóját.

19. Csoportosítsa a titkosítás elleni támadásokat a támadó rendelkezésére álló információ alapján!

1. Csak a kódolt üzenet ismeretén alapuló támadás: A támadónak csak a kódolt üzenetekhez van hozzáférése, semmit sem tud az eredeti üzenetek tartalmáról vagy a titkosításhoz használt kulcsról.

2. <u>Nyíltszöveg ismeretén alapuló támadás:</u> A támadó ismer bizonyos (P,C) párokat, vagyis ismer bizonyos nyílt üzeneteket a megfelelő kódolt üzenetekkel együtt, ezek az információk viszont adottak. Nem ismert még a titkosításhoz használt kulcs.

3. <u>Választható nyílt szövegen alapuló támadás:</u> Marvin akármilyen általa választott nyílt üzenetet tud titkosítani Alice módszerével (ismeri és használja E_k-t, így ő maga tudja előállítani a (P,C) pá- rokat). Az ilyen típusú támadásoknak a nyilvános kulcsú rendszereknél van nagy jelentősége, ahol a titkosításhoz használt k kulcs nyilvános, tehát a támadó akármilyen nyílt szöveget kódolhat vele.

4. <u>Választható titkosított üzeneten alapuló támadás:</u> A támadó akármilyen általa választott titkosított üzenetnek megkaphatja a dekódolt változatát (tehát gyakorlatilag ismeri és használja $D_{k'}$ -t) anélkül azonban, hogy a k' kulcsot ismerné.

21.Mi a feltétlen biztonság (unconditional security, perfect security)?

Függetlenül a rendelkezésre álló titkos szöveg mennyiségétől, időtől és

számítási kapacitástól a titkosítás nem törhető fel, mert a titkosított szöveg

a kulcs ismerete nélkül nem hordoz elég információt a nyílt szöveg

rekonstruálásához. (ilyen a one-time pad, amikor is a kulcs hossza nagyobb mint a kodolando szoveg)

22. Van-e feltörhetetlen titkosítási algoritmus?

Az általunk ismert titkosítási algoritmusok csak gyakorlatban feltörhetetlenek, elméltileg azonban feltöretőek. Ugyanakkor a ONE TIME PAD elméletileg ilyen.

23.Mi a számítási biztonság (computational security)?

A ma ismert algoritmusokkal, belathato idon belul, lehetetlen megfejteni a titkositasi eljarast

24. Hogyan növelhető egy megbízható titkosítási algoritmus esetén az általa megvalósított titkosság mértéke?

A kriptográfiai algoritmus biztonsága függ

• a választott algoritmus erősségétől

• a kulcs hosszától

Jó algoritmus esetén a kulcshossz növelésével a biztonság növelhető.

Például: Ha egy algoritmus csak teljes kipróbálással (Brute Force) törhető, akkor plusz egy bit kétszeres biztonságnövelést jelent.

25.Mi a hamis biztonság csapdája?

Ha a saját magunk által kitalált es vagy implementált titkosítást erősebbnek gondoljuk, mint amilyen az valójában, akkor veszélyes tévedésben élünk. Inkabb megbizhato implementaciokat hasznaljunk.

26. Mi a különbség a következő fogalmak között: számítási biztonság – feltétlen biztonság?

Számítási biztonság alatt értjük azt, hogyha egy kriptorendszer elméletileg feltörhető, viszont gyakorlatilag nincs rá elegendő idő, hogy bárki is feltörje. A feltétlen biztonság alatt pedig azt értjük, hogy elméletileg feltörhetetlen a kriptorendszer.

27. Mit jelent a csak kódolt üzenet ismeretén alapuló támadás?

A támadónak (Marvinnak) csak a kódolt üzenetekhez van hozzáférése, semmit sem tud az eredeti üzenetek tartalmáról vagy a titkosításhoz használt kulcsról.

28. Mit jelent a nyílt szöveg ismeretén alapuló támadás?

A támadó (Marvin) ismer bizonyos (P,C) párokat, vagyis ismer bizonyos nyílt üzeneteket a megfelelő kódolt üzenetekkel együtt, ezek az információk viszont adottak (például egy kém vagy egy titkos ügynök szerzi be neki). Nem ismeri még a titkosításukhoz használt kulcsot, ezért ő maga nem tud ilyen párokat előállítani.

29. Mit jelent a választható nyílt szövegen alapuló támadás?

A támadó (Marvin) akármilyen általa választott nyílt üzenetet tud titkosítani (vagy titkosíttatni) a küldő (Alice) módszerével (tehát gyakorlatilag ismeri és használja Ek-t, így ő maga tudja előállítani a (P,C) párokat). Az ilyen típusú támadásoknak a nyilvános kulcsú rendszereknél van nagy jelentősége, ahol a titkosításhoz használt k kulcs nyilvános, tehát a támadó akármilyen nyílt szöveget kódolhat vele.

30. Mit jelent a választható titkosított üzeneten alapuló támadás?

A támadó (Marvin) akármilyen általa választott titkosított üzenetnek megkaphatja a dekódolt változatát (tehát gyakorlatilag ismeri és használja Dk´ -t) anélkül azonban, hogy a k´ kulcsot ismerné.

31.Mi a kriptoanalízis? Adjon rá egy példát!

- A kriptoanalizis titkositasi eljarasok elemzesevel es feltoresevel foglalkozik.

1. Pelda: a kodolt szovegrol betugyakorisag statisztikat keszitunk es ezt osszevetjuk a nyelv(pl. angol) betugyakorisagaval.

2. Pelda: brute force = sorra vesszuk az osszes lehetseges esetet
3. Pelda: birtokunkban all tobb kodolt szoveg is es tudjuk azt, hogy mindegyikben szerepel egy kozos resz, pl. nemet idojarasjelentesek esete, ahol tudtak azt, hogy minden szoveg vegen ugyanugy koszontek el.
4. Pelda: ismerunk tobb (P,C) part is, es ezeket osszevetve eszrevehetunk bizonyos osszefugggeseket, mint peldaul ismetlodeseket egy konkret poziciotol egy bizonyos tavolsagra.

32.Definiálja a kriptorendszer fogalmát!

- Kriptorendszernek nevezzuk azt az agat, amely titkositasi eljarasok felepitesevel/szerekesztesevel foglalkozik.

33.Miért kell a kriptorendszer definíciójában az Ek titkosító leképzésnek injektívnek lennie?

- Ha a titkosito fuggveny injektiv, akkor maga az eljaras bijektivve valik, azaz lesz inverze. Ez azt jelenti, hogy egy nyilt szovegnek megfelel egy es csak is egy kodolt szoveg, es egy kodolt szovegnek megfelel egy es csakis egy nyilt szoveg.

34.Definiálja az (eltolásos) Caesar-titkosítót kriptorendszerként!

- A modszer lenyege abban all, hogy a nyilt szoveg betuit egyenkent eltoljuk korkorosen jobbra egy konkret szammal az ABC-ben.

35.Definiálja a monoalfabetikus kódot kriptorendszerként!

- A monoalfabetikus kriptorendszer betunkent kodolja a nyilt szoveget ugy, hogy egy bizonyos betunek mindig ugyanazt a betut felelteti meg. Ilyen pl. a CAESAR kod.

36. Mekkora a kulcstér mérete monoalfabetikus kód esetén?

Monoalfabetikus kód esetén a kulcstér mérete megegyezik a használt ABC betűi permutációinak számával.

Tehát pl. ha az angol ABC-t használjuk (26 betű), akkor a kulcstér mérete 26!.

37. Definiálja az affin titkosítót kriptorendszerként!

Az *affin-rejtjel* lényege szintén egy partikuláris betűpermutáció, tehát az affin-rejtjel is egy monoalfabetikus helyettesítő kód. Akárcsak a Caesar-rejtjel esetében, itt is veszünk egy n betűs ábécét, és a betűket azonosítjuk a Zn-beli elemekkel a 0-tól számított ábécébeli sorszámuk alapján.

Kulcs: $k = (a, b) \in Z2n$ úgy, hogy létezik $a-1$ (mod n), vagyis $(a, n) = 1$.

Kódolás: $C = Ek(P) = (aP + b)$ mod n, ahol $P \in Zn$ (egy betű).

Dekódolás: *P* = *Dk*(*C*) = (*a*−1*C* −*a*−1*b*) mod *n*, ahol *C* ∈ *Zn* (egy betű).

38. Hogyan alkalmazhatók a nyelvi statisztikák a monoalfabetikus helyettesítés kriptoanalízisében?

2. Betűgyakoriság-vizsgálattal elegendő két betű kódolt megfelelőjét beazonosítani. Legyenek ezek (P_1, C_1) és (P_2, C_2). Ekkor

$$\begin{cases} C_1 = (aP_1 + b) \bmod n \\ C_2 = (aP_2 + b) \bmod n \end{cases} \Longrightarrow a(P_1 - P_2) \equiv (C_1 - C_2) \pmod n$$

és ez a kongruencia biztosan megoldható. Legyen $(P_1 - P_2, n) = d$. A következő két eset lehetséges:

(a) Ha $d = 1$, akkor $a \equiv (P_1 - P_2)^{-1}(C_1 - C_2) \pmod n$.

2.1. KLASSZIKUS TITKOSÍTÁSI RENDSZEREK 21

(b) Ha $d > 1$, akkor $a \equiv \left(\frac{P_1-P_2}{d}\right)^{-1}\left(\frac{C_1-C_2}{d}\right) \equiv u \pmod{\frac{n}{d}}$, ahol $u = \left(\frac{P_1-P_2}{d}\right)^{-1}\left(\frac{C_1-C_2}{d}\right)$.

Az a lehetséges értékei tehát:

$$\begin{aligned} a &\equiv u \pmod n \\ a &\equiv u + \frac{n}{d} \pmod n \\ a &\equiv u + 2\frac{n}{d} \pmod n \\ &\vdots \\ a &\equiv u + (d-1)\frac{n}{d} \pmod n \end{aligned}$$

Van tehát összesen d lehetőségünk a-ra.

39. Milyen módszereket lehetne alkalmazni a monoalfabetikus helyettesítés könnyű feltörhetőségének elkerülésére?

A monoalfabetikus helyettesítés könnyű feltöréhetőségének elkerülése érdekében fontos, hogy a használt ABC egy random permutációját használjuk a kódoláshoz.

40. Titkosítsa kulcsszavas Caesar-kóddal a „…” nyílt szöveget a „…” kulccsal (ékezetek nélküli 26- betűs angol ábécét használva).

Kulcsszavas Ceaser-kódolás:

* Felírjuk az ABC egy sorba

* Alája beírjuk a kulcsszót, olyan formában, hogy ha ismétlődik benne egy betű, akkor csak egyszer írjuk le (az első előfordulásakor)

* Ha leírtuk a kulcsszót, akkor az ABC megmaradt betűit (ami nincs a kulcsban) folytatólagosan leírjuk

* Maga a kódolás folyamata: Felső sor betűi -> alsó sor betűi

* Pl. kulcs = ALMA

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

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

41. Fejtse meg a kulcsszavas Caesar-kóddal titkosított „…” üzenetet, ahol a kulcs „…” (ékezetek nélküli 26-betűs angol ábécét használva).

-Ugyanugy mint fenneb, csak visszafele.

42. Hogyan működik a Vigenère-rejtjel (vagy Vigenère-titkosító)?

Kulcs: A kulcs egy szó, melynek egymásutáni ismétlésével megkapjuk a kulcsszöveget. A kulcs hosszát periódusnak nevezzük.

Kódolás: A kódolt szöveg k-adik betujét a következo módon kapjuk meg: megkeressük az eredeti szöveg kadik betujét (mondjuk ez az ábécé i-edik betuje) és a kulcsszöveg k-adik betujét (mondjuk ez az ábécé j-edik betuje); ekkor a kódolt betu a Vigenère-tábla i-edik sorában és j-edik oszlopában levo betu. Vegyük észre, hogy ez megegyezik a tábla j-edik sorában és i-edik oszlopában levo betuvel, hiszen a tábla szimmetrikus a foátlóra nézve.

Dekódolás: Az eredeti szöveg k-adik betujét a következo módon kapjuk vissza: ha a kulcsszöveg k-adik betuje az i-edik az ábécében, akkor a tábla i-edik sorában megkeressük a kódolt szöveg k-adik betujét. Tételezzük fel, hogy ez az i-edik sor j-edik pozíciójában van. Ekkor az eredeti betu az ábécé j-edik betuje.

43. Titkosítsa Vigenère-titkosítással a „…” nyílt szöveget a „…” kulccsal (ékezetek nélküli 26-betűs angol ábécét használva).

Kulcs: LE

Szoveg: ATTA

Kulcsfolyam:LELE

A sor L.ik betuje: L

T sor E.ik betuje: X

T sor L.ik betuje: E

A sor E.ik betuje: E

Kodolt szoveg: LXEE

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

44. Fejtse meg a Vigenère-titkosítással kódolt „…” üzenetet, ahol a kulcs „…” (ékezetek nélküli 26- betűs angol ábécét használva).

Kodolt: LXEE

Kulcsf: LELE

L.ik sor L betuje: A

E.ik sor X betuje: T

L.ik sor E betuje T

E.ik sor E betuje: A

45. Mi a Kasiski-teszt és mire jó?

W. Kasiski módszerevel periódustlehet meghatározni. (A kulcs hosszát periódusnak nevezzük.)

W. Kasiski módszer lényege a következo: a kódolt szövegben ismétlodo szövegrészeket keresünk. Legyen például UCLA egy ismétlodo szövegrész. Megkeressük, mikor a legkisebb két UCLA elofordulás között a távolság, és ekkor a periódus hossza osztja az elso el ofordulás U betujétol (elso betujétol) a második el ofordulás U betujéig (U-t kizárva) a betuk számát.

Vagyis ha: U C L A X U S T A V Z B U C L A

1 2 3 4 5 6 7 8 9 10 11 12 akkor m osztója 12-nek.

46. Hogyan működik a Playfair-kód (vagy Playfair-titkosító)?

Kulcs: A kulcs egy 5 × 5-ös betutábla, melyben az angol ábécé betui szerepelnek a J kivételével. Ezt a betutáblát egy kulcsszó segítségével is ki lehet tölteni a kulcsszavas Caesar-kódnál megismert módon: a kulcsszó (betuismétlések nélkül) a legfelso sor bal sarkából indul balról jobbra és fentrol lefele (ha 5 betunél hosszabb); a kimaradt betuk ábécésorrendben követik a kulcsszót balról jobbra és fentrol lefele.

Kódolás: A kódolás betupáronként történik, ehhez a szöveget betupárokra bontjuk, az esetleges J betuket I-re cserélve. Ha egy betupárban kétszer szerepel ugyanaz a betu, akkor az elso betu után egy X-et szúrunk be. Ha a betuk száma páratlan, akkor a szöveget egy X-szel egészítjük ki a végén. A betupárokat három eset szerint a következo módon kódoljuk:

1. Ha a betupár betui a tábla azonos sorában vannak, akkor a táblában ciklikusan jobbra toljuk oket egy pozícióval.

2. Ha a betupár betui a tábla azonos oszlopában vannak, akkor ciklikusan lefele toljuk oket egy pozícióval.

3. Ha a betupár betui nincsenek sem azonos sorban, sem azonos oszlopban, akkor a táblában az általuk (átellenes csúcsokként) meghatározott téglalap másik két átellenes csúcsába kódolódnak úgy, hogy az eredeti betupár elso betuje és a kódolt betupár elso betuje egy sorban legyenek.

Dekódolás: Ugyanúgy történik, mint a kódolás, csak az elso esetben ciklikusan balra tolunk egy pozícióval és a második esetben ciklikusan felfele tolunk egy pozícióval.

47. Titkosítsa Playfair-titkosítással a „...” nyílt szöveget a „...” kulccsal (ékezetek nélküli 26-betűs angol ábécét használva).

https://en.wikipedia.org/wiki/Playfair_cipher#Example

48. Fejtse meg a Playfair-titkosítással kódolt „...” üzenetet, ahol a kulcs „...” (ékezetek nélküli 26- betűs angol ábécét használva).

49. Hogyan működik a Vernam-titkosító?

A módszer neve angolul onetime pad. Lényege az, hogy az eredeti szöveget szövegegységenként (karakterenként, betunként, bitenként) összeadjuk modulo n a szöveggel azonos hosszúságú kulccsal. Itt n a használt karakterek száma (pl. abc esetében 26).

Kulcs: Az eredeti szöveggel azonos méretu, lehetoleg véletlenszeru szövegegység sorozat, amelyet csak egyszer használunk fel (innen az angol elnevezés). Az ilyen kulcsot még kulcsfolyamnak is nevezzük. Jelöljük k_i-vel a k kulcs i-edik egységét.

Kódolás: $E_k(P) = C$, $C_i = (P_i + k_i) \bmod n$, ahol P_i az eredeti szöveg i-edik egysége, C_i pedig a titkosított szöveg i-edik egysége.

Dekódolás: $D_k(C) = P$, $P_i = (C_i - k_i) \bmod n$.

50. Mit jelent az, hogy az egyszeri hozzáadásos módszer (one-time pad) feltétlenül biztonságos?

Feltétlen biztonság: matematikailag bizonyítottan nem feltörhető; elméletileg sincs támadás ellene

Azt jelenti, hogy a kódolt szöveg nem árul el semmi információt az eredeti szövegrol. Ha a kulcsot többször használjuk fel, akkor a rendszer sebezhetové válik a nyílt szöveg ismeretén alapuló támadással szemben. Rendkívül fontos, hogy egy bizonyos kulcs felhasználása egyszeri legyen.

51. Igaz-e, hogy az egyszeri hozzáadásos módszer (one-time pad) feltétlenül biztonságos? Válaszát inkodolja!

Igaz, mivel ha egy n. hosszúságú szöveget titkosítottunk és a támadó minden kulcsot kipróbál akkor visszakap minden n hosszúságú szöveget amelyek közül nem lehet eldönteni, hogy melyik volt az eredeti kódolatlan szöveg. Pl ha titkosítjuk, hogy igen akkor brute-force támadás esetén a támadó azt is visszakapja majd, hogy alma.

52. Ha az egyszeri hozzáadásos módszer (one-time pad) feltétlenül biztonságos, akkor miért nem elterjedt a gyakorlatban?

Mert nem praktikus, túl sok tárhelyet foglalna a kulcsok tárolása es kicserelese koltseges.

53. Mi a C-36 rejtjelező gép és mi a működési elve (illetve matematikai modellje)? Melyik titkosító generációhoz tartozik?

A C-36 rejtjelező gép egy mechanikus rejtjelező a klasszikus titkosító rendszerekhez tartozik. Matematikai modelljének alapja két mátrix egy cipelő mátrix(M eleme $M_{6,27}(Z_2)$) és egy lépcsőforma.

A cipelő mátrix egy 6*27 es bináris mátrix úgy, hogy minden oszlopban legfeljebb két egyes van. A lépcsőforma első sora 17, a második 19, ..., az ötödik 25, a hatodik 26 darab értéket tartalmaz. A lépcsőformát kiegészíthetjük egy sor ismétlésével, így egy 6*∞ mátrixot kapva. Kulcsnak egy (M,N) mátrixpárt választunk.

54. Milyen hosszú kulcsot vár a szabványosan megvalósított DES és mennyi ebből az effektív kulcsméret? Hogyan értékelhető ez az algoritmus biztonsága szempontjából?

A DES 64 bites kulcsot vár azonban ebből az effektív kulcsméret az 56 bit mivel minden byte utolsó bitje a kulcsban az hibaellenőrzésre van fenntartva. Ezáltal csökken a DES kulcstere (64 - ről 56 - ra) így a biztonsága is.

55. Hogyan szokták megadni a DES S-dobozait? Igaz-e, hogy egyetlen DES S-doboz egy 4 bites bemenetből 6 bites kimenetet készít?

Nem igaz, 6 bites bemenetből készít 4 bites kimenetet. Az S dobozokat úgy szokták megadni, hogy egyenletes eloszlású legyen a tábla (minden sorában egyszer szerepeljen egy szám 0..F -ig) és hogy minél jobban ellenálljon a differenciál - kriptoanalízisnek (a kimeneti differencia különbözzön a bemeneti differenciától).

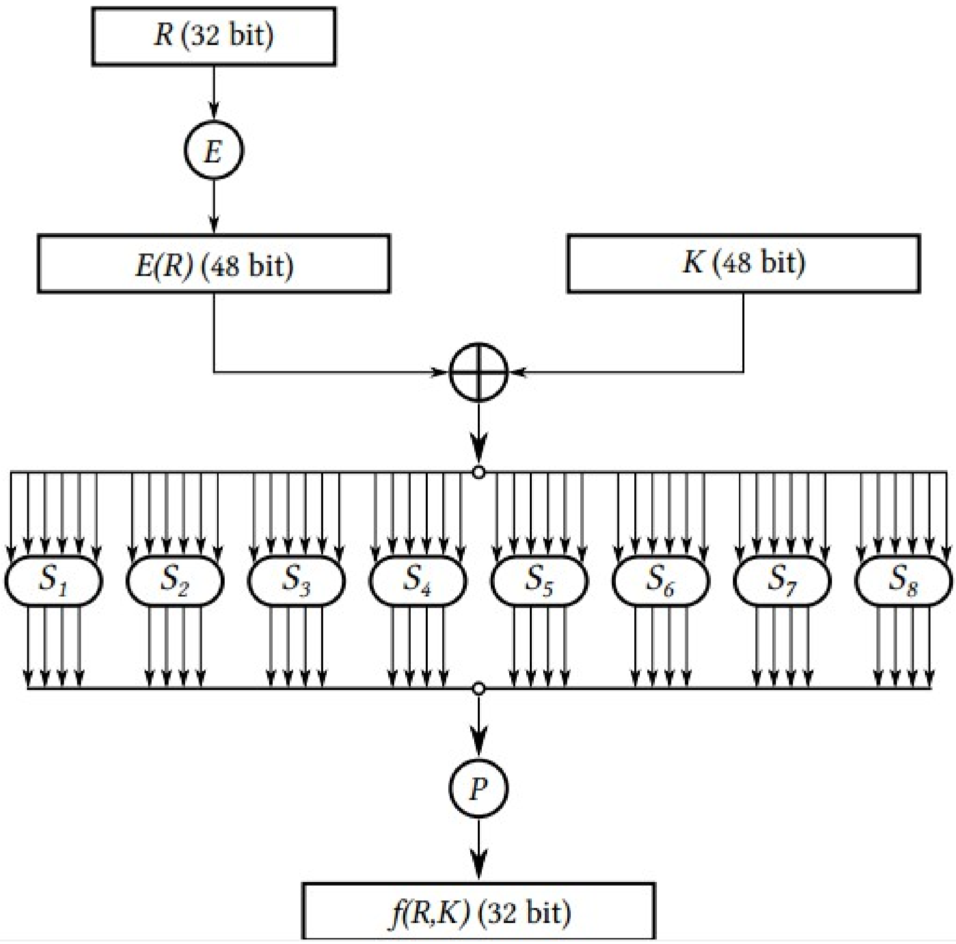
56.Vázolja a DES egy körfüggvényét!

R – jobb fele a kodolando resznek(64/2=32bit)

E – expansion(48 bitet csinal a 32bol)

K – elozoleg eloallitott segedkucs

S_i – i.ik S-box (6bit bol 4-et csinal)   P – a vegen még permutal rajta egyet



57.Hogyan lehet előállítani az AES S-dobozait?

Az AES S-dobozait úgy kell előállítani, hogy egyenletes eloszlást mutassanak és habár az S-boksz differenciál táblája nem lehet tökéletesen egyenletes eloszlású, de legyenek magasan kiugró értékek. Szóval az értékeket minél inkább úgy kell beállítani, hogy a rendszer ellenálljon a differenciál-kriptoanalízisnek.

58.Mi a különbség a szabványosított AES változatok között (AES-128, AES-192, AES-256)?

A különböző változatok eltérnek a kulcsok hosszában, ahogy a nevük is mutatja. Ha N_k–val jelöljük a kulcs szavakban mért hosszát, akkor a lehetséges értékei: 4, 6 és 8. Ezen kívül különbözik a menetek számában is, N_r rendre: 10, 12, 14.

59.Mi a lavinahatás? Rendelkezik-e vele a DES vagy az AES?

Egy kis változtatás az eredeti üzenetben vagy kulcsban a kimenet drasztikus változását eredményezi. Ez a tulajdonságot a hash függvények szigorúbban veszik, de a DES és az AES is rendelkezik velük.

60.Ismertesse röviden a véges testek szerkesztését, legalapvetőbb tulajdonságait, a testbeli műveletek elvégzésének módját!

Ha (K, +, ·) véges test, akkor |K| = p^n, ahol p prímszám

Izomorfizmus erejéig egyetlen q = p^n elemszámú test létezik,egyik alakja

Fq = Zp[X]/(f), ahol f egy n-ed fokú irreducibilis főpolinom Zp[X]-ben.

F_q= = $\{a_{n-1}X^{n-1} + \cdot\cdot\cdot + a_1X + a_0 \text{ mod } f \mid a_i \in Z_p\}$.
A testbeli műveletek az összeadás és szorzás amit mindig a redukálás követ moduló f.

61.Hogyan szerkeszthető meg egy 4 elemű véges test?

Ha p^n=4, akkor p=n=2 és vesszük a polinomokat Z_2-ben. Szükségünk van egy f є Z_2[X]irreducibilis polinomra. Legyen x^2+x+1, és felírjuk a $\{a_1X+ a_0 \text{ mod } f \mid a_i \in Z_2\}$= {0,1, X, X+1} halmazt.

62.Hogyan szerkeszthető meg egy 9 elemű véges test?

Ha p^n=9, akkor p=3, n=2 és vesszük a polinomokat Z_3-ban. Szükségünk van egy f є Z_3[X]irreducibilis polinomra, legyen x^2 +1, és felírjuk a $\{a_1X+ a_0 \text{ mod } f \mid a_i \in Z_3\}$= {0, 1, 2, X, X+1, X+2, 2X, 2X+1, 2X+2} halmazt.

63.Miért előnyös a 256 elemű test használata a gyakorlatban?

A gyakorlatban az információt bináris formában tároljuk, így a blokkok hossza [$\log_2$ 256] = 8. Vagyis pont egy byte. Az AES használ 256 elemű testeket az algoritmusaiban.

64.Milyen véges testek jelennek meg az AES algoritmusában és mi a szerepük?

Az AES algoritmusában megjelenik az F_{2^8}=Z_2[X]/(f) test. A S*ubBytes*, *MixColumns* eljárásoknál és ezek ínverzeinél, az S állapottábla egy-egy oszlopának transzformálásánál kap lényeges szerepet.

65.Mi a szerepe a nyilvános kulcsú kriptográfiában a nyilvános kulcsnak?

A nyilvános kulcsú kriptográfiában a nyilvános kulcs szerepe az, hogy megkerüli a kulcs cserét. Így az illetőnek bárki kódolhat üzeneteket, de megfejteni csak ő tudja saját titkos kulcsával.

66. Mi a szerepe a nyilvanos kulcsu kriptografiaban a titkos kulcsnak?
Az *f* = *Ek* fuggveny értékeit roved ido alatt könnyen kiszámíthatjuk, f^{-1} = *Dk′* értékeit viszont csak nagyon nehezen (nagyon hosszú ido alatt) tudjuk meghatározni, kivéve ha ismerjük a *k′* kiskaput (a dekódolás titkos kulcsát). A kiskapu ismerete f^{-1} = *Dk′* számolását hatékonnyá teszi.

67. Miert hivjak aszimmetrikusnak is a nyilvanos kulcsu kriptografiat?
Azert, mert az *Ek* kódoló eljárás kulcsa is nyilvános, tehát az egyetlen titkos adat a *Dk´* dekódoló eljárás *k´* kulcsa.

68. Mit jelent az, hogy 2048 bites RSA titkositas? Pontosan mi 2048 bites?
n = p*q ahol p es q nagy primek. Ekkor n tekintheto az RSA kulcsanak. Mivel 2010-ben egy 768bit hosszusagu szamot faktorizaltak, igy javasolt legalabb 2048bites n szamot venni. Ezt jelenti a 2048 bit az elnevezesben.

69. Milyen alkalmazasai vannak a nyilvanos kulcsu kriptografianak?
- Ha egy hálózatban *n* személy szeretne páronként titkosan kommunikálni, akkor aszimmetrikus kulcsú rendszer esetében csupán *n* titkos kulcsra van szükség (mindenki kap egy titkos dekódoló kulcsot, a kódoló kulcs nyilvános).
- az úgynevezett digitális aláírásokat csak nyilvános kulcsú rendszerekkel lehet megvalósítani

70. Milyen fuggvenyeken alapszik a nyilvanos kulcsu kriptografia biztonsaga (altalaban)? Milyen tulajdonsagokkal kell rendelkezniuk ezeknek a fuggvenyeknek?

Kell egy $f = Ek$ bijektív függvényt úgy, hogy *f* nyilvános és értékeit rovid ido alatt könnyen kiszámíthatjuk, $f^{-1} = Dk'$ értékeit viszont csak nagyon nehezen (nagyon hosszú ido alatt) tudjuk meghatározni.

Az elobbi tulajdonsággal rendelkezo *f* függvényt *csapóajtó-függvény*nek nevezzük.Az ilyen függvények képezik az alapját valamennyi nyilvános kulcsú rendszernek.

Csapóajtó-függvények esetébenf^{-1} kiszámításának nehézsége általában egy NP-feladatra vezetheto vissza.

71. Konkretan milyen fuggvenyeken alapszik az RSA illetve a Diffie–Hellman rendszerek biztonsaga?

Az RSA eseteben nagyon nagy primek szorzataval dolgozunk, mig a Diffie-Hallman rendszereknel maradekosztalyon belul hatvanyozunk. Ezek adjak a rendszerek kulcsat amelyeket nem lehet visszafejteni mivel faktorizalnunk illetve diszkert logaritmalnunk kellene. Jelenleg emberi idon belul ezeket nem tudjuk kivitelezni.

72. Mi a velemenye a kovetkező allitasrol: „Az RSA biztonsagosabb az AES-nel, mert az RSA nyilvanos kulcsu, az AES pedig titkos kulcsu modszer."
Osszehasonlithatatlan a ketto biztonsagi szempontbol. Mind2 ugyanolyan jo a maga teruleten.

73. Mi a velemenye a kovetkező allitasrol: „A nyilvanos kulcsu modern rendszerek idővel ki fogjak szoritani a szimmetrikus modszereket."
Jelenleg e ket rendszer kez a kezben dolgozik hogy a leheto legangyobb biztonsagot biztositsak. Ketlem hogy valamelyik is kepes lenne a masikat kiszoritani.

74. Mi a velemenye a kovetkező allitasrol: „A modern szimmetrikus kulcsu titkositasok joval gyorsabbak a jelenleg ismert nyilvanos kulcsu modszereknel."

Jap, ez igy van.

75. Mi a velemenye a kovetkező allitasrol: „Az RSA titkositas azert biztonsagos, mert nem ismerunk hatekony algoritmust annak eldontesere, hogy egy szam prim-e vagy sem."

Ez nem igaz. Hatekony algoritmusok vannak egy szam eldonteserol hogy prim-e, pl. Miller-Rabin teszt. Az RSA azert biztonsagos mert nincs hatekony algoritmusunk a faktorizaciora.

76. Ismertesse a Miller–Rabin-tesztet!

A Miller–Rabin-teszt. Legyen n egy nagy (100 számjegyű) páratlan szám. El akarjuk dönteni, hogy n prímszám-e vagy sem. Legyen $n - 1 = 2^s t$ úgy, hogy t páratlan. Választunk egy véletlenszerű b természetes számot úgy, hogy $0 < b < n$ és $(b, n) = 1$. Ezután kiszámoljuk b^t mod n-et. Ha ± 1-et kapunk, akkor n átment a teszt első lépésén, és választunk egy másik véletlenszerű b alapot. Ha $b^t \bmod n \neq \pm 1$, akkor kiszámoljuk sorra b^{2t} mod n, $b^{2^2 t}$ mod n, $\ldots$, $b^{2^{s-1}t}$ mod n-et. Ha valamelyik ezek közül -1, akkor megállunk, és n átment a teszten. Ha viszont egyik sem -1, akkor n elbukta a b alapra a tesztet, következésképpen n biztosan összetett.

Ha n átmegy a teszten k darab különböző alapra, akkor a fenti tétel alapján annak az esélye, hogy n mégis összetett legyen $\left(\frac{1}{4}\right)^k$, tehát n $1 - \left(\frac{1}{4}\right)^k$ valószínűséggel prímszám.

78. Ismertesseaz RSA kulcsgenerálását, titkosítási- ésmegfejtésialgoritmusát!

Kulcs: Válasszunk véletlenszerűen két, legalább 100 számjegyű prímszámot úgy, hogy az egyik (kettes számrendszerben felírva) néhány bittel hosszabb, mint

a másik (lásd a 2. függeléket). Jelölje p és q ezeket a prímeket, legyen $n = pq$ és $\varphi(n) = (p-1)(q-1) = n - p - q + 1$ (az Euler-függvény értéke n-ben). Válasszunk egy e egész számot 1 és $\varphi(n)$ között úgy, hogy $(e, \varphi(n)) = 1$ és e-nek lehetőleg minél kevesebb 1-es bitje legyen. Jó választás e-re például $17 = 2^4 + 1$ és $65537 = 2^{16} + 1$ feltéve, hogy ezek egyike sem osztja $\varphi(n)$-et. A 17 és 65537 is prímszám, tehát ha nem osztják $\varphi(n)$-et, akkor relatív prímek vele. Legyen $d = e^{-1} \bmod \varphi(n)$. Ekkor a nyilvános kulcs $k = (n, e)$, a titkos kulcs pedig $k' = d$.

Kódolás: Tételezzük fel, hogy a szövegünk egy N betűs ábécében íródott. Legyen ℓ olyan, hogy $N^\ell \le n \le N^{\ell+1}$, vagyis $\ell = \lfloor \log_N n \rfloor$. Legyen a P nyílt üzenet egy ℓ betűs tömb. Ez azt jelenti, hogy

$$P = (b_{\ell-1} \dots b_0)_N = b_{\ell-1}N^{\ell-1} + \dots + b_1 N + b_0 < N^\ell \le n,$$

tehát $P \in \mathbb{Z}_n$. Ekkor $C = E_k(P) = P^e \bmod n$, tehát $C \in \mathbb{Z}_n$, $C < n < N^{\ell+1}$, vagyis C egy $\ell + 1$ betűs tömb.

Dekódolás: $P = D_{k'}(C) = C^d \bmod n$.

79. Ismertesse a Merkle–Hellman (knapsack) kriptorendszerkulcsgenerálását, titkosítási-ésmegfejtésialgoritmusát!

Kulcs: Választunk egy szupernövekvő $v = (v_0, v_1, \dots, v_{n-1})$ sorozatot, egy $m \in \mathbb{N}$-t úgy, hogy $m > \sum_{i=0}^{n-1} v_i$ és $a \in \mathbb{N}$-t úgy, hogy $(a, m) = 1$ és $0 < a < m$. Ezeket az adatokat véletlenszerűen generálhatjuk a következő módon. Először kiválasztunk egy véletlenszerű pozitív egészekből álló $n + 1$ hosszúságú $z_0, \dots, z_n$ sorozatot. Legyen $v_0 = z_0$, $v_i = z_i + v_{i-1} + v_{i-2} + \dots + v_0$, $i = \overline{1, n-1}$, $m = z_n + \sum_{i=0}^{n-1} v_i$. Ezután választunk egy szintén véletlenszerű $a_0 < m$ értéket. Legyen a az első pozitív egész úgy, hogy $a \ge a_0$ és $(a, m) = 1$. Ha megvannak az előbbi adatok, meghatározzuk (euklidészi algoritmussal) a $b = a^{-1} \bmod m$-met ($b < m$) és a $w = (w_0, \dots, w_{n-1})$, $w_i = av_i \bmod m$, $w_i < m$ sorozatot. Ekkor a nyilános kódolási kulcs $k = (w_0, \dots, w_{n-1})$, a titkos dekódolási kulcs pedig

$k' = (b, m)$, ahonnan azonnal megvan a és k ismeretében $(v_0, v_1, \dots, v_{n-1})$, hiszen $bw_i = v_i \bmod m$.

Kódolás: A P üzenet most egy n-bites tömb, vagyis $P = (\varepsilon_{n-1}\varepsilon_{n-2} \dots \varepsilon_1\varepsilon_0)_2$. Ha például angol ábécét használó szövegünk van, akkor minden betű az ábécébeli szorszámán alapulva 5 biten ábrázolható:

$$\begin{aligned} \mathbf{A} &\to 0 = (00000)_2 \\ \mathbf{B} &\to 1 = (00001)_2 \\ &\vdots \\ \mathbf{Z} &\to 25 = (11001)_2 \end{aligned}$$

Ekkor $C = E_k(P) = \sum_{i=0}^{n-1} \varepsilon_i w_i \in \mathbb{N}^*$.

Dekódolás: Legyen $V = bC \bmod m$, $V < m$. Ekkor

$$\begin{aligned} V &= bC \bmod m \\ &= \sum_{i=0}^{n-1} \varepsilon_i b w_i \bmod m \\ &= \sum_{i=0}^{n-1} \varepsilon_i b a v_i \bmod m \\ &= \sum_{i=0}^{n-1} \varepsilon_i v_i \bmod m, \end{aligned}$$

80. Ismertesse Shamir háromlépésesprotokollját!

3.3.1. Shamir háromlépéses protokollja

Ezt a titkosítási módszert Shamir 1980-ban dolgozta ki.

Legyen q egy nagy prímszám. Minden X felhasználó választ magának egy titkos $e_X \in \{1, 2, \ldots, q-2\}$ kitevőt úgy, hogy $(e_X, q-1) = 1$, és meghatározza $d_X = e_X^{-1} \bmod (q-1)$-et (euklidészi algoritmussal).

Alice a következő módon küldi el Bobnak a $P \in \{1, \ldots, q-1\}$ üzenetet:

1. lépés: Alice elküldi Bobnak $P^{e_A} \bmod q$-t.
2. lépés: Bob elküldi Alice-nek $P^{e_A e_B} \bmod q$-t.
3. lépés: Alice elküldi Bobnak $P^{e_A e_B d_A} = P^{e_B} \bmod q$-t.

Bob a $P^{e_B} \bmod q$-t dekódolni tudja d_B segítségével, hiszen $P^{e_B d_B} \equiv P \pmod q$. Itt felhasználtuk, hogy $P^{e_A d_A} \equiv P^{1+\ell(q-1)} \equiv P \pmod q$, hiszen a kis Fermat-tételből $P^{q-1} \equiv 1 \pmod q \Longrightarrow P^{\ell(q-1)} \equiv 1 \pmod q$. Hasonlóan $P^{e_B d_B} \equiv P \pmod q$.

81.A Diffie–Hellman-hipotézis

A Diffie–Hellman-hipotézis

Legyen $\mathbb{F}_q$ egy véges test, $q = p^\ell$ elég nagy, $\mathbb{F}_q^* =< g >$ és $a, b \in \{1, \ldots, q-1\}$ véletlenszerűek. Ekkor g, g^a és g^b ismeretéből g^{ab} nem vezethető le, csakis diszkrét logaritmálással.

82.Az ElGamal titkosítási rendszer

3.3.3. Az ElGamal titkosítási rendszer

Ezt a titkosítási rendszert 1985-ben szerkesztette T. ElGamal.

Feltételezzük, hogy $\mathbb{F}_q$ véges test rögzített ($q = p^\ell$ elég nagy) és $\mathbb{F}_q^* =< g >$. Ezek az adatok (g is) mind nyilvánosak.

Kulcs: Minden X felhasználó választ magának egy véletlenszerű $a_X \in \{1, \ldots, q-1\}$ titkos kulcsot, és nyilvánossá teszi $g^{a_X} \in \mathbb{F}_q^*$-ot. Tehát a nyilvános kulcs $k = g^{a_X}$, a titkos pedig $k' = a_X$.

Kódolás: Tételezzük fel, hogy Bob Alice-nek akar küldeni egy $P \in \mathbb{F}_q^*$ üzenetet. Véletlenszerűen választ egy $b \in \{1, \ldots, q-1\}$ egész számot, és elküldi a $(C, C') = (g^b, Pg^{a_A b})$ párt. Vegyük észre, hogy ez kiszámolható, hiszen g^{a_A} nyilvános (ez Alice nyilvános kulcsa).

Dekódolás: Alice kiszámolja $s = C^{a_A}$-t, majd $C's^{-1} = P$-t, és megkapja a nyílt szöveget. Valóban, $C's^{-1} = Pg^{a_A b}(g^{b a_A})^{-1} = P$. Itt fontos megemlíteni, hogy $s \in \mathbb{F}_q^*$-ből $s^{-1} \in \mathbb{F}_q^*$-t könnyen meg lehet kapni kiterjesztett euklidészi algoritmussal.

Más lehetőség: Alice kiszámolja $s' = C^{q-1-a_A}$-t és ekkor $C's' = P$. Valóban, $Pg^{a_A b} g^{b(q-1-a_A)} = Pg^{b(q-1)} = P$, mert $g^{q-1} = 1$.

Ez is a diszkret logaritmalason alapul.

83.Blokktitkosító módok

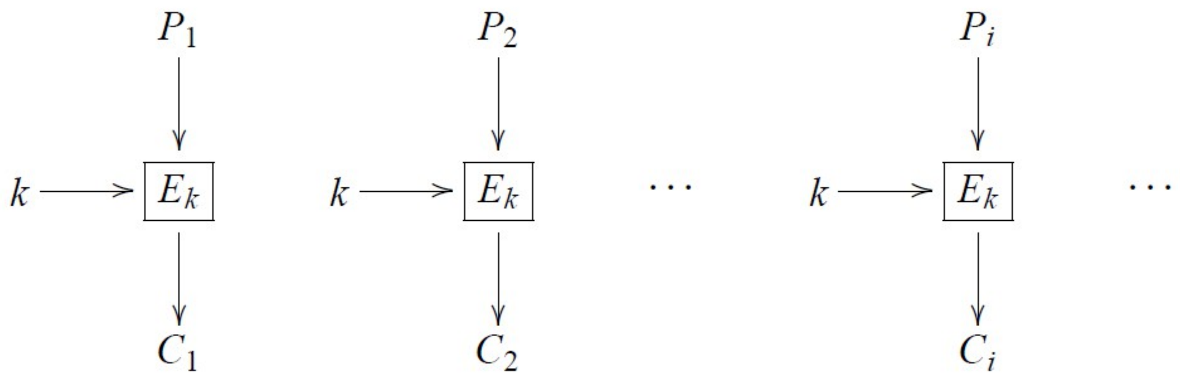
Az ECB mód, a CBC mód, CFB mód, a CTR mód.

84. Mi az ECB mód? Mire alkalmas és mire nem? Miért?

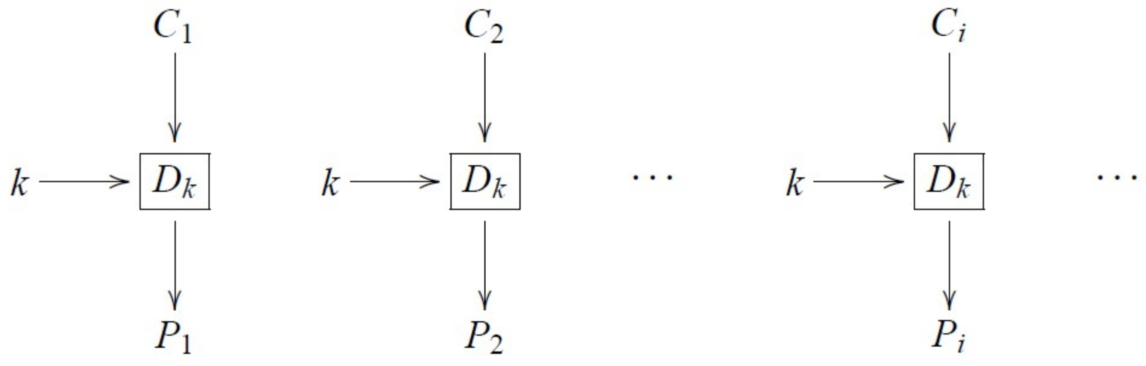
Az ECB mód (Electronic codebook mode) identikus tömböket identikus tömbökbe kódol, (ha $P_i = P_j$, akkor$C_i = C_j$), ami nem takarja el eléggé az eredeti adatstruktúrát. Ezért ezt a működési módot manapság már nem javasolják.

Előnye, hogy párhuzamosítható, ezért gyors, viszont nem mindig biztonságos (nagy adatstruktúra esetén)

Kódolás:$C_i = E_k(P_i)$, minden i tömbre. Grafikusan:



Dekódolás:$P_i = D_k(C_i)$, minden i tömbre. Grafikusan:



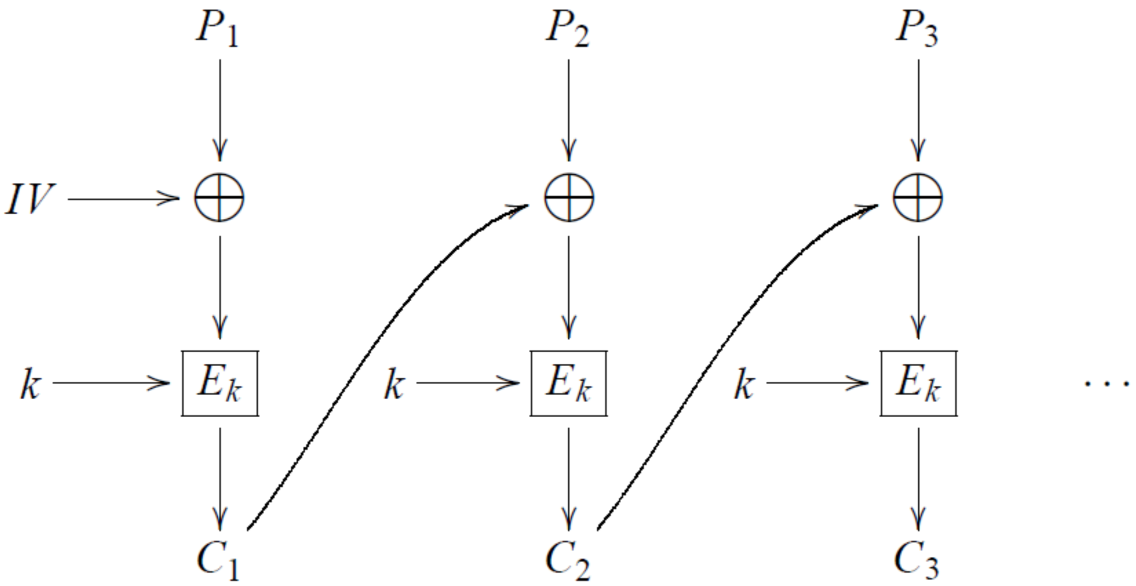
85. Mi a CBC mód? Mik az előnyei?

A CBC mód (Cipher-block chaining mode) az egyik legelterjedtebb működési mód. Hátránya, hogy a titkosítás szekvenciális(nem tehető párhuzamossá). A dekódolás viszont párhuzamosan is megvalósítható,hiszen P_i csak C_i-től és C_{i-1}-től függ. Egy bit változtatása a P_i tömbben megváltoztatja a C_j-ket, minden $j \geq i$-re, egy bit változtatása C_i-ben megváltoztatja az egész P_i-t és egy darab bitet P_{i+1}-ben.

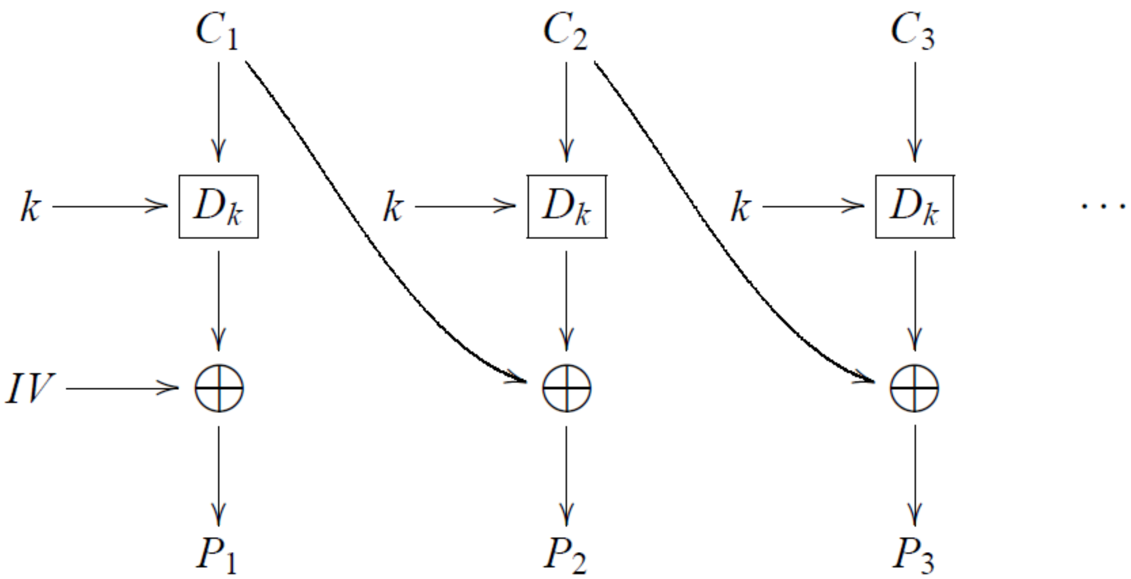
Előnye a biztonság, de lassúbb, mint az ECB, mivelnem párhuzamosítható.

Legyen egy IV kezdeti tömbünk.

Kódolás:$C_0 = IV$, minden további C_i tömbre: $C_i = E_k(P_i \oplus C_{i-1})$, $\forall i > 0$. Grafikusan:



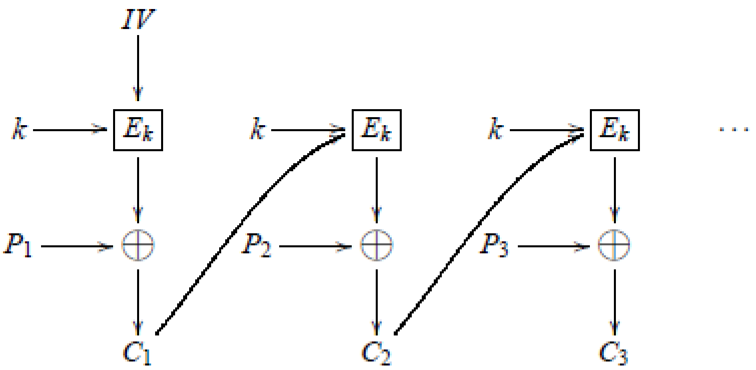
Dekódolás:$P_i = D_k(C_i) \oplus C_{i-1}$, $C_0 = IV$. Grafikusan:



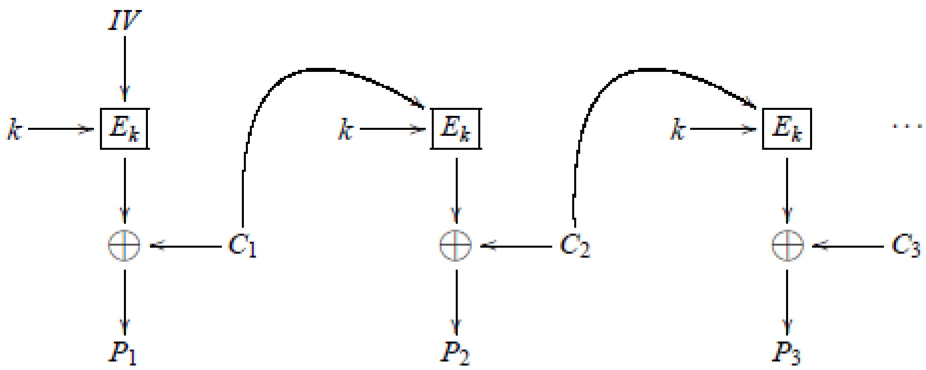
86. Mi a CFB mód és milyen előnye van a CBC móddal szemben?

A CFB mód a tömbtitkosítót átalakítja egy folyamtitkosítóvá.

Kódolás: $C_i = E_k(C_{i-1}) \oplus P_i$, $C_0 = IV$. Grafikusan:



Dekódolás: $P_i = E_k(C_{i-1}) \oplus C_i$, $C_0 = IV$. Grafikusan:



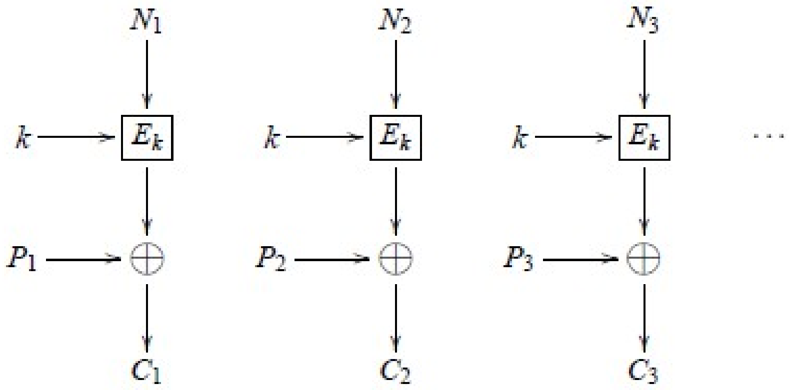
A CFBmód hasonlít a CBCmódra abban, hogy kódolása nem, viszont dekódolása párhuzamossá tehető. Ugyanakkor, egy bit változtatása P*i*-ben megváltoztat minden C*j*-t, bármely *j* ≥ *i*-re. Egy bit változtatása C*i*-ben egy bitet módosít P*i*-ben, és megváltoztatja az egész P*i+1*-et.

87. Mi a CTR mód? Mik az előnyei?

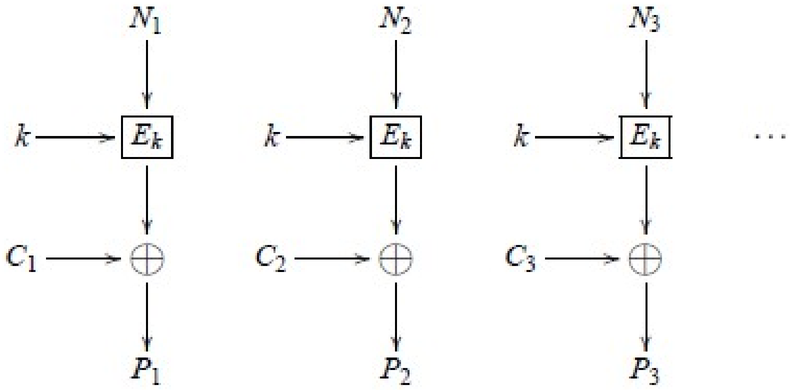
Counter mode

Ha ||-sal jelöljük a bittömbök egymásután való illesztését (konkatenálását), akkor:

Kódolás: $C_i = P_i \oplus E_k(N_i)$, ahol $N_i = IV \parallel \underbrace{00\ldots0i}_{\text{számláló}}$. Grafikusan:



Dekódolás: $P_i = C_i \oplus E_k(N_i)$, ahol $N_i = IV \parallel \underbrace{00\ldots0i}_{\text{számláló}}$. Grafikusan:



A CTR mód nagy előnye, hogy alkalmas párhuzamos kódolásra és dekódolásra, tehát nagyon gyors. Azonban itt is vigyáznunk kell, hogy az *IV* kezdeti vektort ne használjuk újra ugyanazzal a kulccsal.

88. Mi az üzenet helykitöltése (padding)? Miért van rá szükség?

Padding (kitöltés): az eredeti üzenet kiegészítése bizonyos számú bittel úgy, hogy az üzenet bithossza a rögzített tömbméret egész számú többszöröse legyen (tömbtitkosítók esetében).

Tipikus kitöltési módok tömbtitkosítóknál:

1. 0 értékű bitekkel pótolunk;
2. (DES módszer) 1 értékű bit, majd 0 értékű bitek;
3. (Schneier, Ferguson) *n* bájttal egészítjük ki az üzenetet, melyek értéke *n*;n

89. Mi a különbség a blokk- és a folyamtitkosítók (stream cipher) között? Mi a folyamtitkosítók általános működési elve?

A folyamtitkosítók az eredeti szöveg egységeit (betűit, bitjeit vagy bájtjait) egyenként titkosítják, míg a tömbtitkosítók adott méretű adattömböket titkosítanak.

A folyamtitkosítók működése hasonló a one-time pad-hez: a nyílt szöveg hosszával megegyező hosszúságú kulcsfolyam (key stream) elemei a nyílt szöveg elemeivel XOR-olva vannak, egyszerre egy elemet titkosít.
Egy kulcs(folyam)ot itt is csak egyszer szabad használni.

90. Hogyan működik a Diffie–Hellman-kulcscsere és mire jó?

Figyelembe véve a nyilvános kulcsú rendszerek előnyeit és hátrányait , a legjobb, ha a tulajdonképpeni titkosítást szimmetrikus rendszerrel végezzük (pl. DES, AES), de a kulcskezelést (kulcscserét) aszimmetrikus kulcsú titkosítási rendszerrel valósítjuk meg.
A *Diffie–Hellman-kulcscsere* protokollt kifejezetten ehhez szerkesztették 1976-ban. Ebben jelenik meg először a nyilvános kulcs fogalma is. Előnye, hogy a két kommunikáló fél megegyezhet az általuk használt szimmetrikus titkosítási rendszer közös kulcsában anélkül, hogy ezt a kulcsot valamilyen formában el kellene küldeni**.**

Legyen $\mathbb{F}_q$ egy nyilvános véges test (q elég nagy) és g egy nyilvános generátora $\mathbb{F}_q^*$-nak, vagyis $\mathbb{F}_q^* =< g >$. Minden X felhasználó véletlenszerűen választ egy $a_X \in \{1, \ldots, q-1\}$ titkos elemet (ez lesz a titkos kulcs), és nyilvánossá teszi $g^{a_X} \in \mathbb{F}_q^*$-ot (ez a nyilvános kulcs). Hogyan egyezik meg Alice és Bob az (E_k, D_k) szimmetrikus kulcsú rendszer k közös kulcsában?

Alice titkos kulcsa $k'_A = a$, nyilvános kulcsa $k_A = g^a$. Bob titkos kulcsa $k'_B = b$, nyilvános kulcsa $k_B = g^b$. Ekkor a közös titkos kulcs $k = g^{ab}$. Ezt Alice és Bob is ki tudja számolni (Alice $(g^b)^a$, Bob pedig $(g^a)^b$ módon), de egy harmadik fél – Marvin

– már nem, csak diszkrét logaritmálással, feltéve persze, hogy teljesül a már említett Diffie–Hellman-hipotézis (g, g^a és g^b ismerete csak diszkrét logaritmálással vezet g^{ab} ismeretéhez). Természetesen $k = g^{ab} \in \mathbb{F}_q^*$, viszont ez megfeleltethető egy q-nál kisebb természetes számnak a következő módon: ha $q = p^n$ (lásd a 3. függeléket), akkor $\mathbb{F}_q = \mathbb{Z}_p[X]/(f) = \{a_{n-1}X^{n-1} + \cdots + a_1X + a_0 \bmod f \mid a_i \in \mathbb{Z}_p\}$.

Egy $a_{n-1}X^{n-1} + \cdots + a_1X + a_0 \bmod f$ elemnek egyértelműen megfelel $a_{n-1}p^{n-1} + \cdots + a_1p + a_0 \in \mathbb{Z}$ érték, ahol $a_i \in \{0, \ldots, p-1\}$, mitöbb, $0 \le a_{n-1}p^{n-1} + \cdots + a_1p + a_0 < p^n = q$. Az így nyert természetes számból bármely szimmetrikus rendszer titkos kulcsa valamilyen módon felépíthető.

91.Mik a hash függvények (lenyomatkészítő függvények) és hogyan használhatók üzenetek hitelesítésére?

A Hash függvények olyan nem bijektív függvények, melyek egy tetszőleges hosszúságú halmazbol vett bemenetre, egy rögzített hosszúságú elemet adnak meg kimenetként.

f: M->Mr hash ha M tetszőleges hosszúságá elemek halmaza és Mr az r hosszúságú elemeké.

f(M) = MD (lenyomat)

A hash függvényeket úgy használják hitelesítésre, hogy az elkűldött üzenettel együtt, elkűldik annak hash lenyomatát. Aki megkepja az üzenetet, le tudja ellenőrizni, hogy hiteles üzenetet kapott-e, méghozzá úgy, hogy lehash-eli a kapott üzenetet és a kapott lenyomatot hasonlítja a kapott hashlenyomathoz.

92.Ismertesse a hash függvények szerkesztéséhez használt Merkle–Damgård-konstrukciót!

A **Merkle–Damgård** konstrukció szerint:

t hosszúságú tömbökre osztjuk a kapott üzenetet, ha nem egész számú többszörös akkor padding-eljük.

1. Lépés: **Padding**: 1 db 1-es, néhány db 0-ás és az egész végén az eredeti bemenet hossza.
   Mp = M || 1 || 000...0 || L(M)
   A bemenet a t egész számú többszöröse kell legyen.
2. Lépés: **Darabolás:** Mp = B1 || B2 || ... || Bn – t hosszúságú töbszakaszok
3. Lépés: **Rekúrzió:** Szükséges egy f: Mt x Mr –> Mr függvényre, illetve egy r hosszúságú initial value-ra (IV).
   Ekkor :
   H0 = IV -> H1 = f(B1,H0) -> H2 = f(B2,H1) -> ... -> Hn = f(Bn,Hn-1) = MD

93.Mit jelent egy hash függvény gyenge ütközésmentessége?

Egy hash függvény gyengén ütközésmentes, ha ismerünk M, M' párt úgy, hogy f(M) = f(M') = MD de nem tudunk készíteni tetszőleges M-re M'-et, úgy, hogy f(M) = f(M') = MD. Vagyis tetszőleges bemenetre nem tudunk gyártani megfelelő, az eredetivel akár ellenkező tartalmú bemenetet, úgy, hogy ugyanaz legyen a hash lenyomatuk.

94.Mit jelent egy hash függvény erős ütközésmentessége?

Egy hash függvény erősen ütközés mentes, ha nem ismerünk egyetlen, olyan M, M' párt sem, amelyre f(M) = f(M') = MD. Vagyis, nem ismerünk egyetlen olyan bemenet part sem melyre megegyeznének a kimenetek.

95.Ismertesse a születésnap-paradoxonra alapuló támadást egy 64-bites hash függvény ellen!

A születésnapi támadás a születésnap-paradoxonon alapúl, mely megadja, hogy milyen eséllyel van egy embercsoportban két olyan ember, akinek egy időben lenne a születésnapja. Ezt a módszert alkalmazva 64-bites. hash függvény ellen, körülbelül 2^(64/2) = 2^32 pár kipróbálása után, nagyon valószínű, hogy ütközést kapunk. (A képletet Szántó mondta 11. Kurzus 2015.12.14 Generikus tamadasok hash fgv-ek ellen)

96.Biztonságosabb-e az SHA-1, mint az MD5? Miért?

Az SHA-1 több szempontból is biztonságosabb, mint az MD5. Elősször is még nem volt feltörve, még csak az erős ütközésmentességét sem sikerűlt megdönteni. Ezzel szemben az MD5-re már rengeteg párt ismerünk, és újebbakat tudunk előállítani. Ezen felül az SHA-1 kimenete nagyobb(160 bit) mint az MD5 kimenete(128 bit) ezért időigényesebb klasszikus módszerekkel ütközéseket találni.

97.Melyek a digitális aláírás feladatai?

A digitális aláírás feladatai, többek közt, az üzenet hitelesítése, integritásának ellenőrzése illetve az üzenőpartner azonositása.

98.Melyek a digitális aláírás tulajdonságai?

Egy digitális aláírás egy nyílvános kúlcsu kriptorendszert és egy hash függvényt alkalmazva egy szövegre, kap egy aláírást, ami egy titkosított adat. Ez az aláírás kell tudja hitelesíteni a kűldőt, ezen felül, tartalmaznia kell az eredeti szöveg hash lenyomatát is az integritás checking-eléshez.

99.Ismertesse a digitális aláírás szerkesztésének általános elvét!

C = EB (P || **DA (EB (h(P)))**), ebből a vastagjtott rész az aláírás, ahol

C – az az üzenet melyet a fogadó kap, ez egy kódolt üzenet és tartalmazza a digitális aláírást is

EB – a B (fogadó) nyílvános titkositási módszere

DA – az A (kűldő) titkos megfejtési módszere

P – a nyílt szöveg

h – a használt hash függvény

100. Mit tartalmaz egy digitális (nyilvános kulcs-) tanúsítvány? Ki állít ki digitális tanúsítványokat?

A digitális tanusítvány a következőket tartalmazza: Verziószám, Szériaszám, A kiadó hatóság adatai, Érvényességi adatok, Az alany adatai, A kúlcs adatai, Melléklet, A hatóság aláírása, A hash típusa, A titkosító eljáras típusa.

Bármilyen megbízható hatóság vagy cég kiállíthat digitális tanusítványokat.