



Adatbiztonság, adatvédelem

A DES működése és feltörése



A DES

- 4. generációs titkosítási algoritmusok őse
- DES = Data Encryption Standard
- 1976-ban állt munkába
- Pályázat során alkották meg
- 1997-ben sikerült először feltörni
- 2001-ben váltotta le az AES (Advanced Encryption Standard)
- Tervezésében részt vett az NSA



A DES

- Kifejlesztése az 1970-es évek elején kezdődött az IBM Lucifer nevű titkosító algoritmusával
- Több megoldás volt akkoriban is titkosításra, de mindenki által elfogadott szabvány nem létezett.
- A káosz megelőzése miatt az NBS (National Bureau of Standards - mai nevén NIST) pályázatot írt ki.



A DES

- Az IBM a Lucifer algoritmusával nevezett
- A pályázat elvárásai:
 - Nyújtson magas szintű biztonságot.
 - Egyszerű felépítésű, könnyen megérthető legyen
 - A biztonság csak a kulcstól függjön, ne az algoritmustól
 - Gazdaságosan alkalmazható legyen elektronikus eszközökben



A DES

- A Lucifer nyerte a pályázatot, de a szabványosított változat tervezésébe már az NSA is beleszólt
- Eredetileg 128 bites titkosítás volt 128 bites blokkokkal



A DES szabvány

- 64 bites titkosítás
- 64 bites blokkokban dolgozik egy 64 bites kulcs segítségével
- A kulcs valójában „csak” 56 bites, mivel a kulcs minden bájtjának utolsó bitje paritás bit.
- Teljesen nyílt szabvány, így az algoritmust mindenki megismerheti
- Tehát az adatok védelme csak a kulcs bonyolultságától függ.

A DES szabvány

- Bizalmatlanul fogadták kezdetben, mivel konkrétan felezték a bitek számát, ami gyakorlatban azt jelenti, hogy az eredeti kulcstér 99,6%-a ki lett dobva.
- Összeesküvés elméletek szerint erre azért volt szükség, hogy az NSA gond nélkül meg tudja törni, de más kisebb csoportok ne.
- Ez a bizalmatlanság döntő szerepet játszott a PGP megszületésében, de erről majd később...



A dES szabvány

- Bitszintű műveletekkel dolgozik, ezért hardverből nagyon egyszerű implementálni
 - Gyakorlatilag egy 64 bemenetű és 64 kimenetű kombinációs hálózat
- Éppen ezért a szabvány elfogadása után számos integrált áramkör született meg, ami DES titkosítást tudott.



A DES szabvány

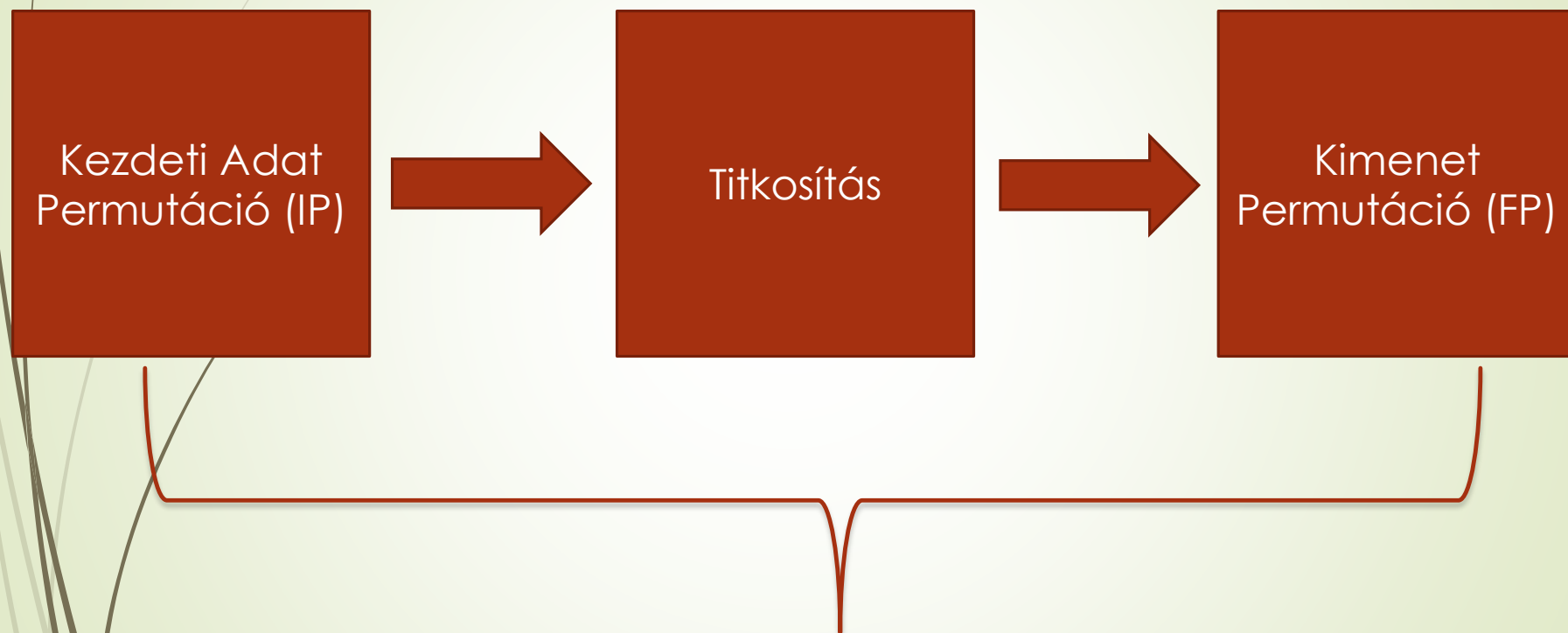
- A hardveres implementációkat az NBS bevizsgálta
- A szabványt 5 évente felülvizsgálták biztonság szempontjából egészen 1997-ig.
- A szabvány megjelenése után és a kezdeti bizalmatlanság miatt célba vették, megpróbálták feltörni, de ez csak 1997-ben sikerült.



A DES szabvány

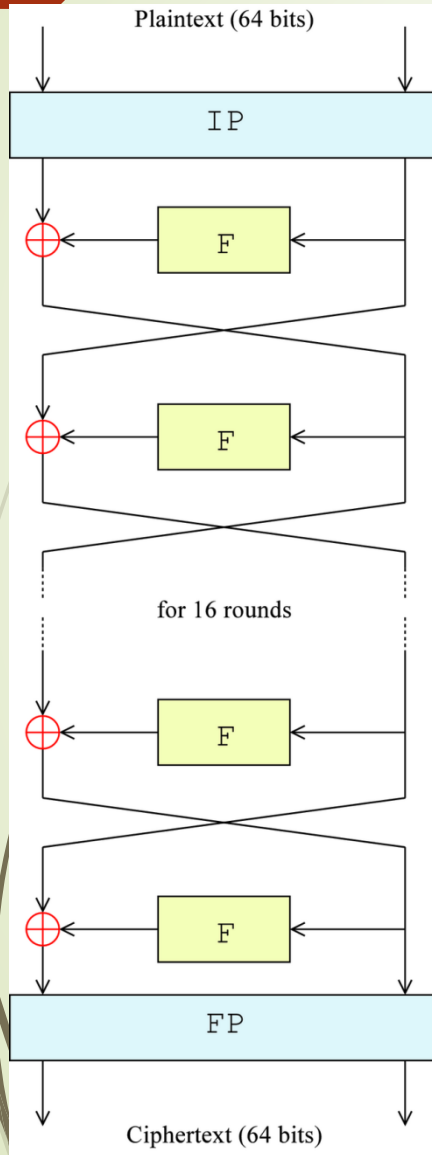
- A DES, mint szabvány a DEA (Data Encryption Algorithm) algoritmust használja.
- Az évek során a két kifejezésből a DES terjedt el, így ma már „DES”-t mondunk, ha az algoritmusról beszélünk
- Szabványban és a hivatalos dokumentumokban a két kifejezés között különbséget tesznek.

A Des működése



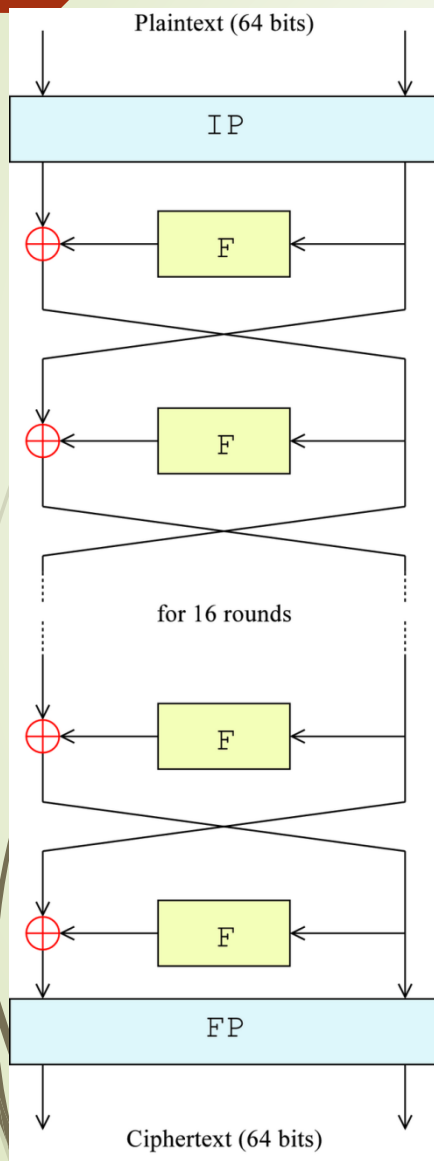
Egymás hatását kioltják, titkosításban nem sok szerepe van.
Optimalizációs céllal került az algoritmusba az 1970-es évek gépei miatt

A Titkosító algoritmus



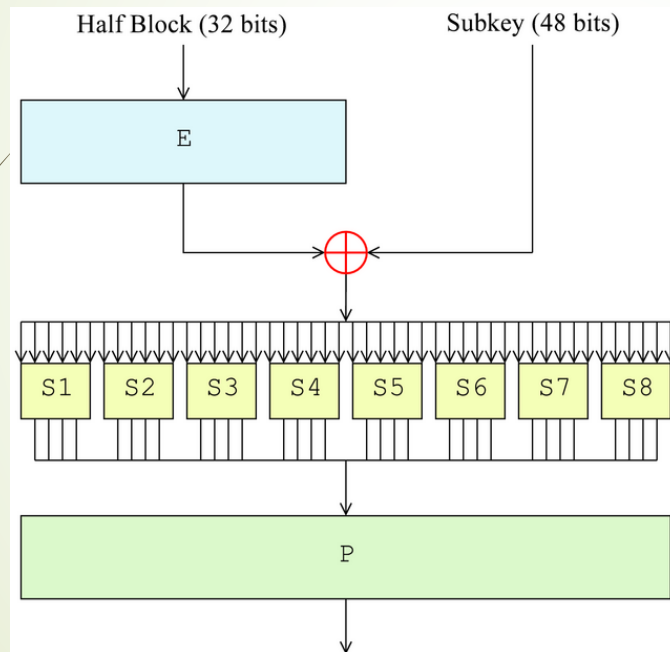
- A bemeneti 64 bit hosszú blokk a titkosítás folyamán 2db 32 bites blokként van kezelve
- A 2db blokk ugyanazzal a funkcióval van titkosítva
- A titkosítási algoritmus 16 azonos körből áll.
- XOR – al vannak összegezve az egyes körökben kapott blokkok, amelyek cserélődnek folyamatosan az algoritmus során

A Titkosító algoritmus



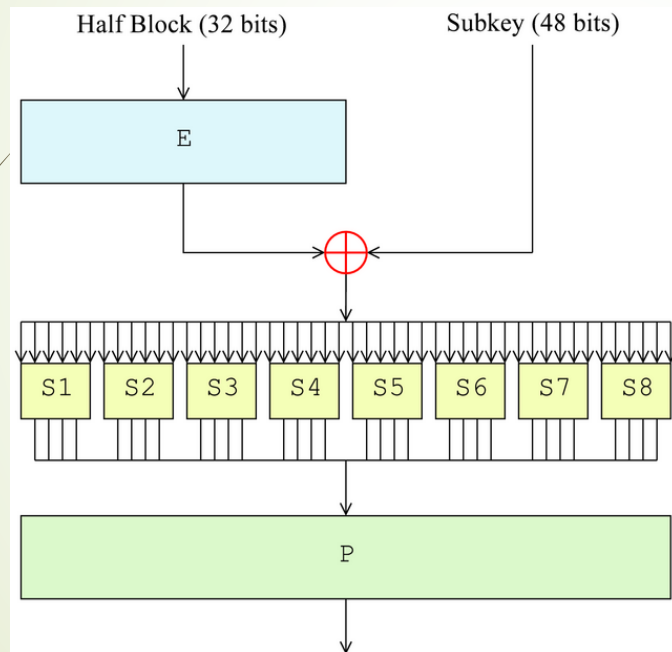
- Utolsó kör után a két blokkot megcserélik, ami a visszafejtéshez kell.
- Az utolsó körben alkalmazott csere és F funkció felépítése miatt fejthető vissza ugyanazon algoritmussal

Az F vagy Feistel funkció



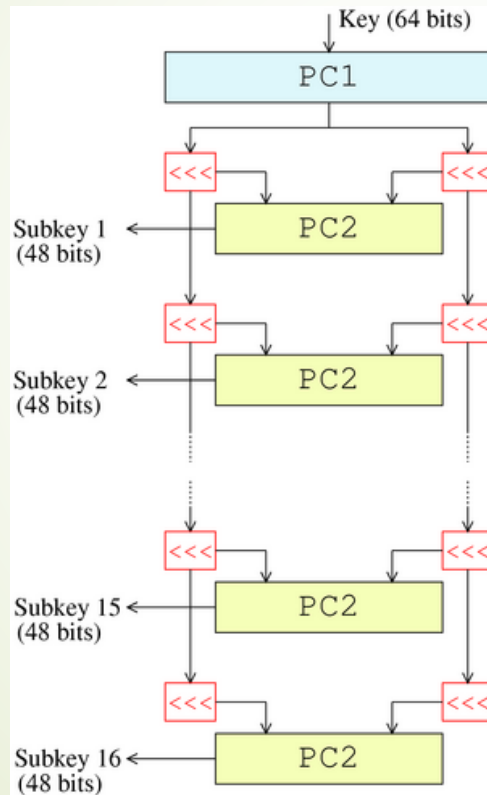
1. Bemeneti 32 bit
48 bitre bővítése
(E), a bitek
felének
duplázásával
2. Kulcskeverés:
alkulcs XOR
adat elven.
Minden F
híváskor (16 van
összesen) más
az alkulcs.

Az F vagy Feistel funkció



1. XOR után az adat 8×6 bitre van osztva. A 6 bit egy táblázat alapján cserélődik 4 bitre. A 6 bitből nem lineáris módon lesz 4 bit.
2. Végző permutáció a 8×4 bit kimeneten

Kezdeti kulcsból alkulcsok előállítása



1. Paritás leválasztása a 64 bitből. Eredmény: 56 bit.
2. 56 bit 2x24 bitre osztása
3. Bit eltolások 1 vagy 2 bittel.
4. Kimeneti 48 bit a 2db 24 bites szám permutációjaként áll elő

A Des működése

- Működése során két fő elvet egyesít:
 - Feistel-struktúra
 - Produkciós titkosító
- A produkciós titkosító kettő vagy több eltérő elvű művelet kombinálásával szolgáltatja eredményét.
- Ha azonos elvű titkosítókat kötünk sorba, előfordulhat, hogy azok egymás hatását kioltják vagy a biztonságot nem növelik, csak a feldolgozási időt



A Des működése

- Emiatt elfogadott az a tervezési elv, hogy a produkciós részegységek egymástól eltérő elven működjenek.
- Egyik speciális eset a helyettesítő-keverő hálózat, mely helyettesítéseket és keveréseket végez egymás után

Lavinahatás

- A lavinahatás elve azt mondja ki, hogy ha a bemeneti blokk kicsit megváltozik, akkor a kimeneti blokk jelentősen változzon meg hozzá képest.
- Pontosabban, ha a bemeneti blokk egy bitje megváltozik, a kimeneti blokk bitjeinek körülbelül a fele változzon meg.
- Ez nehezíti a kriptóanalízist.
- A DES rendelkezik lavinahatással.

A DES biztonsága

- 56 bit kulcs, nagyjából $7,21 \cdot 10^{16}$ kulcs lehetőség
- Nyers, optimalizálatlan Brute Force - al ha 1 millió kulcsot próbálunk ki 1 mp alatt, akkor is ~1150 év lenne megtörni.
- Speciális Cél Hardver segítségével Brute Force támadással 1998-ban törték meg először pár napon belül.

A DES biztonsága

- Az 1990-es évek elején fejlődött annyit a kriptográfia, hogy ki tudták következtetni:
 - egy kulcsot 2^{37} - 2^{38} -on ismert bemenet és ismert kimenet mintából ki lehet találni.
- De az elmélet más, mint a gyakorlat. Bitek szintjén védett, tehát az algoritmus nem hibás, de viszonylag kicsi a kulcsméret, ami kellő erőforrással törhető lesz.



Brute Force Célgéppel

- 1991-ben már voltak rá tervek
- Akkor durván 1 millió \$-ra becsülték az építés költségét.
- Elvben 3,5 óra alatt tudta volna visszafejteni a kulcsot.
- Sosem épült meg pénzhiány miatt.

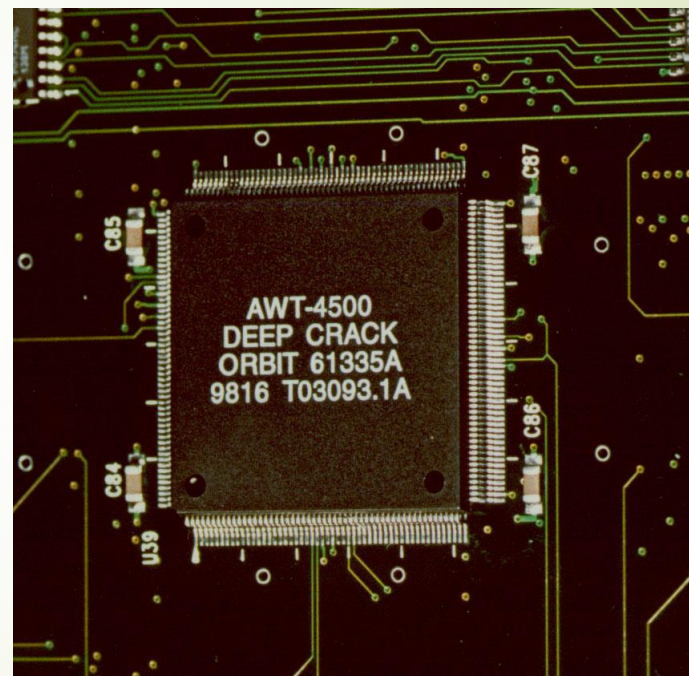
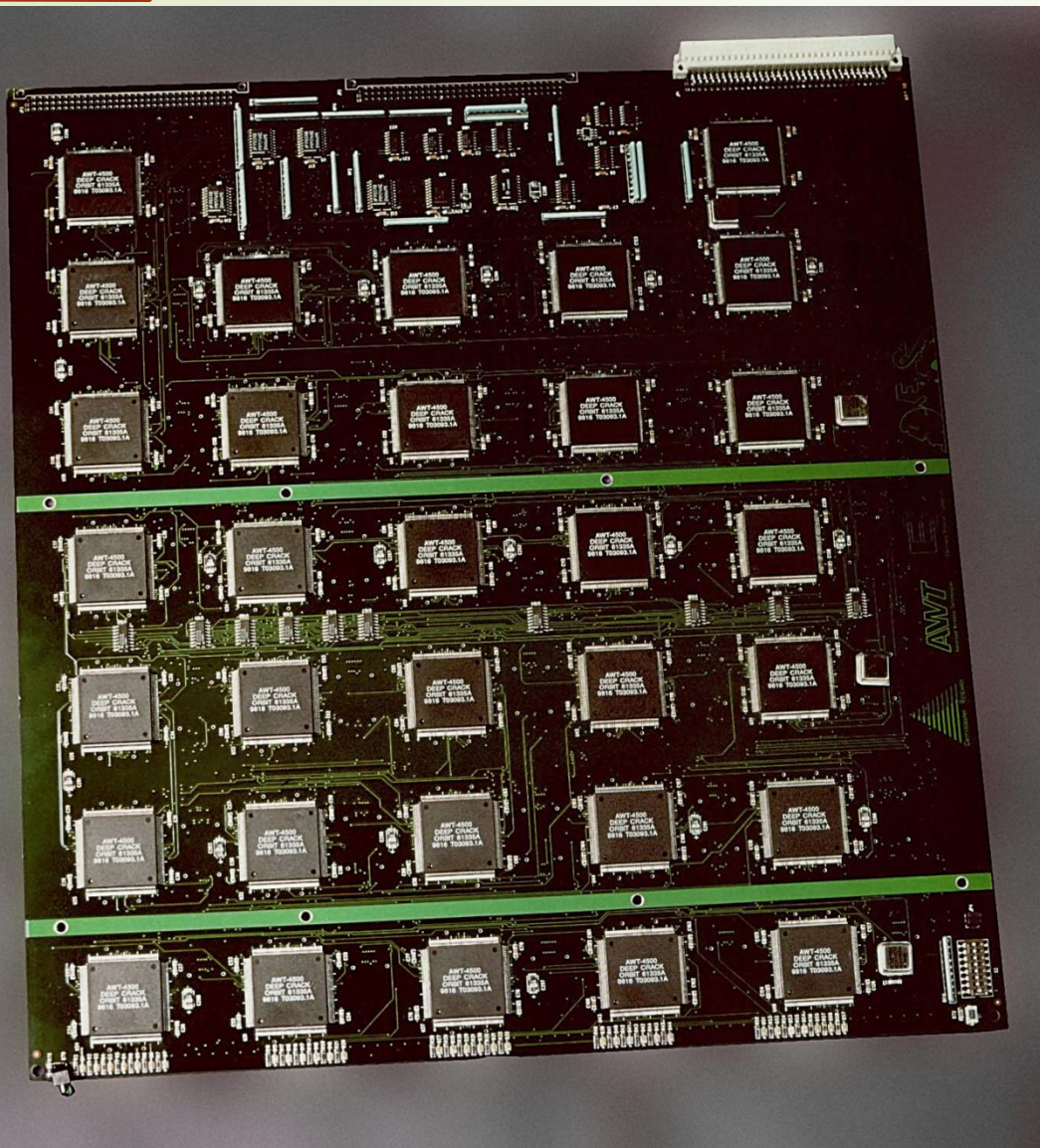
Brute Force Célgéppel

- DeepCrack elnevezésű gép – EFF alapítvány rendelte meg.
- $64 * 28 = 1792$ egyedi tervezésű FPGA-t tartalmazott.
- 28 alaplagra szerelve
- Egy DES kulcs megtörése 4-5 nap alatt, bonyolultságtól függően.
- 250 000 \$ volt a megépítés költsége, jelenlegi árfolyamon durván 55 millió Ft.
- Később, durván 50 000 \$-ból építhető volt hasonló gép.

Deep Crack számítógép



Deep Crack számítógép





Deep Crack számítógép

- Az FPGA áramkörökben összesen 50167db DES cella volt, vagyis egy áramkör 28 szálon futtatott DES titkosítást egyszerre.
- Egy szál csak annyit tudott, hogy próbálgatták a lehetséges kulcsokat egészen addig, amíg érdekes szöveget nem találtak a kimeneten.
- Érdekes szövegnek számított az alfanumerikus karakterek egymást követő felbukkanása.

Deep Crack számítógép

- A rendszer órajele csupán 40MHz volt, de a sok „mag” miatt egy másodperc alatt 107 520 000 000 kulcsot tudott kipróbálni.



- Ami valljuk





Rövid kis szösszenet arról, hogy mi is az Az FPGA?

- Field Programmable Gate Array.
- Olyan programozható logikai egység, amely logikai cellákból épül fel.
- Egy cella architektúrájától függően lehet 4 vagy 8 bites, vagy n bites, és bármilyen szinkron/aszinkron hálózat megvalósítható vele n biten.



Rövid kis szösszenet arról, hogy mi is az Az FPGA?

- A cellák kimeneti és bemeneti fizikai elhelyezkedése a chip-en belül szabadon programozható.
- Maga az áramkör belső felépítése is bármikor szabadon átprogramozható, mivel a konfigurációs adatokat a belső RAM memóriájába külső tárból tölti be az eszköz.



Rövid kis szösszenet arról, hogy mi is az Az FPGA?

- Masszívan párhuzamosítható számítások elvégzésére a leginkább alkalmas.
- Manapság kellően olcsó, így $n+1$ helyen alkalmazzák őket. Pl:
 - Bitcoin bányászat
 - Hardveres H.264 / VP8 kódolás
 - Egyedi CPU-k fejlesztése
 - stb...

Rövid kis szösszenet arról, hogy mi is az Az FPGA?


- Egyetlen egy baja az FPGA áramköröknek az, hogy bonyolult programozni őket, mivel a digitális logika szintjén kell gondolkodni.
- Vannak már kísérletek C/C++ fordítók átültetésére, de még komoly eredmények nem születtek.
- Bővebb olvasnivaló a téma iránt érdeklődőknek:
http://en.wikipedia.org/wiki/Field-programmable_gate_array

Törési versenyek

- RSA Inc. támogatta, célja az volt, hogy bebizonyítsák, hogy a DES elavult.
- Rekordok:
 - Pentium1 CPU + 16Mb ram -> 96 nap; 1997. január
 - Több géppel -> 41 nap; 1997. február
 - EFF DeepCrack -> 56 óra; 1998 július
 - Interneten összekapcsolt több géppel -> 20 óra 19 perc; 1999. január 19. (DeepCrack + 100000 PC)

Mégis hogy lehetséges?

- Összetett kriptóanalízissel sikerült optimalizálni a Brute Force eljárást
- Mindenki számára publikusan letölthető a Cracking DES c. könyvben
- Amazon.com-on nagyjából 4\$-ért megvehető.
- Számos publikus törőprogram. Pl:
<http://www.brianhpratt.net/cms/index.php?page=des-cracker>

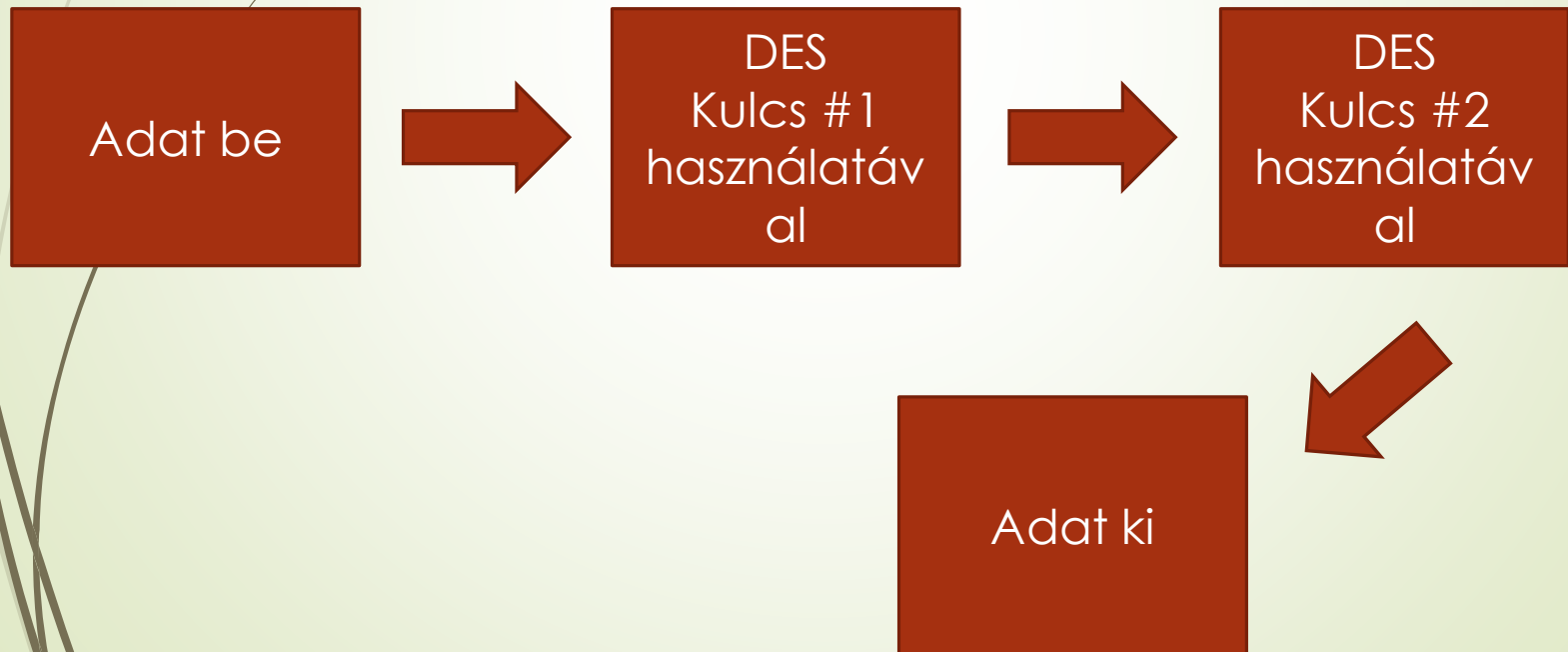


Des újra biztonságossá tétele

- Dupla DES (Double DES)
- Tripla DES (Triple DES)
- 3DES

Dupla DES

- DES titkosítással titkosított adat ismételt DES titkosítása más jelszóval.



Dupla DES problémája

- Elvileg $2 \cdot 56 \text{ bit} = 112$ bites kulcstérnek kellene keletkeznie
- Azonban matematikailag bebizonyították a „meet in the middle” támadással, hogy valójában ha 2x titkosítok valamit, az csak duplázza a lehetőségeket
 - vagyis egy bittel növeli az eredeti kulcsteret, ahelyett, hogy megduplázná azt
 - Dupla DES esetén ez 57 bit

Meet In the middle támadás

- Az m üzenet titkosítva van $K1$ kulccsal és a titkosítás eredménye ismét titkosításra kerül $K2$ kulccsal:
 - $M = C_{K2}(C_{K1}(m))$
- Ha a D_{K2} megfejtő függvényt az egyenlet mindkét oldalán alkalmazzuk, akkor az eredmény:
 - $D_{K2}(M) = C_{K1}(m)$
 - Magyarul: Az egyik kör megfejtő kulcsa a másik kör titkosító kulcsa

Meet In the middle támadás

- Ezt a matematikai összefüggést felhasználva az egyenlet jobb és bal oldalán kiszámoljuk az összes lehetőséget, amiből utána csak ki kell választani azt, ahol az egyenlőség teljesül.
- A támadás fő problémája, hogy a táblázatok tárolásához 2^{57} DES-szó, azaz 2^{60} bájt szükséges, így ebben a formában nem kivitelezhető.

Meet In the middle támadás

- Azonban a műveletsor optimalizálható úgy, hogy az algoritmus ideje duplázódik, de a szükséges tárterület feleződik.
- A Dupla DES ötletét azonban elvetették, helyette a gyakorlatban a 3x alkalmazott DES vált be, amit már 1979-ben javasolt az IBM

Tripla DES és 3DES

- Tripla DES: 3 körös DES, 3 különböző jelszóval
- 3DES
 - Nem azonos a Tripla DES algoritmussal
 - Két jelszót alkalmaznak, így a kulcstér 112 bit, ha 3 jelszót alkalmazunk, akkor 168 bit.
 - 168 bitet túlzásnak érezték, ezért maradt a két jelszó és végül ez lett a 3 DES

3DES



3DES Biztonsága

- 112 bites titkosítás
- Ezen elven Brute Force törés ellen tovább növelhető lenne a biztonsága extra körök beiktatásával, feltéve, ha a körök száma páratlan.
- Olcsó megoldás új algoritmus helyett.
- Tetszőleges algoritmusra alkalmazható az elve miatt



3DES Biztonsága

- Azonban elvénél fogva előbb-utóbb megtörhető ez is.
- Ideiglenes megoldásnak azonban jó volt.



A DES valódi utódja

- AES titkosítás
- Erről majd egy másik előadáson lesz részletesen szó.
- Sokkal bonyolultabb, mint a DES.
- Szintén szabványosított.



Köszönöm a figyelmet