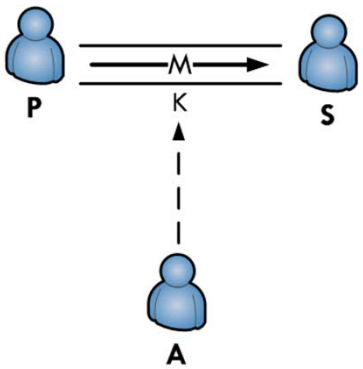
KRIPTOGRAFIJA

Varna izmenjava podatkov

- **P** - pošiljatelj (Alice)
- **S** - prejemnik (Bob)
- **M** - sporočilo (message)
- **K** - prenosni medij (channel)
- **A** – vsiljivec (adversary)
- manipulacija s sporočilom:
  - blokiranje
  - prestrezanje
  - spreminjanje
  - ponarejanje



Varna izmenjava podatkov

- skrivni algoritmi
  - problem dokazovanja varnosti
  - problem novega algoritma za vsakega naslovnika
- ključi
  - enak algoritem, lahko javen
  - različni ključi

Kaj je kriptografija?

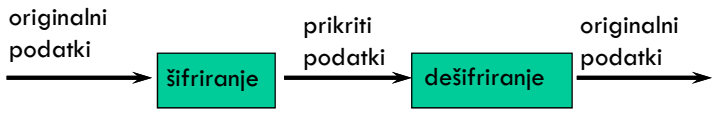
- prvoten pomen termina je prikrivanje podatkov (encryption)
  - sporočila “zamaskiramo” tako, da ni možno ugotoviti njihove vsebine
- potreba izhaja iz vojskovanja
- cilje je ohraniti **zaupnost** podatkov
- danes kriptografije zajema veliko več kot samo prikrivanje podatkov

Kaj zajema kriptografija?

- šifrirni algoritmi - šifre
  - simetrični (symmetric encryption)
  - asimetrični (asymmetric encryption) – tudi kriptografija javnega ključa
- zgoščevalne funkcije
  - zgoščevalne funkcije (modification detection codes - MDC)
  - funkcije za overjanje sporočil (message authentication codes - MAC)
- varnostni protokoli:
  - overitveni protokoli (authentication protocols)
  - protokoli za izmenjavo ključev (key agreement protocols)
- algoritmi za digitalno podpisovanje (signature schemes)

Šifrirni algoritmi

Šifrirni algoritmi: delovanje



Šifrirni algoritmi

- šifriranje je proces kodiranja sporočila tako, da njegov pomen ni očiten
  - dešifriranje je obratni proces
- sistem za šifriranje in dešifriranje se imenuje kriptosistem
- Terminologija:
  - izvorno sporočilo (plaintext), *P*
  - Kriptogram (ciphertext), *C*
- šifrirni algoritmi - šifre
  - brez ključa
  - simetrični
  - asimetrični

Šifre

- doseganje zaupnosti podatkov
- včasih
  - šifriranje besedila
  - preprosti algoritmi
- danes
  - šifriranje podatkov v digitalni obliki (biti)
  - kompleksni algoritmi
- uporaba algoritmov (postopkov)
  - struktura algoritmov je javno znana
  - varnost temelji na ključu

Zgodovine: Nekaj dejstev

- Hebrejci so šifrirali nekatere besede v svojih skriptah
- pred 2000 leti je Julij Cezar uporabil preprosto substitucijo, danes poimenovano Cezarjeva šifra
- Roger Baconje opisal več metod prikrivanja v 13. stoletju
- Leon Alberti je izdelal šifrirno kolo in opisal princip frekvenčne analize v letih okoli 1460
- Blaise de Vigenére je izdal knjigo o kriptografiji v letu 1585 in opisal polialfabetično substitucijo
- uporaba se je povečevala skozi stoletja, posebej v diplomaciji in v vojnah

Zgodovina: "klasična" doba

- 1900 pr. n. št.
  - začetki uporabe šifrirnih algoritmov na nagrobnikih
  - prva znana uporaba kriptografije
- 475 pr. n. št.
  - Šparta razvije prvo uporabo kriptografije in prvo kriptografsko napravo
- 60 pr. n. št.
  - Julij Cezar postane prvi znani uporabnik substitucijske šifre v vojne namene – Cezarjeva šifra

Zgodovina: moderna doba

- 1971 – danes
  - Moderna kriptografija
  - Moderni algoritmi AES, 3DES,…
- 1976
  - Diffie-Hellman – kriptografija javnega ključa, protokol za vzpostavitev ključa
- 1977
  - RSA – prva asimetrična šifra
  - DES je postal standard
- 1997
  - DES algoritem je razbit z napadom grobe sile
- 2000
  - sprejet Advanced Encryption Standard (AES) - algoritem Rijndael (Vincent Rijmen, Joan Deamen)

Idealen algoritem

- ključ velikosti podatkov
- je absolutno varen algoritem
- resnično naključen ključ
- uporaba operacije XOR, da ključ kombiniramo s podatki
- XOR – pomemben v vseh sodobnih kriptografskih algoritmih
- slabosti:
  - podvajanje prenesene količine podatkov –za vsak poslan bit sporočila je treba prenesti en bit ključa
  - če je ključ poslan po nezavarovanem kanalu ga lahko kdo prestreže
  - težko je izdelati veliko količino (resnično) naključnih števil

Cezarjeva šifra

- $c_i = E(p_i) = p_i + n$
- preprost algoritem, ki si ga je mogoče hitro zapomniti
- uporablja permutacije črk abecede
- ključ je število zamikov
  (primer zamika za 4 črke v levo) :
  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
  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

Monoabecedni algoritmi

- Substitucijski algoritmi
  - Menjava črke s črko
- Dovzetni na napade s frekvenčno analizo

Poliabecedni algoritmi

- So se razvili kot odgovor na slabosti osnovnih algoritmov že pred nekaj stoletji
- Vigenerov algoritem
- Vernamov algoritem

Vigenerov algoritem

- Blais de Vigenere v 16 stoletju
- Kadar eno črko vedno zamenjamo z isto črko govorimo o **monoabecednih** algoritmih

  SPOROČILO
  KLJUČKLJU
- Kriptirano sporočilo je vsota črk sporočila in ključa

- Rezultat je sestavljen iz toliko Cezarjevih algoritmov, kolikor dolg je ključ

Vigenerov algoritem

- Možen napad z grobo silo
  - Poiskati je treba ključ za več Cezarjevih algoritmov
  - V pomoč je lahko dolžina ključa
- Iskanje ponavljajočih blokov in iskanje skupnega delitelja lahko razkrije dolžino ključa
- Brez težav pri poznanem ali izbranem sporočilu

Ekskluzivni ALI

- Enkratna preglednica se v dobi računalnikov uporablja za enkripcijo bitov
- XOR – rezultat funkcije je 1, kadar sta oba bita enaka in 0 kadar sta različna
- Ekskluzivni ali ima pomembno vlogo v vseh modernih algoritmih

Enigma in drugi algoritmi z rotirajočim bobnom

- Od prve svetovne vojne naprej so se uporabljali Vigenerovi algoritmi za enkripcijo radijskih sporočil
- Stroj za enkripcijo z rotirajočim bobnom poznamo iz leta 1918 – izumili so ga neodvisno 4 izumitelji v istem času
- V osnovi so to poliabecedni substitucijski algoritmi

Frekvenčna analiza

Frekvenčna analiza

- Metoda, ki omogoča razkritje besedila pri t.i. substitucijskih algoritmih, ki izvajajo enočrkovno zamenjavo (npr. Cezarjeva šifra)
- Omogoča razkritje besedila brez poznavanja ključa
- Metoda izkorišča neenakomerno frekvenco ponavljanja črk v besedilu, saj se določene črke pojavljajo pogosteje kot druge - npr. “a” ali “e” se v slovenščini pojavljata bolj pogosto kot “f”

Postopek frekvenčne analize

- Za frekvenčno analizo potrebujemo frekvenco črk določenega jezika, ki jo nato primerjamo s šifriranim besedilom
- V ta namen uporabimo t.i. referenčne datoteke – datoteka z besedilom izbranega jezika, ki mora biti čim daljša, da daje boljši rezultat
- Izvedemo frekvenčno analizo referenčnega besedila in dobimo frekvenco črk danega jezika oz. referenčnega besedila

Postopek frekvenčne analize

- Nato izvedemo frekvenčno analizo šifriranega besedila in dobimo njeno frekvenco
- Zamenjamo znake glede na rezultate frekvenčne analize – *pozor gledamo pozicijo črk in ne dejansko število oz. odstotek posameznih črk (glej prosojnico s primerom)*

Primer

Referenčna datoteka:

a 10%
m 8%
c 6%
…
i 4%
s 4%

Šifrirana datoteka:

t 9,5%
x 8%
h 6,5%
…
f 4%
l 4%

Primer

- Zamenjave črk, ki jih izvedemo na podlagi frekvenčnih analiz:
  "t" z "a"
  "x" z "m"
  "h" s "c"
  …
- V primeru, da imajo črke enako frekvenco pojava (v tem primer "i" in "s") je vrstni red zamenjave poljuben:
  - "f" z "i" in "l" s "s" ali "f" s "s" in "l" z "i"

Primer šifriranega besedila

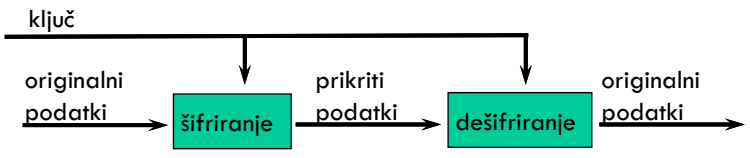
AXGN UGXYGH
MGNEČ TDNOE
ČFDJDXANTIG NDXEWG
A
X WEUR 1660 ČGJNME JNEXE PETEOG MRWAMG,
PHFDWEWD ME ZD RWAOGQ WMRBWMGNTIAQ XTE
ZDWND ZDJELEWTIEFG ZWEPTUXG. ZD
ZHDJGMGWNAOGQ TD UA XAUEČA NGIRZDXGWA, IGH TD
ZDUHEBDXGWA ČGTE AN TXDMD JHRLAND. PWGMCA
TANDXA - ZG TD TE X NDXAQ DZHGXGQ FNEUWA ZD
UWGIR, WGČAWA ČG PECYGNTIAPA JEIWEUA,
ZHEUEZGWA TE Č JAMGCUXDP AN PETGHMA UEH
BGNIEUDXGWA TEJGM UR, TEJGM UGP.

Primer dešifriranega besedila

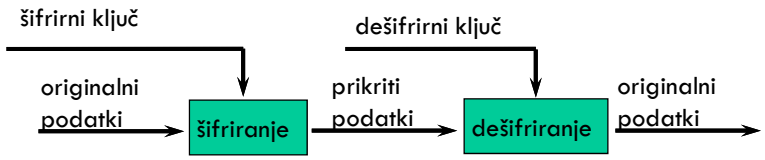
IVAN TAVČAR
JANEZ SONCE
ZGODOVINSKA NOVELA
I
V LETU 1660 ZADNJE DNEVE MESECA JULIJA, MRGOLELO
JE PO ULICAH LJUBLJANSKIH VSE POLNO
PODEŽELSKEGA PLEMSTVA. PO PRODAJALNICAH SO TI
VITEZI NAKUPOVALI, KAR SO POTREBOVALI ZASE IN
SVOJO DRUŽINO. MLAJŠI - SINOVI - PA SO SE V NOVIH
OPRAVAH GNETLI PO TLAKU, LAZILI ZA MEŠČANSKIMI
DEKLETI, PRETEPALI SE Z DIJAŠTVOM IN MESARJI TER
BANKETOVALI SEDAJ TU, SEDAJ TAM.

Sodobne šifre

Simetrično šifriranje



Šifriranje javnega ključa (asimetrično)



Šifriranje javnega ključa

- rešuje težavo upravljanja s ključi
- vpeljuje koncept infrastrukture javnih ključev (Public Key Infrastructure – PKI)
- uporaba para ključev
  - javni ključ (šifriranje)
  - zasebni ključ (dešifriranje)
- uporaba v namene šifriranja, tudi digitalnega podpisovanja

Šifriranje javnega ključa

- prednosti:
  - preprosto upravljanje s ključi
  - uporaba tako za šifriranje kot digitalno podpisovanje
- slabosti:
  - počasnosti (1000 počasnejši od simetričnih)
  - temeljijo na kompleksnih matematičnih osnovah

Šifriranje javnega ključa

- predstavniki algoritmov: RSA, ElGamal,…

Dolžina ključa - simetrični algoritem	Dolžina ključa - asimetrični algoritem
112 bitov	2048 bitov
128 bitov	3072 bitov
192 bitov	7680 bitov
256 bitov	15360 bitov

- primeri uporabe:
  - GPG, implementacija OpenPGP-ja
  - Orodja za šifriranje (npr. TrueCrypt)
  - SSL (Secure Socket Layer) / TLS
  - SSH

Simetrično šifriranje

- dolžina ključa določa težavnost razkritja prikritih podatkov s pomočjo tehnike grobe sile (brute-force)
- bločni šifrirni algoritmi
  - šifriranje podatkov blok po blok
  - pogosto uporabljeni
- tokovni šifrirni algoritmi
  - šifriranje podatkov bit po bit
  - redkeje uporabljeni (npr. rc4 pri wlan)

Bločni šifrirni algoritmi

- blokovne šifre
- podatki se delijo na bloke
  - blok – niz podatkov fiksne dolžine
- procesirajo podatke blok za blokom
- prednosti:
  - hitri
  - temeljijo na preprostih konstruktih
- slabosti:
  - problem upravljanja s ključi

Bločni šifrirni algoritmi

- predstavniki algoritmov:
  DES, IDEA, SAFER, Blowfish, RC5, CAST-128, MARS, RC6, SERPENT, TWOFISH, **AES**, CAST-256, DEAL

Dolžina ključa	Čas potrebe za razbitje s pomočjo grobe sile
56 bitov	1s
64 bitov	4 min
112 bitov	10^9 let
128 bitov	10^{14} let
192 bitov	10^{33} let
256 bitov	10^{52} let

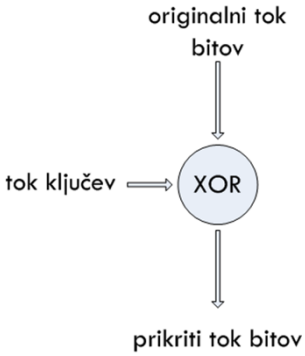
- **Starost vesolja je 10^{10} let!!**

Tokovni šifrirni algoritmi

- tokovne šifre
- originalni podatki so kombinirani s pseudo-naključni zaporedjem bitov, t.i. tok ključev (keystream)
- tipično se uporablja operacija eksluzivni-ali (xor)
- šifriranje poteka bit za bitom oz. bajt za bajtom

Tokovni šifrirni algoritmi

- predstavniki algoritmov: A5/1A5/1, A5/2, FISH, PANAMA, RC4, Salsa20, SEAL, SNOW, VEST, WAKE,...



Simetrična vs. asimetrična kriptografija

- simetrični algoritmi:
  - preprostejši
  - hitrejši
  - lažje za razvit
  - manjša računska zahtevnost
- asimetrični algoritmi:
  - temeljijo na kompleksnih matematičnih operacijah in problemih
  - težje razviti
  - počasnejši
  - večja računska zahtevnost

Sodobne simetrične šifre

Sodobni simetrični algoritmi

- DES (Data Encryiption Standard)
- Triple-DES
- IDEA (International Data Encryption Algorithm)
- RC2 (Rivest Chiper 2)
- CAST (Carlisle Adams, Stafford Tavares)

- Skipjack
- MISTY (Mitsubishi)
- AES (Advanced Encryption Standard) – Rijndael

DES

- Razvil IBM v 1970-tih letih
- 1977 sprejet kot standard za enkripcijo – US National Institute of Standards and Technology (NIST)
- NSA vključena v razvoj
- Ključni kriteriji za izdelavo algoritma so ostali skrivni
- Smiselna uporaba samo z računalnikom, bolj zapleten kot Vigenere ali rotor

DES

- Je kombinacija enkratne preglednice, permitacijskih in substitucijskih algoritmov
- Izvaja se nad biti in ne nad znaki
- Deluje nad bloki po 64 bitov
- Kriptirano sporočilo je enako dolgo kot izvorno sporočilo
- Uporablja osnovne operacije: ekskluzivni ali, permutacijo, substitucijo

DES ključ

- Ima 64 bitov
- 56 bitni ključ + 8 kontrolnih bitov
- Generiranje ključev
  - Kontrola ključa s pomočjo kontrolnih bitov
  - Generiranje 16 x 48-bitnih ključev
  - Ključi so torej med seboj odvisni, kar pa se ni izkazalo kot slabost DES

Dekripcija DES

- Se izvede z enakim algoritmom kot enkripcija
- 16 ključev se generira v obratnem vrstnem redu

Varnost DES

- Je prvi zgled, da je javno objavljen algoritem varnejši od skrivnih
- Ključ je kratek, IBM je želel 128-bitni ključ, kar je preprečila NSA
- Po 1990 so gotovo že obstajali računalniki v lasti tajnih služb, ki zlomijo DES
- Računalnik za nekaj milijonov EUR lahko zlomi DES v nekaj sekundah

DES danes

- Ni primeren za sporočila z visoko stopnjo zaupnosti
- Še vedno primeren za on-line bančništvo in on-line plačevanje

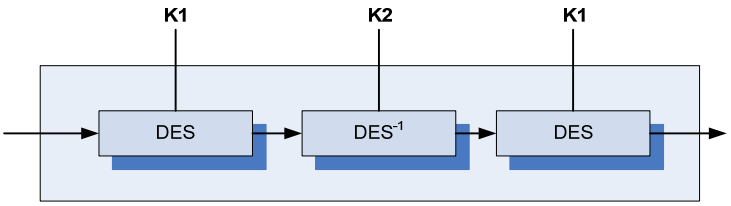
Slabosti DES

- Izdelan za uporabo v strojni izvedbi – počasen v programskih izvedbah
- Optimiziran za strojno opremo 1970-tih let
- Začetna in končna permutacija v programski implementaciji samo upočasnita algoritem, ni pravega učinka
- Ima fiksno določeno velikost bloka in ključa
- Kriteriji načrtovanja še vedno niso znani

DES je neprimeren za uporabo v vladnih aplikacijah ZDA

- Julij 2004 – predlog, da se umakne certifikat DES za uporabo v vladnih aplikacijah ZDA
- Po mnenju NIST (National Institute of Standards and Technology)
- Zaradi napredka masovnega paralelnega procesiranja

Trojni DES



- Napad z grobo silo traja 2^{56} krat dlje od DES
- Z dovolj pomnilnika traja napad teoretično samo 2x dlje od DES
- Je 3x počasnejši od DES
- Če je K1=K2 je lahko tudi navadni DES

IDEA

- IDEA – International Data Encryption Standard
- 1990, Zurich – Xueija Ali in James Massey
- Uporabljen v PGP (+ CAST + Triple-DES)

- Do sedaj še ni bila izpostavljena nobena slabost, podobno kot za DES
- Zgleduje se po DES
- Dolžina ključa je 128 bitov
- **Patentiran!**

AES

- Advanced Encryption Standard
- izbran na natečaju – zmaga algoritma Rijndael
- avtorja Vincent Rijmen in Joan Deamne s KU Lueven
- karakteristike:
  - dolžina ključev 128, 192, 256 bitov
  - dolžina bloka 128 bitov
  - Preprosta implementacija (strojna in programska)
  - ni patentiran
  - hiter
  - male zahteve glede pomnilniškega prostora

AES: Varnost

- napadi:
  - napad s stranskim kanalom (side-channel attack) – generični napad na vse šifrirni algoritme
  - NIST je opredelila, da je AES dovolj varen za uporaba v ameriških vladnih organizacijah
  - leta 2003 je vlada ZDA izjavila, da je AES dovolj varen tudi za varovanje zaupnih informacij (Top Secret)

Dolžina ključa	Čas potrebe za razbitje s pomočjo grobe sile
128 bitov	10^{14} let
192 bitov	10^{33} let
256 bitov	10^{52} let

AES: Varnost

- razbitje algoritma pomeni, da je napad mogoče izvesti hitreje kot napad z grobo silo
  - Napad na 128-bitni ključ AES, ki potrebuje 2^{120} operacij (od 2^{128} možnih ključev) se smatra kot razbitje algoritma.
- Uspešnost dosedanjih poskusov:
  - AES vsebuje 10 ciklov za 128-bitni ključ, 12 ciklov za 192-bitni ključ in 14 ciklov za 256-bitni ključ
  - do sedaj je najboljši poskus napada bil izveden na 7 ciklov za 128-bitni ključ, 8 ciklov na 192-bitni ključ in 9 ciklov za 256-bitni ključ

Sodobne assimetrične šifre

RSA

- Rivest – Shamir – Adleman
- 1978
- prvi algoritem asimetrične kriptografije
- karakteristike:
  - dolžina ključev 768 - 4096 bitov
  - ni patentiran (je bil do leta 2003)
  - najbolj razširjen algoritem

RSA: Varnost

- ker lahko izberemo *n* je RSA algoritem s spremenljivo dolžino ključa
- varnost je odvisna od dolžine ključa, običajno 1024 ali več
- za asimetrične algoritme razen običajnih:
  - samo šifrirano sporočilo (groba sila)
  - poznano sporočilo
  - izbrano sporočilo
- poznamo še dva napada:
  - napad na javni ključ
  - napad z izbranim kriptiranim sporočilom

RSA: Napad z grobo silo

- vesolje je staro "samo" 10^{18} sekund…
- da bi preizkusili vse ključe v tem času, bi morali preveriti 10^{58} ključev na sekundo…
- RSA je varen pred napadom z grobo silo

RSA: faktorizacijski napad

- je napad na javni ključ
- napadalec prestreže n in iz njega izračuna p in q
- trenutno nobena poznana metoda faktorizacije ni dovolj učinkovita za napad na 1024 bitni ključ
- za 512 bitov so v letu 1999 uporabili 250 računalnikov več kot 4 mesece

RSA: napad na majhen eksponent

- običajno se uporablja majhna vrednost *e*
- če isto sporočilo pošljemo e prejemnikom in uporabimo isti *e*, je možno odkriti izvorno sporočilo
- v nekaterih primerih tudi, če se ponavlja samo del sporočila

Algoritmi za izmenjavo ključev

Diffie-Hellman izmenjava ključev

- Whitfield Diffie in Martin Hellman, 1976
- Izračunamo število $a=g^x \pmod p$
- Pošljemo število a prejemniku
- Prejemnik izračuna $b=g^y \pmod p$
- Prejemnik pošlje b pošiljatelju

Diffie-Hellman izmenjava ključev

- S pomočjo b pošiljatelj izračuna $k_1 = b^x \pmod p$
- S pomočjo a prejemnik izračuna $k_2 = a^y \pmod p$
- $k_1 = k_2 = \mathbf{k = (g^x)^y = (g^y)^x}$
- Da bi s prisluškovanjem ugotovili **k** bi morali rešiti diskretni logaritem

Varnost Diffie-Hellman

- Za g, x in y lahko izberemo poljubno velikost (bitov)
- Večje je število bitov, bolj varen je algoritem
- Groba sila ni najprimernejša oblika napada

Varnost Diffie-Hellman

- Obstajajo postopki za reševanje diskretnega logaritma, zahtevajo pa veliko procesorske moči
- Danes se uporablja 512 bitov in več

Zgoščevalni algoritmi

Zgoščevalne funkcije

- Modification Detection Code (MDC)
- "prstni odtis" sporočila izdelamo s pomočjo zgoščevalne funkcije (hash)
- če je *f* zgoščevalna funkcija
- h je rezultat funkcije
- *h = f(m)*
- dolžina *h* je omejena, medtem ko je lahko *m* poljubne velikosti
- ni ključa

Zgoščevalne funkcije in dig. podpisi

- zgoščevalna funkcija mora biti izdelana tako, da je čim težje najti trk
- preprečevati mora možnost izračuna drugega vhoda z enakim izvlečkom
- zgoščevalna funkcija ki ustreza vsem trem zahtevam je "kriptografska zgoščevalna funkcija"
- drugi termini: cryptographic hash, Message Digest (MD)

Zgoščevalne funkcije: delovanje

- uporaba principov, ki so podobni simetričnim algoritmom:
  - delitev sporočila v bloke (block)
  - procesiranje vsakega bloka v več ciklih (round)
- nimamo opravka s ključem
- zagotavljanje celovitosti (integrity)
- uporabljajo se preproste operacije – XOR, seštevanje, rotiranje, premikanje (shifting)

Zgoščevalne funkcije: varnost

- trki
  - dva različna vhodna podatka se preslikata v isti izvleček
- predslika
  - na podlagi izvlečka želimo najti podatek, ki se slika v izvleček

MD5

- Message Digest 5 (1992)
- podlaga za SHA
- rezultat je 128-biten
- 1996 prikazane prve pomanjkljivosti
- se smatra kot razbit in se opušča

- možno je poiskati trk v 8 urah (s procesorjem Pentium 1.6 GHz)

MD5

- spletna podatkovna baza za MD5 - http://gdataonline.com/
- 4.8.2009 je imela 1.133.763.180 vnosov
- uporaba:
  - ugotavljanje izvornega gesla iz MD5 zgostitve
  - Ne deluje, če uporabljamo sol (salt)

SHA-1

- Secure Hash Algorithm
- trenutno najpomembnejša kriptografska zgoščevalna funkcija
- algoritem je razvila NSA leta 1991

- bloki velikosti 512 bitov
- 160-bitni izvleček
- do pred kratkim ni bil znan učinkovit napad

SHA-1

- februarja 2005 odkrita prva “resna” pomanjkljivost
- do sedaj je bilo za napad z grobo silo treba preizkusiti 2^{80} zgoščenih vrednosti
- z novim odkritjem je potrebnih “le” še 2^{69}, kar je 2000x manj kot prej
- časovna zahtevnost najnovejšega napada je 2^{63} z možnostjo izboljšanja
- SHA-1 ima vlogo v S/MIME, TLS, IPSec, PKI

Sodobni zgoščevalni algoritmi

- družina SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)
- zaradi novih tehnik napadov na SHA-1 NIST pripravlja razpis za novi kriptografski zgoščevalni algoritem
- odsvetuje uporabo SHA-1
- priporoča uporabo družine SHA-2
- RIPEMD-160 - algoritem razvit v Evropi
- temelji na MD4
- izvleček 160 bitov, podobno kot SHA-1

Prihodnost zgoščevalnih funkcij

- natečaj za SHA-3, ki ga organizira NIST
  - določeni minimalni kriteriji
- 15 kandidatov v drugi fazi izbora
- izbor znan predvidoma 2012

Od ključa odvisne zgoščevalne funkcije

- Message Authentication Check (MAC)
- kadar želimo nadzirati kdo lahko izračuna rezultat zgoščevalne funkcije
- potrebujejo manj računske moči kot elektronski podpis

Pasti

Kriptografske pasti

- neustrezna uporaba algoritmov
  - izdelava lastnih
  - uporaba “šibkih”
  - neustrezna uporaba
- neuspešno varovanje ključev
  - shranjevanje na nezavarovanem mestu
  - predolgo obdobje uporabe
- človeški faktor

Obramba

- periodična reciklaža ključev
- shranimo ključe na zunanji napravi
- uporabimo dovolj dolge ključe
- ne implementiramo lastnih kriptografskih algoritmov