Napadi na kriptosustave – uvod u kriptoanalizu

lakše prikupiti podatke

teže izvesti napad

- cilj: doznati tajni ključ *K*

Vrste napada prema onome što je napadaču dostupno:

- napad s odabranim čistim tekstom (*chosen-plaintext attack*) - napadač posjeduje neograničene količine parova (*M,C*), primjer s pametnim karticama
- napad s odabranim kriptiranim tekstom (*chosen-ciphertext attack*) - napadač posjeduje po svojoj volji odabrani *C* i pripadni *M* (također neograničene količine parova)
- napad s poznatim čistim tekstom (*known-plaintext attack*) - napadač posjeduje **neke** parove (*M,C*) - odgovaraju mu svi parovi, ali treba mu za napad određena količina parova
- napad s poznatim kriptiranim tekstom (*only-ciphertext attack*) - napadač posjeduje samo *C* a pokušava saznati *K* i *M* – napadaču je ovaj napad najteže uspješno provesti

1

Pretraživanje cijelog prostora rješenja

- napadač pokušava dekriptirati kriptirani tekst sa svim mogućim ključevima
- najjednostavnija i najsporija vrsta napada
- nije moguće spriječiti ovaj napad
- uspješnost svih napada na kriptosustave mjeri se usporedbom s pretraživanjem cijelog prostora
- napad koji ima veću složenost od složenosti pretraživanja cijelog prostora smatra se neuspješnim
- Pretpostavka: napadač ili već ima na raspolaganju čisti tekst ili pretpostavlja da čisti tekst ima neku standardnu strukturu koju je moguće prepoznati. Inače, u slučaju dekriptiranja poruke bez prepoznatljive strukture, napadač nema nikakve šanse da pretraživanjem cijelog prostora sazna koji je pravi ključ.

2

Pretraživanje pola prostora rješenja

- može se ostvariti kod mnogih kriptosustava za koje vrijedi simetrija:

  $C = DES(M, K)$ i $C' = DES(M', K')$

  (X' oznaka za bitovni komplement vrijednosti X)
- fiksno se postavi jedan bit ključa u '0'
- za svaki *K* se uspoređuje dobiveni kriptirani tekst *C''* sa *C* i *C'* i ukoliko vrijedi jednakost, radi se o *K* odnosno *K'*
- ušteda je vrlo blizu 50%
- vrijedi i za DES!
- zaštita od napada pretraživanjem pola prostora: koristiti kriptosustav za koji ne vrijedi navedeni tip simetrije ☺

3

Pomoć u napadu na kriptosustav

- uzeti u obzir frekvenciju slova (u promilima):

Tablica 1. Frekvencija slova u hrvatskom jeziku

A	I	O	E	N	S	R	J	T	U	D	K	V	L	M	P	C	Z	G	B	H	F
115	98	90	84	66	56	54	51	48	43	37	36	35	33	31	29	28	23	16	15	8	3

Tablica 1. Frekvencija slova u engleskom jeziku

E	T	A	O	I	N	S	H	R	D	L	C	U	M	W	F	G	Y	P	B	V	K	J	Q	X	Z
127	91	82	75	70	67	63	61	60	43	40	28	28	24	23	22	20	20	19	15	10	8	2	1	1	1

- frekvencija bigrama:
  HR: 2.8% je, 1.5% na, 1% an st an ni ko os ti ij no en
  EN: 3.2% th, 2.5% he, 1.2% an in er re on es ti at
- frekvencija trigrama:
  HR: 0.6% ije, 0.3-0.4% sta, ost, jed, koj, oje, jen
  EN: 3.5% the, 1.1% ing, 1% and, 0.7% ion, tio, ent, ...

4

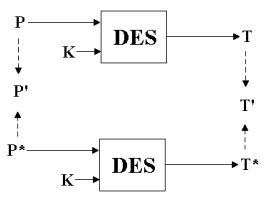
Napadi na DES

- bilo kakvim linearnim promjenama u postupku generiranja ključeva i u funkciji ***F***, DES ne postaje otporniji na napade
- promjena u nelinearnom dijelu algoritma (S tablice) **utječe** na ranjivost algoritma
- DES bitno oslabljuje:
  - promjena redosljeda S tablica
  - slučajno odabrane S tablice
  - umjesto XOR neka složenija funkcija
- pristup: analiza pojednostavljenog kriptosustava (s manje iteracija ili rundi, za primjerice DES sa samo tri runde).

5

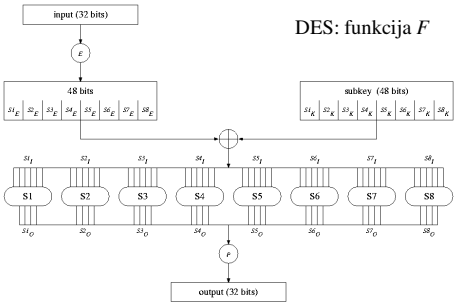
Diferencijalna kriptoanaliza

- Eli Biham, Adi Shamir, knjiga *Differential analysis of DES-like cryptosystems*, 1990.
- tehnika kojom se analizira učinak razlike između dva čista teksta na razliku između dva rezultirajuća kriptirana teksta
- razlike služe za određivanje vjerojatnosti mogućih ključeva
- napad s odabranim/poznatim čistim tekstom



6

♦razlika para podataka je neovisna o dodavanju ključa, linearno ovisi o ekspanziji E, permutaciji P i operaciji XOR



DES: funkcija *F*

$E(X) \oplus E(X^*) = E(X \oplus X^*)$ $\quad P(X) \oplus P(X^*) = P(X \oplus X^*)$

$(X \oplus K) \oplus (X^* \oplus K) = X \oplus X^*$

- S-tablice nisu linearne. Poznavanje razlike ulaznog para ne garantira poznavanje razlike izlaza iz S-tablica.
- Za bilo koju ulaznu razliku kod S-tablica postoji ograničen broj mogućih izlazhih razlika (ima i onih koje se sigurno neće pojaviti).
- Ulaz u S tablicu je veličine 6 bita, a izlaz 4 bita.
- Supstitucijska tablica *S1*:

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

- Postoji $2^6 = 64$ mogućih ulaznih razlika i $2^4 = 16$ izlaznih razlika.
- Svaka ulazna razlika može se ostvariti na 64 načina.
- Sve te mogućnosti mogu se pobrojati i zapisati u tablicu.

Input XOR	Output XOR															
	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	A_x	B_x	C_x	D_x	E_x	F_x
0_x	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1_x	0	0	0	6	0	2	4	4	0	10	12	4	10	6	2	4
2_x	0	0	0	8	0	4	4	4	0	6	8	6	12	6	4	2
3_x	14	4	2	2	10	6	4	2	6	4	4	0	2	2	2	0
4_x	0	0	0	6	0	10	10	6	0	4	6	4	2	8	6	2
5_x	4	8	6	2	2	4	4	2	0	4	4	0	12	2	4	6
6_x	0	4	2	4	8	2	6	2	8	4	4	2	4	2	0	12
7_x	2	4	10	4	0	4	8	4	2	4	8	2	2	2	4	4
8_x	0	0	0	12	0	8	8	4	0	6	2	8	8	2	2	4
9_x	10	2	4	0	2	4	6	0	2	2	8	0	10	0	2	12
A_x	0	8	6	2	2	8	6	0	6	4	6	0	4	0	2	10
B_x	2	4	0	10	2	2	4	0	2	6	2	6	6	4	2	12
C_x	0	0	0	8	0	6	6	0	0	6	6	4	6	6	14	2
D_x	6	6	4	8	4	8	2	6	0	6	4	6	0	2	0	2
E_x	0	4	8	8	6	6	4	0	6	6	4	0	0	4	0	8
F_x	2	0	2	4	4	6	4	2	4	8	2	2	2	6	8	8
⋮																
30_x	0	4	6	0	12	6	2	2	8	2	4	4	6	2	2	4
31_x	4	8	2	10	2	2	2	2	6	0	0	2	2	4	10	8
32_x	4	2	6	4	4	2	2	4	6	6	4	8	2	2	8	0
33_x	4	4	6	2	10	8	4	2	4	0	2	2	4	0	2	4
34_x	0	8	16	6	2	0	0	12	6	0	0	0	0	8	0	6
35_x	2	2	4	0	8	0	0	0	14	4	6	8	0	2	14	0
36_x	2	6	2	2	8	0	2	2	4	2	6	8	6	4	10	0
37_x	2	2	12	4	2	4	4	10	4	4	2	6	0	2	2	4
38_x	0	6	2	2	2	0	2	2	4	6	4	4	4	6	10	10
39_x	6	2	2	4	12	6	4	8	4	0	2	4	2	4	4	0
$3A_x$	6	4	6	4	6	8	0	6	2	2	6	2	2	6	4	0
$3B_x$	2	6	4	0	0	2	4	6	4	6	8	6	4	4	6	2
$3C_x$	0	10	4	0	12	0	4	2	6	0	4	12	4	4	2	0
$3D_x$	0	8	6	2	2	6	0	8	4	4	0	4	0	12	4	4
$3E_x$	4	8	2	2	2	4	4	14	4	2	0	2	0	8	4	4
$3F_x$	4	8	4	2	4	0	2	4	4	2	4	8	8	6	2	2

Broj mogućih izlaznih razlika za pojedinu ulaznu razliku tablice S1 (dio tablice)

Izlazna razlika Mogući ulazi za ulaznu razliku S1I' = 34_x

1_x 03_x, $0F_x$, $1E_x$, $1F_x$, $2A_x$, $2B_x$, 37_x, $3B_x$
2_x 04_x, 05_x, $0E_x$, 11_x, 12_x, 14_x, $1A_x$, $1B_x$, 20_x, 25_x, 26_x, $2E_x$, $2F_x$, 30_x, 31_x, $3A_x$
3_x 01_x, 02_x, 15_x, 21_x, 35_x, 36_x
4_x 13_x, 27_x
7_x 00_x, 08_x, $0D_x$, 17_x, 18_x, $1D_x$, 23_x, 29_x, $2C_x$, 34_x, 39_x, $3C_x$
8_x 09_x, $0C_x$, 19_x, $2D_x$, 38_x, $3D_x$
D_x 06_x, 10_x, 16_x, $1C_x$, 22_x, 24_x, 28_x, 32_x
F_x 07_x, $0A_x$, $0B_x$, 33_x, $3E_x$, $3F_x$

Ulaz u S-tablicu	Mogući ključevi za izlaznu razliku D_x i ulaze S1E = 1_x i S1E'=35_x (ulazna razlika 34_x)
06_x, 32_x 10_x, 24_x 16_x, 22_x $1C_x$, 28_x	07_x, 33_x 11_x, 25_x 17_x, 23_x $1D_x$, 29_x

primjer: $1_x \oplus 23_x = 22_x$ u S1 izlaz je 1_x ulazna razlika je $22_x \oplus 16_x = 34_x$
$35_x \oplus 23_x = 16_x$ u S1, izlaz je C_x izlazna razlika je $1_x \oplus C_x = D_x$

Učinkovitost napada diferencijalnom kriptoanalizom

1990. godine za probijanje DES-a od:

- šest rundi je bilo potrebno 0.3 sekunde i 240 tekstova
- osam rundi je bilo potrebno manje od 2 minute i 50,000 tekstova.

Broj rundi	4	6	8	9	10	11	12	13	14	15	16
Složenost	2^4	2^8	2^{16}	2^{26}	2^{35}	2^{36}	2^{43}	2^{44}	2^{51}	2^{52}	$\mathbf{2^{58}}$

- Eli Biham, Adi Shamir, *Differential cryptoanalysis of the full 16-round DES*, 1991, - opisuje napad diferencijalnom analizom izvediv na potpuni DES koji je brži od pretraživanja pola prostora.
- John Daemen, *Cipher and hash function design strategies based on linear and differential cryptoanalysis*, 1994, (jedan od autora Rijndaela) opisana je **Wide Trail Strategy** metoda koja pruža zaštitu i od diferencijalne i od linearne analize.

Linearna kriptoanaliza

- Cilj: pronaći linearnu aproksimaciju danog algoritma

$$\boldsymbol{P}[i_1, i_2, \ldots, i_a] \oplus \boldsymbol{C}[j_1, j_2, \ldots, j_b] = \boldsymbol{K}[k_1, k_2, \ldots, k_c]$$

Primjer

- neka je aproksimacija nekog algoritma dana izrazom:

  P [1, 4, 13] ⊕ C [1, 2, 3, 4, 6, 9, 11] = K [5, 6, 8]
- neka aproksimacija ima vjerojatnost p = 100%
- paritet 5., 6. i 8. bita ključa jednoznačno je određen paritetom pojednih bitova čistog i kriptiranog teksta
- duljina ključa se efektivno smanjila za 1 bit

Linearna kriptoanaliza

- aproksimacija nikada nema vjerojatnost ni blizu 100%
- taj nedostatak nadoknađuje se uzimanjem veće količine parova čisti/kriptirani tekst
- obično postoji više lineanih aproksimacija za neki algoritam.
- ukoliko izrazi nemaju ovisnosti o bitovima čistog teksta P, već samo o kriptiranom tekstu C i ključu K => napad s poznatim kriptiranim tekstom.
- učinkovitost algoritma raste s $|p - 0.5|$ i s rastom broja poznatih tekstova
- najbolja linearna aproksimacija DES-a reduciranog na 3 runde:

$$P_R[15] \oplus P_L[7, 18, 24, 29] \oplus C_R[15] \oplus C_L[7, 18, 24, 29] = K_1[22] \oplus K_3[22]$$

13

Učinkovitost napada linearnom kriptoanalizom

Računalo HP9750 (PA-RISC/66MHz), u programskom jeziku C:

- DES reduciran na 8 rundi je probijen s 221 tekstova u 40 sekundi;
- DES reduciran na 12 rundi je probijen s 233 tekstova za 50 sati;
- Potpuni DES je moguće probiti sa 247 tekstova brže od pretraživanja cijelog prostora;
- Ako se tekst sastoji samo od rečenica u engleskom jeziku (ASCII kod) DES reduciran na osam rundi moguće je probiti sa 229 kriptiranih tekstova;
- Ako se tekst sastoji od slučajno odabranih ASCII znakova DES reuduciran na 8 rundi moguće je probiti sa 237 kriptiranih tekstova (*only-ciphertext attack*);
- Potpuni DES je moguće probiti brže od pretraživanja cijelog prostora pomoću napada s poznatim kriptiranim tekstovima;
- Zaštita od napada linearne i diferencijalne kriptoanalize: *Wide Trail Strategy*

14

DES *Challenge I: 1997.*

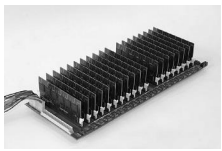
broj bitova ključa	vrijeme pronalaženja ključa
40	78 sekundi
48	5 sati
56	89 dana
64	41 godina
72	10.696 godina
80	2.738.199 godina
88	700.978.948 godina
96	179.450.610.898 godina
112	11.760.475.235.863.837 godina
128	770.734.505.057.572.442.069 godina

15

COPACOBANA

(*A Cost-Optimized Parallel Code Breaker*)

- razvila ga sveučilišta Ruhr iz Bochuma i Christian-Albrechts iz Kiela 2006. g.
- FPGA arhitektura, programibilan sustav (može se iskoristiti i u druge svrhe)
- 400 000 000 enkripcija u sekundi
- pretraga traje prosječno manje od 9 dana
- ≈ 9 kEUR (2006.g.)



16

Pregled asimetričnih kriptosustava

Za razmjenu simetričnih ključeva:
- Diffie-Hellman
- RPK
- KEA

Asimetrični algoritmi s parom ključeva:
- RSA
- Blum-Goldwaser
- **ECC**
- El Gamal
- LUC

17

Elektronički potpis

- 24. siječnja 2002. donešen je Zakon o elektroničkom potpisu

Elektronički potpis je skup podataka u elektroničkom obliku koji su pridruženi ili su logički povezani s drugim podacima u elektroničkom obliku i koji služe za identifikaciju potpisnika i vjerodostojnosti potpisanoga elektroničkog dokumenta.

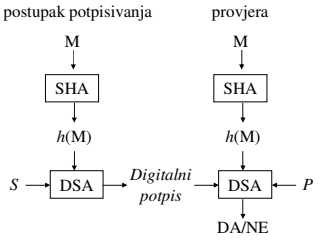
Napredan elektronički potpis je elektronički potpis koji pouzdano jamči identitet potpisnika i koji udovoljava sljedećim zahtjevima:

1. **elektronički potpis je povezan isključivo s potpisnikom,**
2. **nedvojbeno identificira potpisnika,**
3. **nastaje korištenjem sredstava kojima potpisnik može samostalno upravljati i koja su isključivo pod nadzorom potpisnika,**
4. **sadržava izravnu povezanost s podacima na koje se odnosi i to na način koji nedvojbeno omogućava uvid u bilo koju izmjenu izvornih podataka.**

18

DSS/DSA

- NIST je preporučio 1991. g. algoritam za digitalno potpisivanje elektroničkih dokumenata (*Digital Signature Algorithm – DSA*) da postane sastavni dio norme (*Digital Signature Standard - DSS*)



DSA

- je zapravo *ElGamalov* digitalni potpis

Postupak generiranja ključeva

- L – broj bitova ključa
- q – prim broj jednake duljine kao i H
- p – L-bitni prim broj takav da je (p-1) višekratnik od q
- $g = h^{(p-1)/q} \bmod p$ (g>1), gdje je $h \in (1, p\text{-}1)$
- izabrati $x \in (0, q)$ i izračunati
- $y = g^x \bmod p$

privatni ključ je $S = x$, a javni $P = (p, q, g, y)$

DSA

Postupak potpisivanja

- za svaku poruku m generiraj slučajni broj $k \in (0, q)$
- $r = (g^k \bmod p) \bmod q$, $r \neq 0$
- $k^{-1} \in (0,q)$ takav da vrijedi $(k^{-1} k) \bmod q = 1$
- $s = (k^{-1}(H(m) + xr)) \bmod q$

digitalni potpis = (r, s)

Provjera potpisa

- $w = s^{-1} \bmod q$
- $v = ((g^{(H(m)*w) \bmod q} * y^{(r*w) \bmod q}) \bmod p) \bmod q$
- potpis je ispravan ako je $v = r$

Važna svojstva funkcija za izračunavanje sažetka poruke

Otportnost na izračunavanje originala (*preimage resistance*)

- $H=h(M) \quad \Rightarrow M=h^{-1}(H)$ ne postoji

Otpornost na izračunavanje poruke koja daje isti sažetak (*2-nd preimage resistance*)

- za poznati M i $H=h(M)$ je nemoguće pronaći M' koji daje isti H

Otpornost na kolizije (*collision resistance*)

- nemoguće je pronaći bilo koje dvije poruke M_1 i M_2 za koje se dobiva isti sažetak $h(M_1)=h(M_2)$

Napadi hash na funkcije

- 1998. Dobbertin pronalazi kolizije za MD4 unutar 1 sekunde na PC računalu
- 17.8.2004. kineski i francuski znanstvenici su objavili članak: "Kolizija za hash funkcije: MD4, MD5, Haval-128 i RIPEMD"
- superračunalo IBM P960 za 1 sat pronalazi koliziju za MD5 (kolizije za MD4 se mogu pronaći i bez računala tvrde isti autori)
- superračunalo s 256 Itanium procesora pronalazi za 13 dana koliziju za SHA-0
- 13.2.2005. kineski znanstvenici: "Collision Search Attacks on SHA-1"

Napadi hash na funkcije

- kolizije su bezopasne sve dok izgledaju kao slučajan niz
- ipak, gubi se povjerenje u certifikate i protokole koji koriste sažetak slučajnog simetričnog ključa
- problem nevidljivih podataka u Word dokumentu ili slučajnih nizova u slikama!
- primjer: www.win.tue.nl/hashclash/Nostradamus

- u studenom 2007.g. NIST je raspisao natječaj za SHA-3
- algoritmi su se mogli predlagati do listopada 2008.g.
- pristiglo je 64, a za prvi krug je odabran 51 algoritam
- postupak odabira je još uvije u tijeku:
  - 1.krug veljača, 2009.
  - travanj 2009: preostao 41 kandidat
  - 2.krug 2010.
  - konačan odabir 2012.

Birthday attack

- vjerojatnost da dvije poruke iz skupa od $k=1.2(2^n)^{1/2}=$ $=1.2\cdot 2^{n/2}$ poruka daju isti sažetak je veća od 50%, gdje je n duljina sažetka
- analogno: vjerojatnost da dvije osobe u dvorani u kojoj je ukupno $k=1.2\cdot 365^{1/2}=23$ ljudi imaju isti dan rođendan je veća od 50%

M_1: "UGOVOR: Za 657200 kn je Ana Twofish kupila stan od Branka Horvata."

M_2: "UGOVOR: Za 176450 kn je Ana Twofish kupila stan od Branka Horvata."

25

```
M1.txt
00000000 d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
00000010 2f ca b5 87 12 46 7e ab 40 04 58 3e b8 fb 7f 89
00000020 55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 71 41 5a
00000030 08 51 25 e8 f7 cd c9 9f d9 1d bd f2 80 37 3c 5b
00000040 96 0b 1d d1 dc 41 7b 9c e4 d8 97 f4 5a 65 55 d5
00000050 35 73 9a c7 f0 eb fd 0c 30 29 f1 66 d1 09 b1 8f
00000060 75 27 7f 79 30 d5 5c eb 22 e8 ad ba 79 cc 15 5c
00000070 ed 74 cb dd 5f c5 d3 6d b1 9b 0a d8 35 cc a7 e3
M2.txt
00000000 d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
00000010 2f ca b5 07 12 46 7e ab 40 04 58 3e b8 fb 7f 89
00000020 55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 f1 41 5a
00000030 08 51 25 e8 f7 cd c9 9f d9 1d bd 72 80 37 3c 5b
00000040 96 0b 1d d1 dc 41 7b 9c e4 d8 97 f4 5a 65 55 d5
00000050 35 73 9a 47 f0 eb fd 0c 30 29 f1 66 d1 09 b1 8f
00000060 75 27 7f 79 30 d5 5c eb 22 e8 ad ba 79 4c 15 5c
00000070 ed 74 cb dd 5f c5 d3 6d b1 9b 0a 58 35 cc a7 e3

$ md5sum M1.txt
MD5 Sum = a4c0d35c95a63a805915367dcfe6b751
$ md5sum M2.txt
MD5 Sum = a4c0d35c95a63a805915367dcfe6b751
```

26

```
M1.txt
00000000 d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
00000010 2f ca b5 87 12 46 7e ab 40 04 58 3e b8 fb 7f 89
00000020 55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 71 41 5a
00000030 08 51 25 e8 f7 cd c9 9f d9 1d bd f2 80 37 3c 5b
00000040 96 0b 1d d1 dc 41 7b 9c e4 d8 97 f4 5a 65 55 d5
00000050 35 73 9a c7 f0 eb fd 0c 30 29 f1 66 d1 09 b1 8f
00000060 75 27 7f 79 30 d5 5c eb 22 e8 ad ba 79 cc 15 5c
00000070 ed 74 cb dd 5f c5 d3 6d b1 9b 0a d8 35 cc a7 e3
M2.txt
00000000 d1 31 dd 02 c5 e6 ee c4 69 3d 9a 06 98 af f9 5c
00000010 2f ca b5 07 12 46 7e ab 40 04 58 3e b8 fb 7f 89
00000020 55 ad 34 06 09 f4 b3 02 83 e4 88 83 25 f1 41 5a
00000030 08 51 25 e8 f7 cd c9 9f d9 1d bd 72 80 37 3c 5b
00000040 96 0b 1d d1 dc 41 7b 9c e4 d8 97 f4 5a 65 55 d5
00000050 35 73 9a 47 f0 eb fd 0c 30 29 f1 66 d1 09 b1 8f
00000060 75 27 7f 79 30 d5 5c eb 22 e8 ad ba 79 4c 15 5c
00000070 ed 74 cb dd 5f c5 d3 6d b1 9b 0a 58 35 cc a7 e3

$ md5sum M1.txt
MD5 Sum = a4c0d35c95a63a805915367dcfe6b751
$ md5sum M2.txt
MD5 Sum = a4c0d35c95a63a805915367dcfe6b751
```

27

Pregled funkcija za izračunavanje sažetka

HAVAL	128, 160, 192, 224, 256	MD4, MD5	128
Panama	256	RIPEMD	160
Sapphire II	128, 136, 144,...,256	SNEFRU	128, 256
SHA-1	160, 256, 384 i 512	Tiger	192

Preporuke

	Minimalno	Preporuka	US1	US2
Simetrični (ne DES!)	96	256	64	56 (40)
Asimetrični	1024	4096	1024	512
EC	192	256	160	112
HASH	160	SHA-256	-	-
Klasa certifikata	2	3	-	-

28

D. Chaum predlaže:

- primijeniti tehniku "podijeli i odaberi" (*cut and choose*)
- kupac priredi n različitih novčanica s istim iznosom
- svih n novčanica podnese banci na potpis
- banka nasumice odabere n-1 novčanica i za njih od kupca zahtijeva da ih "otkrije"
- ako pregled pokaže da je svih n-1 novčanica ispravno, banka tada "na slijepo" potpisuje onu jednu preostalu novčanicu
- Nedostatak: nezgrapnost "podijeli i odaberi" procedure, jer se za veću razinu sigurnosti protiv prevare kupca mora koristiti veliki broj n.

29

Svojstva Kerberos protokola

- zaštita od napada ponovnim korištenjem starih paketa (*replay attack*): vremenske oznake
- funkcija f
  - jednostavna
  - jednosmjerna - nema f^{-1}
  - može biti i funkcija sažimanja
- moguće je ostvariti autorizaciju u procesu dodjele dozvole (autorizacija nije ostvarena!)
- štiti samo lozinke i povjerljive informacije, a ne sve podatke koji prolaze kroz mrežu
- manje važni podaci (imena, adrese, telefoni...) nisu pohranjeni u Kerberos poslužitelju, već u posebnom, tzv. *Hesiod* poslužitelju

30

PGP (*Pretty Good Privacy*)

- hibridni kriptosustav čija je osnovna namjena zaštita elektroničke pošte
- autor: Philip R. Zimmermann
- *hibridni* – jer omogućuje kriptiranje simetričnim i asimetričnim kriptosustavom, arhiviranje podataka i digitalno potpisivanje
- prije kriptiranja, podaci se komprimiraju
- simetrični kriptografski algoritam IDEA (*International Data Encryption Algorithm*)
- Triple-DES, CAST-128, Blowfish, AES itd.
- asimetrični algoritam: RSA, Diffie-Hellman

31

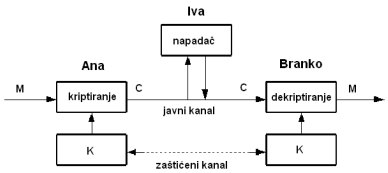
Kvantna kriptografija

- danas se računalna sigurnost zasniva na nedokazanoj činjenici da ne postoji efikasan algoritam za faktorizaciju velikih brojeva te za izračun diskretnog algoritma
- Shor, 1994.: kvantni algoritam (može se ostvariti na kvantnom računalu) za brzu faktorizaciju brojeva
- rješenje: **QKD protocol**

32

Protokol BB84

- prvi QKD protokol
- predložili ga Charles H.Bennett (IBM) i Gilles Brassard (Montreal)
- dva kanala: javni i kvantni (optički kabel)

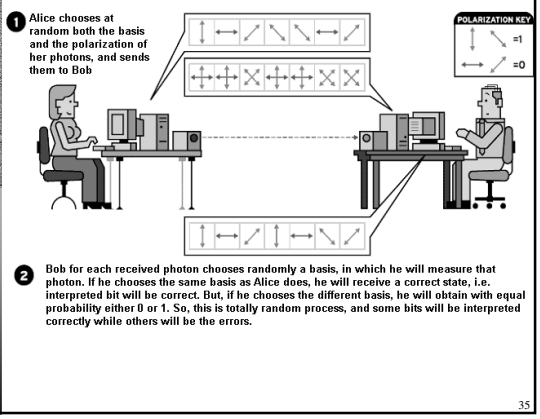


33

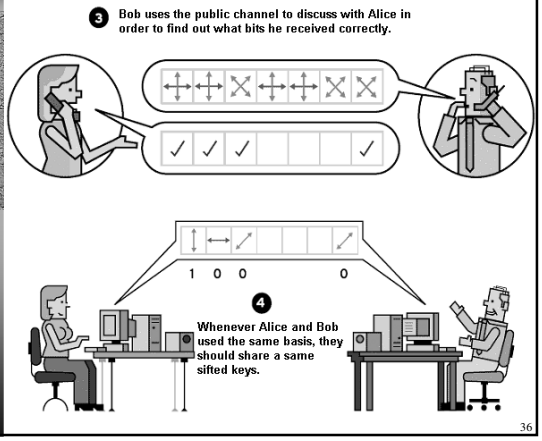
Protokol BB84

- puls polariziranog svjetla s *jednim* fotonom
- 4 moguće polarizacije fotona
  - **baza ⊕**: foton je ili vertikalno (90°) ili horizontalno (0°) polariziran
  - **baza ⊗**: foton je dijagonalno polariziran (45° ili 135°)
- sigurnost protokola temelji se na
  - nemogućnosti kloniranja fotona
  - Heisenbergovom principu neodređenosti

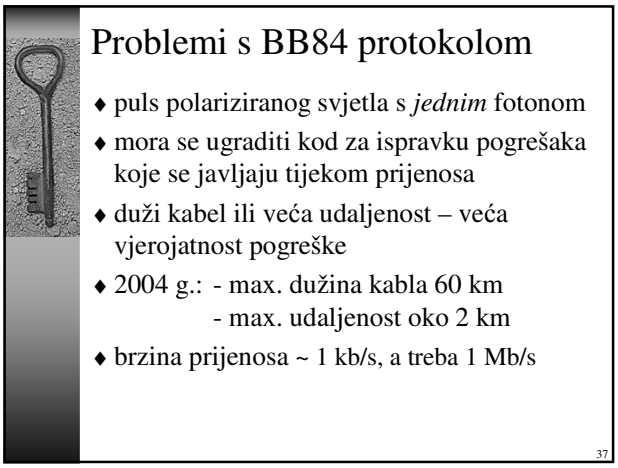
34



35



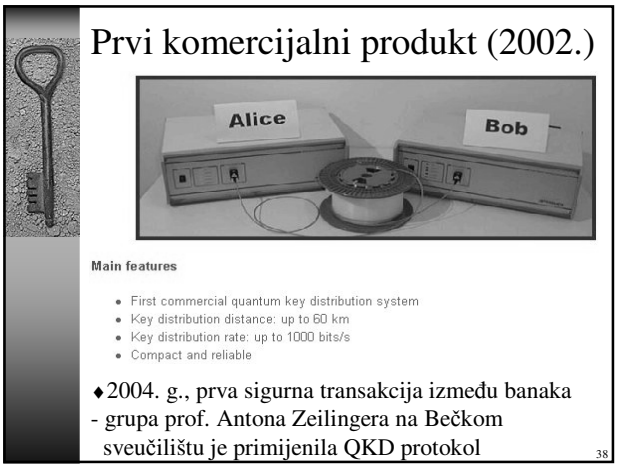
36



Problemi s BB84 protokolom

- puls polariziranog svjetla s *jednim* fotonom
- mora se ugraditi kod za ispravku pogrešaka koje se javljaju tijekom prijenosa
- duži kabel ili veća udaljenost – veća vjerojatnost pogreške
- 2004 g.: - max. dužina kabla 60 km
  - max. udaljenost oko 2 km
- brzina prijenosa ~ 1 kb/s, a treba 1 Mb/s

37



Prvi komercijalni produkt (2002.)

Main features

- First commercial quantum key distribution system
- Key distribution distance: up to 60 km
- Key distribution rate: up to 1000 bits/s
- Compact and reliable

- 2004. g., prva sigurna transakcija između banaka - grupa prof. Antona Zeilingera na Bečkom sveučilištu je primijenila QKD protokol

38