Heš funkcije
Deljenje tajni
Slučajnost

Sadržaj

- Autentifikacija poruke
- Jednosmerne heš funkcije
- MD5
- SHA
- Zaštita heš vrednosti
- HMAC
- Deljenje tajni
- Slučajnost

Zaštita od aktivnih napada

- Šifrovanje nudu zaštitu od **pasivnih** napada.
- Kako se zaštititi od aktivnih napada?
  - Zaštita od ovakvih napada može da se realizuje kroz **autentifikaciju** poruke.
- Poruka, datoteka ili drugi skup podatataka je autentifikovan:
  - kada nije bilo izmena na prenosnom putu,
  - kada je se može utvrditi poreklo poruke (ko je poslao poruku).

Autentifikacija poruke – zahtevi

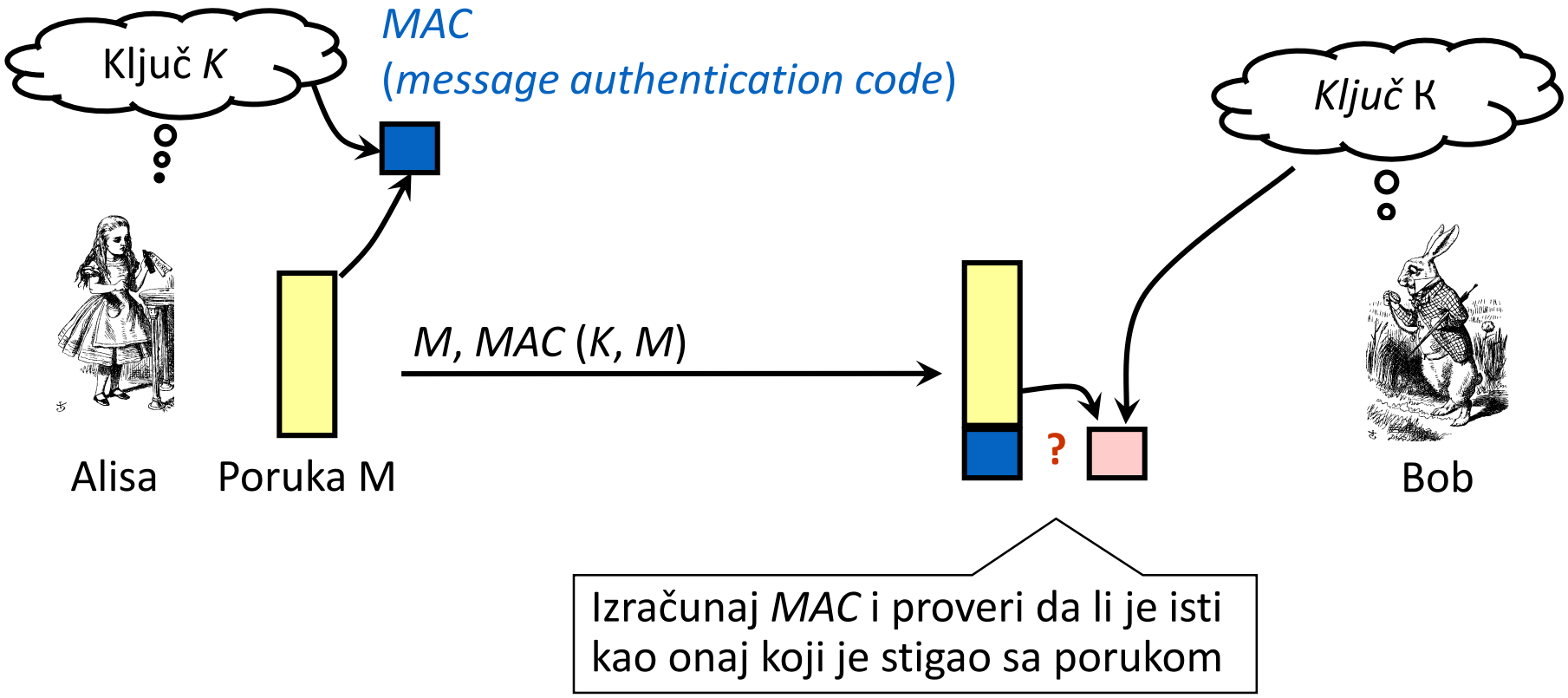
- Mogućnost da se ustanovi da li je primljena poruka **izmenjena na prenosnom putu**.
- Mogućnost da se ustanovi **identitet pošiljaoca**.
- Mogućnost da se ustanovi **vreme kada je poruka poslata**.
  - Poruka može biti snimljena i ponovo poslata ...
  - Primer:
    - Alisa je javila banci da na Bobov račun uplati 100$.
    - Bob snimi poruku i nakon 15 dana ponovo pošalje banci ...

Autentifikacija i simetrični kriptografski sistemi

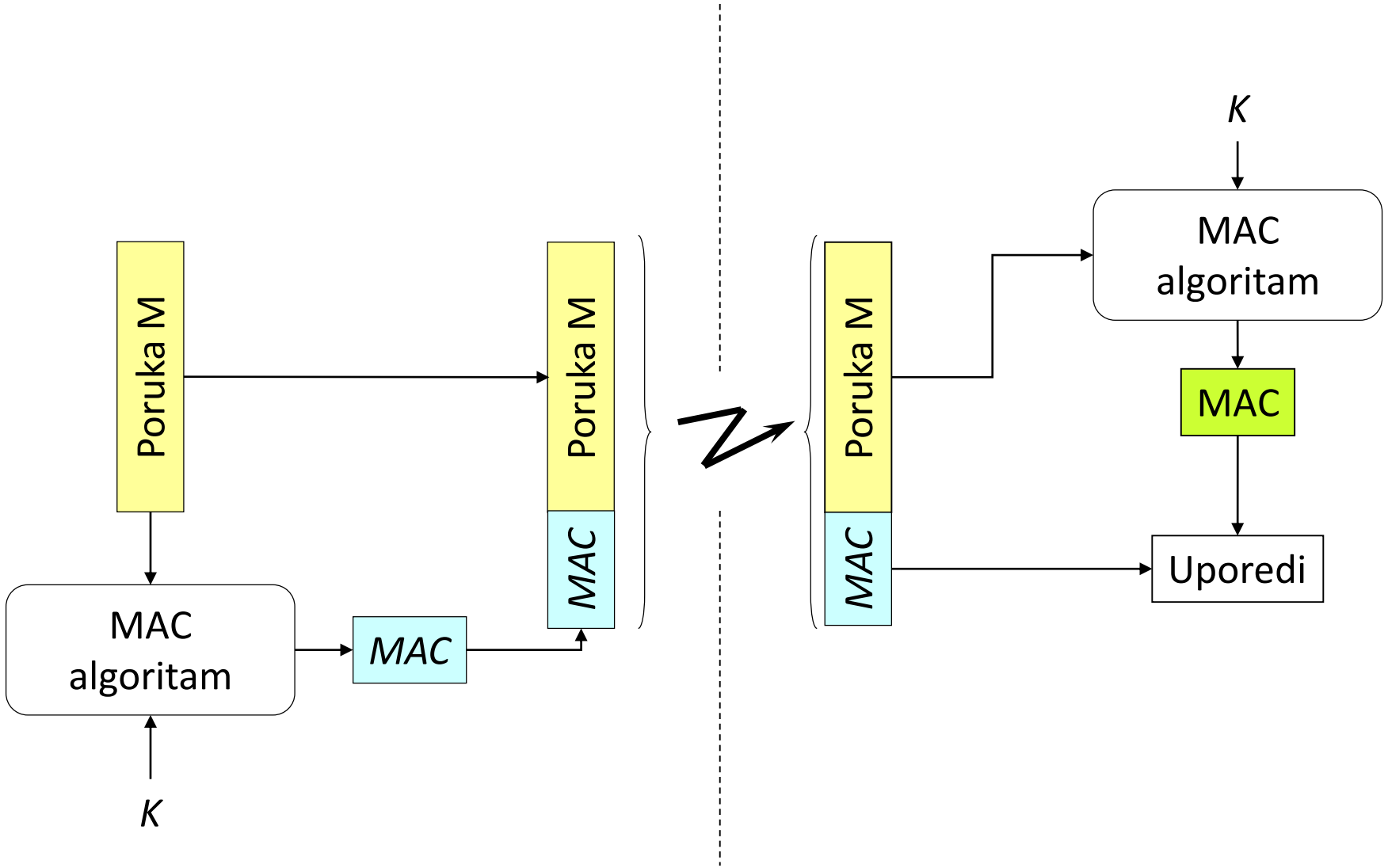
- **Nema servisa neporecivosti**, jer obe strane poznaju ključ.
- Ako postoji poverenje između strana u komunikaciji:
  - Softver za detekciju greške može da registruje moguće izmene na prenosnom putu.
  - Ako poruka sadrži **podatak o vremenu slanja** (*timestamp*), može se otkriti pokušaj ponovnog slanja iste poruke.

Autentifikacija poruke bez šifrovanja poruke

- Integritet i autentifikacija: samo onaj ko **zna ključ** može da izračuna MAC za datu poruku.



Autentifikacija poruke – MAC



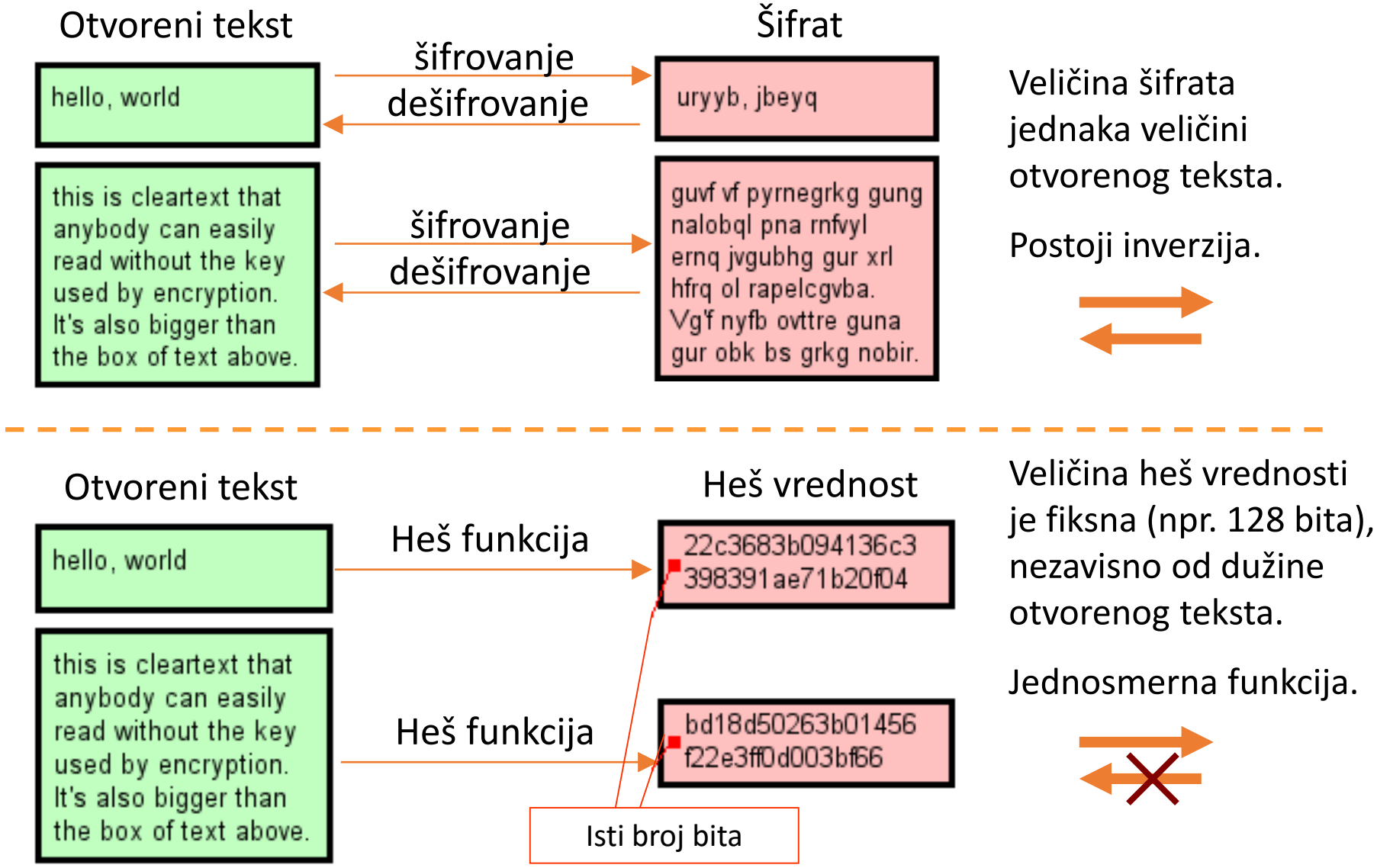
Autentifikacija poruke – MAC

- Ako se pretpostavi da samo dve strane u komunikaciji znaju tajnu vrednost ključa *K*:
  - prijemna strana na osnovu provere MAC vrednosti ima informaciju o tome **da li je poruka menjana** na prenosnom putu,
  - prijemna strana zna **ko je poslao poruku** (samo onaj ko zna ključ može da izračuna MAC).
- Ako poruka sadrži **podatak o vremenu slanja**, izbegava se mogućnost napada ponovnog (zakašnjenog) slanja iste poruke.

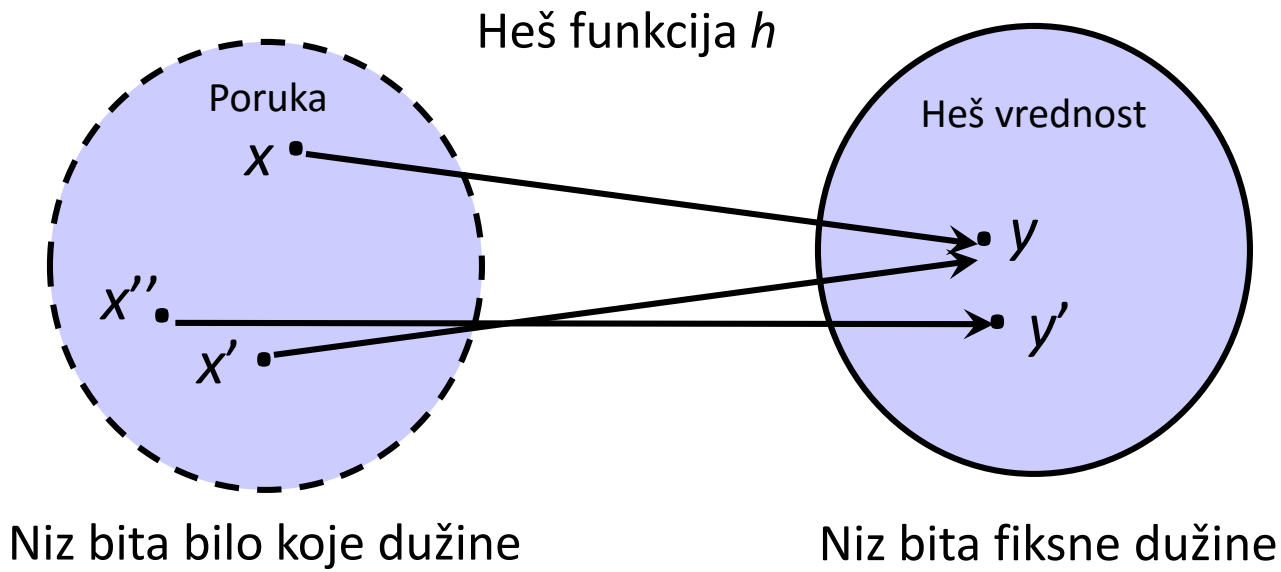
Jednosmerne heš funkcije

- Kriptografska heš funkcija je jednosmerna funkcija koja za ulazni podatak (poruka, datoteka, ...) proizvoljne konačne dužine kao izlaznu vrednost daje niz fiksne dužine.
  - **Ulaz** je niz bitova proizvoljne konačne dužine.
    - U nekim (retkim, specijalnim uvrnutim) slučajevima je ograničena maksimalna dužina poruke!
  - **Izlaz** je heš vrednost uvek iste konačne dužine.
- Namena: autentifikacija poruke, provera integriteta.
- Heš funkcija ne služi za šifrovanje!

Poređenje šifrovanja i heš funkcije



Heš funkcije: osnovna ideja



- *h* predstavlja funkciju kompresije sa gubicima.
  - Kolizije: $h(x)=h(x')$ za neke ulaze x, x'.
  - Rezultat heš funkcije treba da ima osobine slučajanosti.
- Kriptografske heš funkcije treba da imaju i dodatna svojstva …

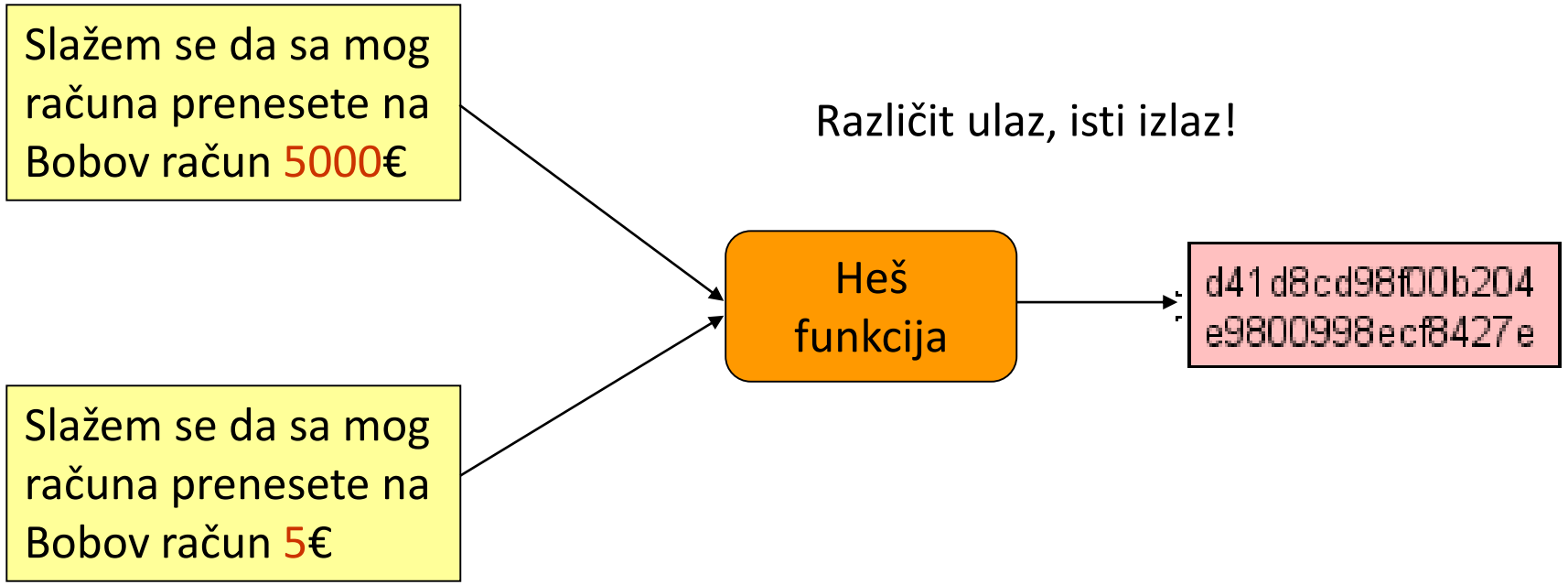
Heš funkcije: osnovna ideja

- Heš funkcija **ne sme biti invertibilna**:
- "*Preimage resistance*"
  - Neka je $h(x')=y \in \{0,1\}^n$ za neko slučajno x'.
  - Za dato y, treba da bude teško da se nađe bilo koje x, takvo da je $h(x)=y$.
- Koliko teško?
  - Potpuna pretraga: probati svako moguće x, proveriti da li je $h(x)=y$.
  - SHA-1 heš funkcija ima izlaz dužine 160 bita.
    - Neka postoji hardver takav da je moguće probati 2^{34} vrednosti (različitih poruka h) u sekundi.
    - Sledi da će se za godinu dana ispitati 2^{89} vrednosti h.
    - Potrebno je 2^{71} godina da se invertuje SHA-1 heš vrednost.

Kriptografske heš funkcije: zahtevi

- Kriptografska heš funkcija $h(x)$ mora da obezbedi:
  - **Kompresiju**.
    - Izlaz mora da bude male dužine.
  - **Efikasnost**.
    - $h(x)$ treba da se lako računa za bilo koje x.
  - **Jednosmernost**.
    - Za zadato y praktično je nemoguće naći x takvo da je $h(x) = y$.
  - **Otpornost na kolizije**.
    - Za zadato x i $h(x)$, praktično je nemoguće naći $y \neq x$, takvo da je $h(y) = h(x)$.
    - Praktično je nemoguće naći bilo koje x i y, $x \neq y$ takve da je $h(x) = h(y)$.
    - Kolizije postoje ali heš funkcija treba da je projektovana tako da ih je teško pronaći!

Primer kolizije



- Ako je relativno lako naći kolizije, Alisa može da tuži banku i traži povraćaj novca, tvrdeći da je poslala drugu poruku.

Rođendanski problem – uvod

- Neka se u sobi nalazi *n* osoba.
  - Kolika je verovatnoća da neka (jedna) osoba nema rođendan istog datuma kao i ja (slučajan odabir)?
    - Rešenje: 364/365
      - (broj povoljnih/broj mogućih događaja).

Rođendanski problem – uvod

- Neka se u sobi nalazi *n* osoba.
  - Kolika je verovatnoća da neka (jedna) osoba nema rođendan istog datuma kao i ja (slučajan odabir)?
    - Rešenje: 364/365
      - (broj povoljnih/broj mogućih događaja).
  - Kolika je verovatnoća da ni jedna od *n* osoba nema ...?
    - Rešenje: $(364/365)^n$
      - (jedna osoba nema, i druga nema .... i n-ta nema)

Rođendanski problem – uvod

- Neka se u sobi nalazi *n* osoba.
  - Kolika je verovatnoća da neka (jedna) osoba nema rođendan istog datuma kao i ja (slučajan odabir)?
    - Rešenje: 364/365
      - (broj povoljnih/broj mogućih događaja).
  - Kolika je verovatnoća da ni jedna od *n* osoba nema ...?
    - Rešenje: $(364/365)^n$
      - (jedna osoba nema, i druga nema .... i *n*-ta nema)
  - Kolika je verovatnoća da bar jedna osoba ima ...?
    - Rešenje: $1 - (364/365)^n$

Rođendanski problem – uvod

- Neka se u sobi nalazi *n* osoba.
  - Kolika je verovatnoća da neka (jedna) osoba nema rođendan istog datuma kao i ja (slučajan odabir)?
    - Rešenje: 364/365
      - (broj povoljnih/broj mogućih događaja).
  - Kolika je verovatnoća da ni jedna od *n* osoba nema ...?
    - Rešenje: $(364/365)^n$
      - (jedna osoba nema, i druga nema .... i n-ta nema)
  - Kolika je verovatnoća da bar jedna osoba ima ...?
    - Rešenje: $1 - (364/365)^n$
  - Koliko mora biti *n* da bi verovatnoća da bar jedna osoba ima rođendan istog datuma kao i ja bila veća ili jednaka 1/2?
    - Rešenje: $1/2 \leq 1 - (364/365)^n$
      - Odavde je $n \geq 253$

Rođendanski problem

- Kolika je verovatnoća da niko od *n* osoba u sobi nema rođendan istog datuma?
  - Rešenje: $364/365 \cdot 363/365 \cdot \ldots \cdot (365-n+1)/365$

Rođendanski problem

- Kolika je verovatnoća da niko od *n* osoba u sobi nema rođendan istog datuma?
  - Rešenje: 364/365 · 363/365 · … · (365-*n*+1)/365
- Kolika je verovatnoća da od *n* osoba dvoje ili više ljudi ima rođendan istog datuma?
  - Rešenje: 1 - 364/365 · 363/365 · … · (365-*n*+1)/365

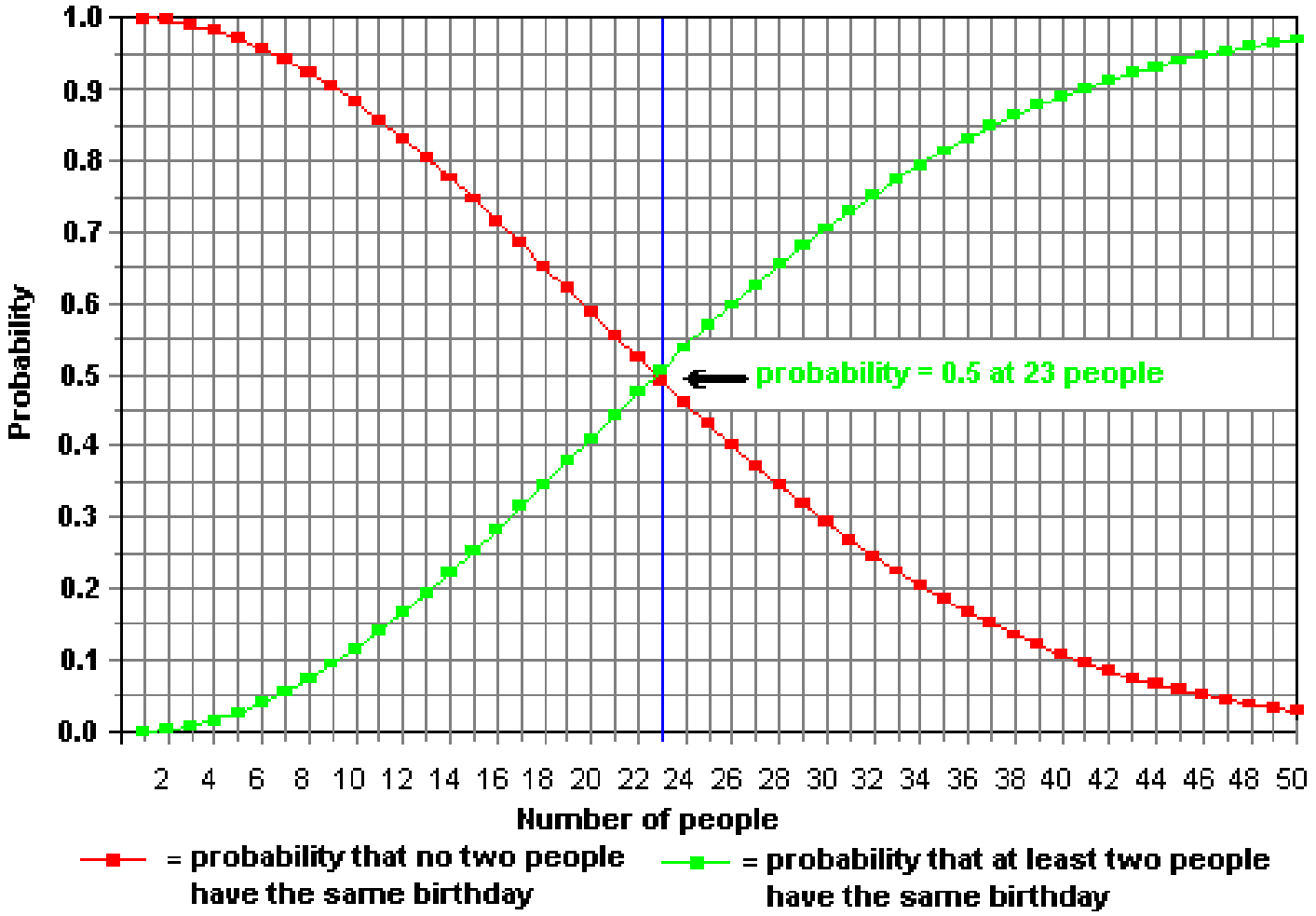
Rođendanski problem

- Kolika je verovatnoća da niko od *n* osoba u sobi nema rođendan istog datuma?
  - Rešenje: $364/365 \cdot 363/365 \cdot \ldots \cdot (365\text{-}n\text{+}1)/365$
- Kolika je verovatnoća da od *n* osoba dvoje ili više ljudi ima rođendan istog datuma?
  - Rešenje: $1 - 364/365 \cdot 363/365 \cdot \ldots \cdot (365\text{-}n\text{+}1)/365$
- Koliko mora biti najmanje ljudi u sobi, da bi verovatnoća da dvoje ili više ljudi ima rođendan istog datuma, bila $\geq 1/2$?
  - $1/2 \leq 1 \cdot 364/365 \cdot 363/365 \cdot \ldots \cdot (365\text{-}n\text{+}1)/365 \rightarrow n \geq 23$

Rođendanski problem

- Kolika je verovatnoća da niko od n osoba u sobi nema rođendan istog datuma?
  - Rešenje: $364/365 \cdot 363/365 \cdot \ldots \cdot (365\text{-}n\text{+}1)/365$
- Kolika je verovatnoća da od n osoba dvoje ili više ljudi ima rođendan istog datuma?
  - Rešenje: $1 - 364/365 \cdot 363/365 \cdot \ldots \cdot (365\text{-}n\text{+}1)/365$
- Koliko mora biti najmanje ljudi u sobi, da bi verovatnoća da dvoje ili više ljudi ima rođendan istog datuma, bila $\geq 1/2$?
  - $1/2 \leq 1 \cdot 364/365 \cdot 363/365 \cdot \ldots \cdot (365\text{-}n\text{+}1)/365 \rightarrow n \geq 23$
- Iznenađenje? Paradoks?
  - Možda i nije.
  - Broj poređenja je $n(n\text{-}1)/2 \approx n^2$.
  - Kako je broj dana u godini 365, može se očekivati poklapanje za $n^2 = 365 \rightarrow n \approx 19$.

Rođendanski problem



Slika preuzeta sa: http://www.people.virginia.edu/~rjh9u/birthday.html

Heš funkcije i rođendanski problem

- Kakva veza postoji?
  - Ako heš funkcija $h(x)$ daje izlaz dužine N bita, onda postoji 2^N različitih heš vrednosti.
  - $\sqrt{2^N} = 2^{N/2}$
  - Iz ovoga sledi da, ako se izračunaju heš funkcije za oko $2^{N/2}$ slučajnih vrednosti, očekuje se pojava jedne kolizije (2 različita ulaza daju isti heš).
  - **Posledica**.
    - Simetrični ključ dužine N bita zahteva $2^N/2 = 2^{N-1}$ operacija da bi se razbio.
    - Za isti posao u slučaju heša dužine N bita potrebno je $2^{N/2}$ operacija.
  - **Zaključak**.
    - Izlaz heš funkcije mora biti 2 puta duži od ključa simetričnog šifarskog sistema da bi imao isti nivo sigurnosti!

Nekriptološki heš

- Neka su ulazni podaci $X = (X_0, X_1, X_2, \ldots, X_{n-1})$, gde svako X_i predstavlja jedan bajt.
- Neka je $h(X) = X_0 + X_1 + X_2 + \ldots + X_{n-1} \pmod{256}$
- Kompresija postoji – izlaz je uvek dužine 8 bita.
- Da li je siguran?
  - $2^{N/2} = 2^4 = 16$
  - Primer:
    - X = (10101010, 00001111)
    - Heš je 10111001
    - Ali ovo je heš i za ulaz Y = (00001111, 10101010)
    - Zaključak: lako se nalaze kolizije, pa ovo **nije** siguran heš.

Nekriptološki heš

- Cyclic Redundancy Check (CRC).
  - Dobar je za detekciju usmerenih (*burst*) grešaka.
  - Lako se mogu naći kolizije.

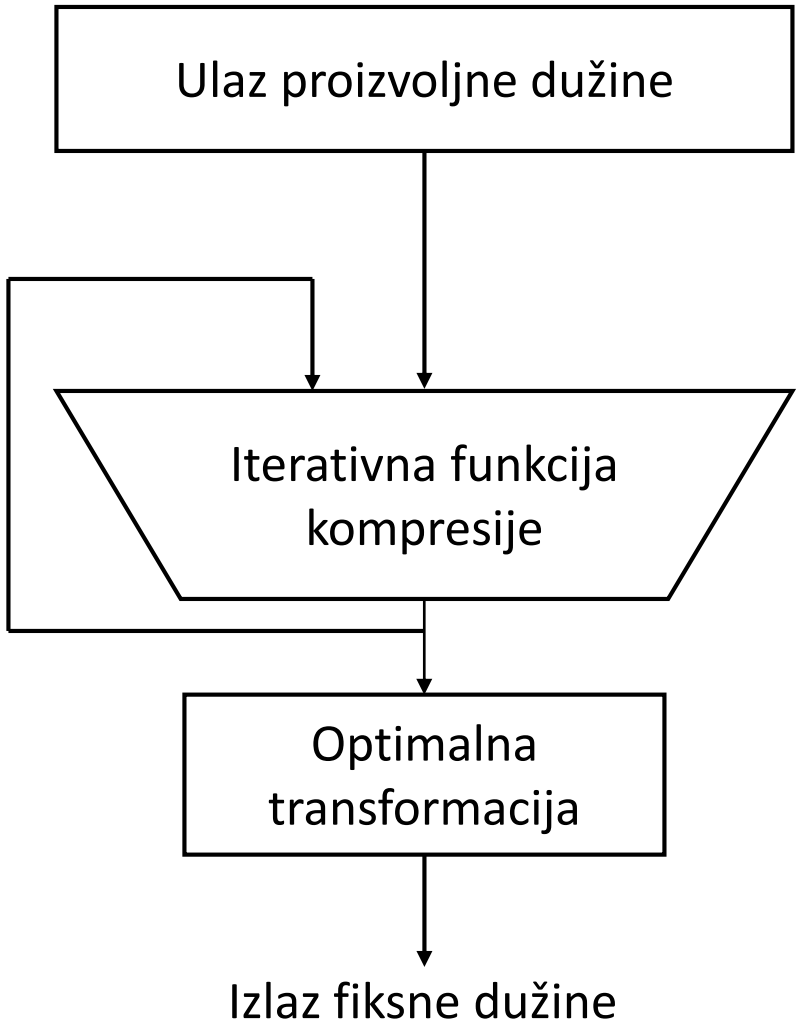
Kriptološke heš funkcije

- MD5.
  - Autor Ron Rivest.
  - Heš vrednost dužine 128 bita.
  - Primedba: MD5 kolizije su pronađene.
- RIPEMD-160.
  - 160-bitna varijanta MD5.
- SHA-1, SHA-2, SHA-3.
  - Standard američke vlade (u osnovi sličan sa MD5).
  - SHA-1: Heš vrednost dužine 160 bita.
  - SHA-2: Heš vrednost dužine $\leq$ 512 bita.
  - SHA-3: Heš vrednost dužine $\leq$ 512 bita.
- Poruka se prvo deli na blokove a potom se računa heš vrednost svakog bloka.

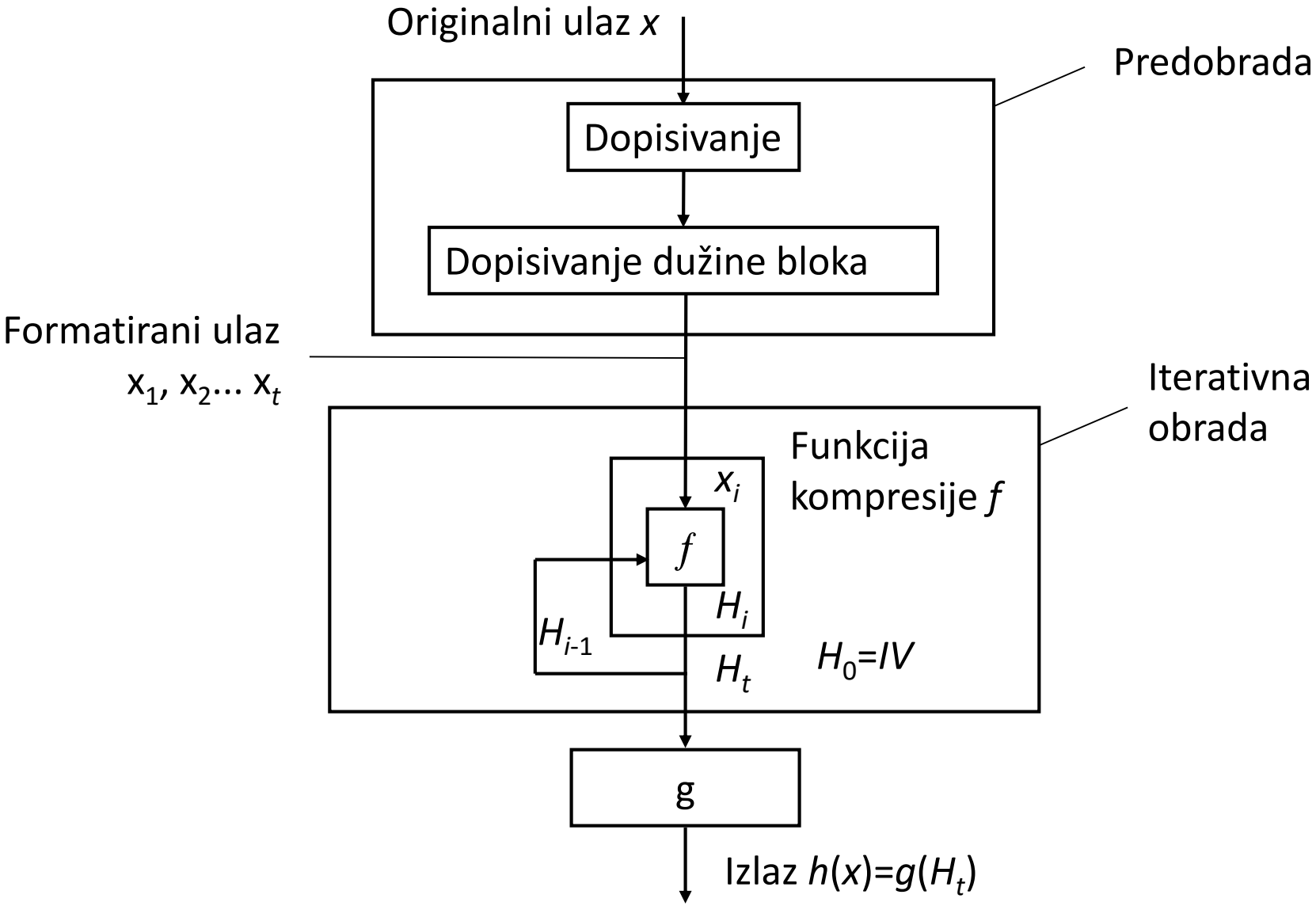
Kriptološke heš funkcije

- Pored navedenih osobina poželjno je da poseduju i osobinu **lavinskog efekta**.
  - Promena u jednom bitu ulaza (poruke) izaziva promene u bar polovini izlaznih bita (heš vrednosti).
- Kriptološke heš funkcije se sastoje od nekoliko rundi.
- Poželjna su brzina i sigurnost.
  - Lavinski efekat treba da se manifestuje već nakon nekoliko rundi.
  - Runde moraju da budu jednostavne za izvršavanje.
- Dizajn sličan blokovskim šiframa.

Opšti model iterativnih heš funkcija



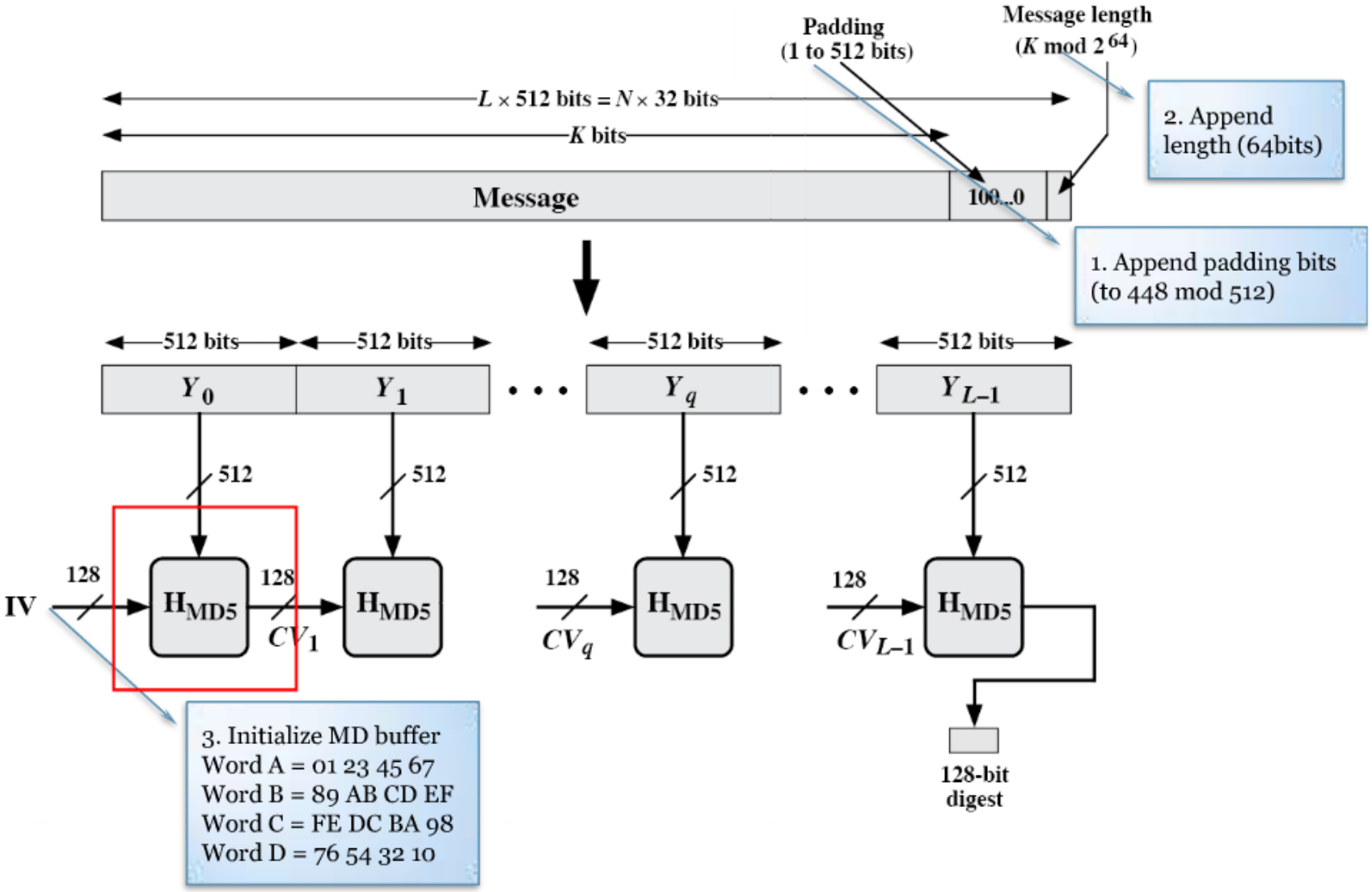
Detaljnije …


Originalni ulaz x
Predobrada
Dopisivanje
Dopisivanje dužine bloka
Formatirani ulaz
x1, x2... xt
Iterativna
obrada
Funkcija
kompresije f
xi
f
Hi
Hi-1
Ht
H0=IV
g
Izlaz h(x)=g(Ht)

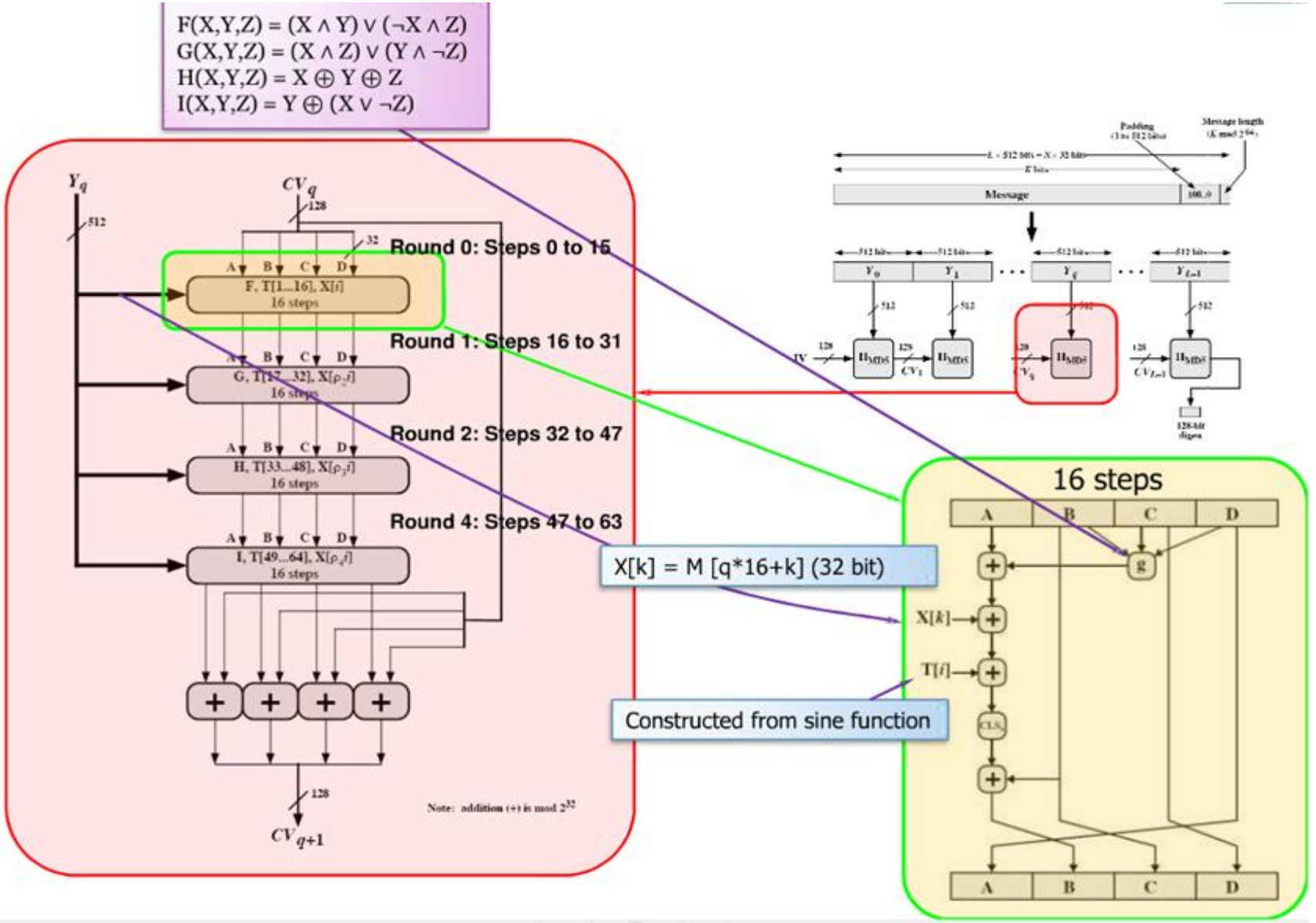
MD5

- MD – skraćenica od *message digest*.
- Jednosmerna heš funkcja koja daje izlaz od 128 bita.
- Ulaz proizvoljne dužine.
- Predstavlja unapređenu verziju funkcije MD4.

MD5


Padding
(1 to 512 bits)
Message length
(K mod 2 64)
L × 512 bits = N × 32 bits
K bits
Message
100...0
2. Append
length (64bits)
1. Append padding bits
(to 448 mod 512)
512 bits
512 bits
512 bits
512 bits
Y 0
Y 1
Y q
Y L–1
512
512
512
512
IV
128
128
128
128
H MD5
H MD5
H MD5
H MD5
CV 1
CV q
CV L–1
3. Initialize MD buffer
Word A = 01 23 45 67
Word B = 89 AB CD EF
Word C = FE DC BA 98
Word D = 76 54 32 10
128-bit
digest

MD5


F(X,Y,Z) = (X ∧ Y) ∨ (¬X ∧ Z)
G(X,Y,Z) = (X ∧ Z) ∨ (Y ∧ ¬Z)
H(X,Y,Z) = X ⊕ Y ⊕ Z
I(X,Y,Z) = Y ⊕ (X ∨ ¬Z)
Round 0: Steps 0 to 15
Round 1: Steps 16 to 31
Round 2: Steps 32 to 47
Round 4: Steps 47 to 63
16 steps
X[k] = M [q*16+k] (32 bit)
Constructed from sine function
Note: addition (+) is mod 2^32
Message

MD5

- Na poruku dopisati bite tako da ukupna dužina bude 448 (mod 512).
  - Poruka se deli na blokove od 512 bita
  - U poslednji blok treba dopisati 64 (512-448) bita.
- U poslednji blok upisati 64-bitnu vrednost dužine originalne poruke (dužina pre prethodnog koraka).
  - Ukupna dužina formatirane poruke je k*512 bita.
- Inicijalizovati MD bafer koji čuva trenutne (i konačne) vrednosti funkcije (četiri 32-bitska registra A, B, C i D).

MD5

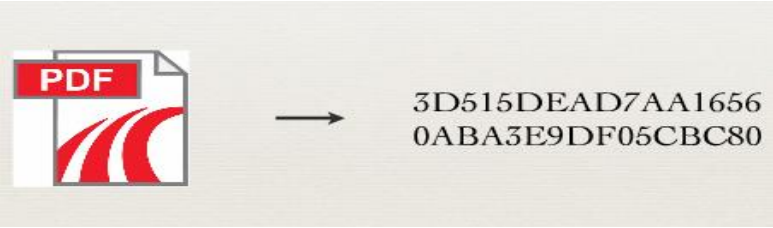
- Obrada poruke podeljena u blokove od 512 bita:
  - Svaki blok se podvrgava obradi u 4 runde.
  - Svaka runda ima sličnu strukturu ali različite logičke fuinkcije.
  - Svaka runda ima ulaz od 512 bita i trenutne vrednosti registara ABCD.
  - Nakon svake runde se modifikuje vrednost registara.
- Nakon obrade svih blokova dobija se izlaz – heš vrednost dužine 128 bita.

MD5

- Nostradamusov napad.
  - 30.11.2007. objavljeno, http://www.win.tue.nl/hashclash/Nostradamus/

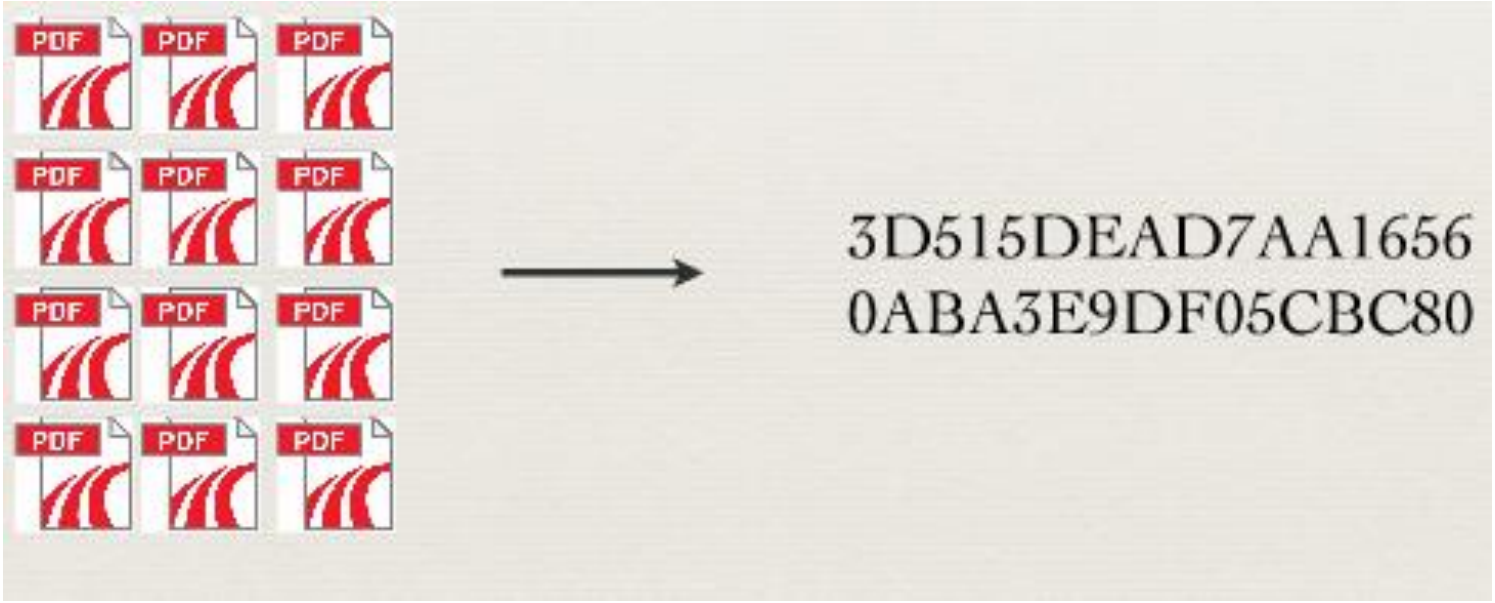
> *"We have used a Sony Playstation 3 to correctly predict the outcome of the 2008 US presidential elections. In order not to influence the voters we keep our prediction secret, but commit to it by publishing its cryptographic hash on this website. The document with the correct prediction and matching hash will be revealed after the elections."*
>
> - Marc Stevens, Arjen Lenstra and Benne de Weger

  - Kao dokaz, ponuđena je MD5 heš vrednost.



MD5

- Nostradamusov napad.
  - Kako su znali?
  - Nisu. Napravili su više dokumenata sa istim hešom.
  - Kolizije očigledno postoje!



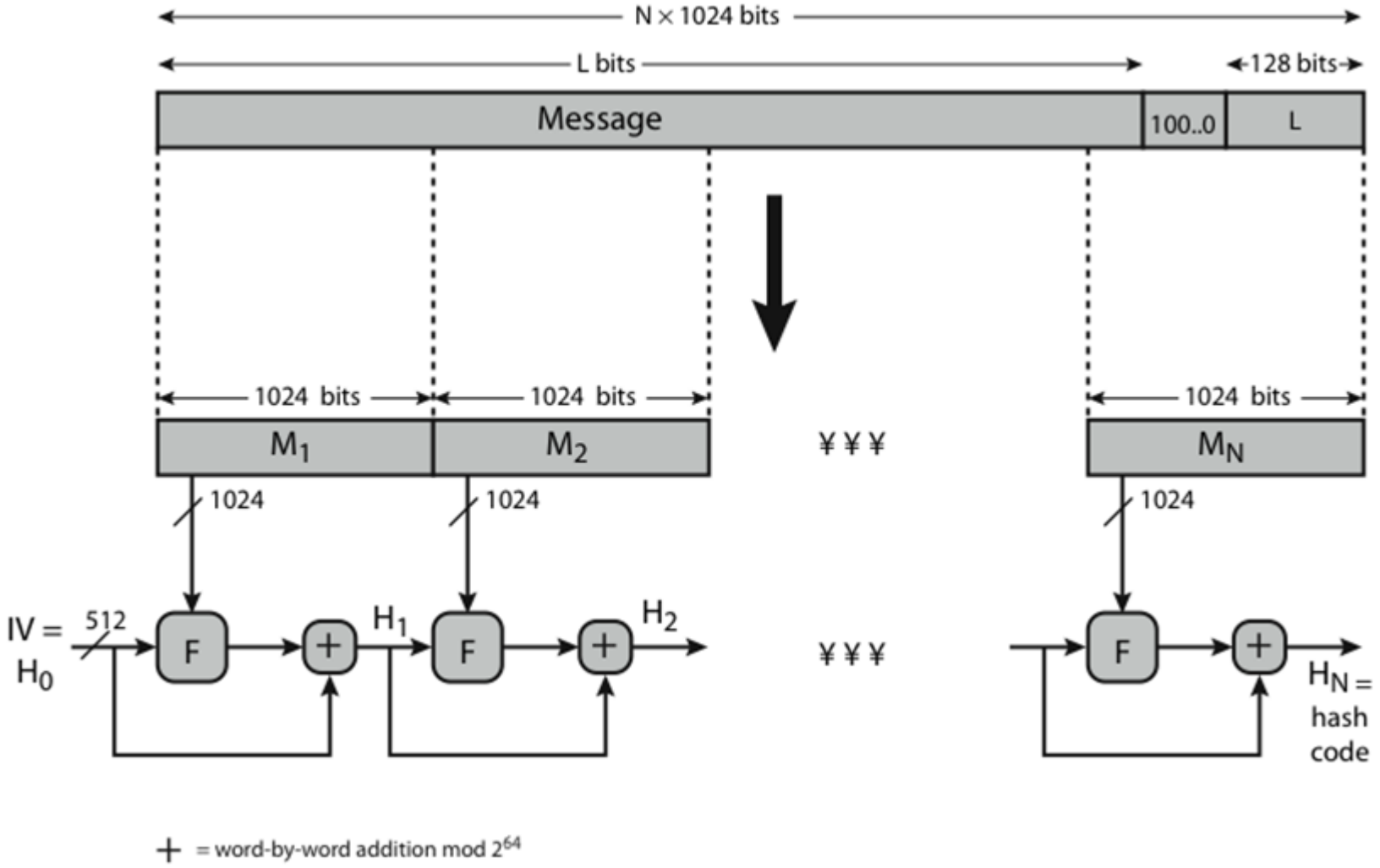
MD5

- Već 1996. godine su otkriveni neki od nedostataka MD5 heš funkcije, mada u to vreme nisu
- predstavljali realnu opasnost.
- 1. marta 2005, Arjen Lenstra, Xiaoyun Wang, i Benne de Weger demonstrirali slabosti MD5 heš funkcije.
- Nekoliko dana kasnije predstavljen je bolji algoritam za generisanje MD5 kolizija koji može da se izvrši za nekoliko časova na jednom *notebook* računaru.
- 18. marta 2006. objavljen algoritam koji može pronaći kolizije za jedan minut na jednom *notebook* računaru koristeći efekat nazavan *tunneling*.

SHA

- *Secure Hash Algorithm*.
- Postoje četiri grupe algoritama: SHA-0, SHA-1, SHA-2 i SHA-3.
- **SHA-0** je dizajnirala NSA, a objavio NIST 1993. godine.
- **SHA-1** predstavlja reviziju prethodne verzije.
  - SHA-1 je je najduže korišćen.
  - Ustanovljen 1995. godine, 2005. godine su otkriveni potencijalni nedostaci.
  - Heš vrednost dužine 160 bita.
- NIST je izdao reviziju standarda 2002. godine i predložio nove verzije.
- **SHA-2** predstavlja familiju algoritama koji se razlikuju po dužini heš vrednosti i označavaju se: SHA-256, SHA-384, i SHA-512.
  - **SHA-2** predstavlja varijantu prethodnog rešenja.
- **SHA-3** je objavio NIST 5. avgusta 2015. godine.
  - “SHA-3 se u osnovi razlikuje značajno od MD5-olike structure karakteristične za SHA-1 i SHA-2.”
  - Familija algoritama: SHA3-224, SHA3-256, SHA3-384, SHA3-512.

SHA-512

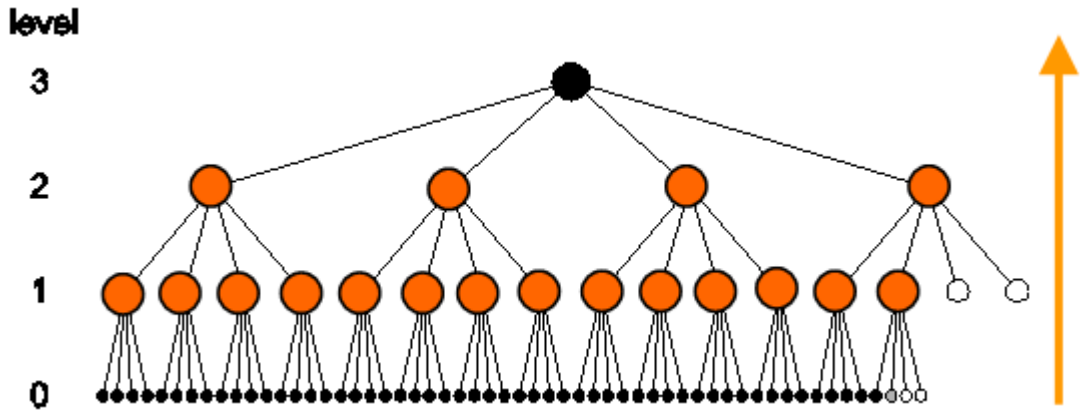

N × 1024 bits
L bits
128 bits
Message
100..0
L
1024 bits
1024 bits
1024 bits
M1
M2
MN
1024
1024
1024
IV =
H0
512
F
H1
F
H2
F
HN =
hash
code
+ = word-by-word addition mod 2^64

SHA-512

- SHA-512 se sastoji iz sledećih koraka:
  - 1. Dopisivanje bita
  - 2. Dopisivanje dužine
  - 3. Inicijalizacija bafera
  - 4. Deljenje poruke na blokove od 1024 bita
  - 5. Obrada
  - 6. Genrisanje izlazne vrednosti.
- Funkcija kompresije predstavlja osnovu algoritma!
  - Deluje na blok podataka dužine 1024 bita.
  - Obavlja se u 80 rundi.
  - Kao ulaz koristi:
    - 512 bita iz bafera (ABCDEFGH).
    - 64 bitna vrednosta *Wt* koja se dobija iz tekućeg bloka poruke.
    - Konstanta *Kt*.
    - ...

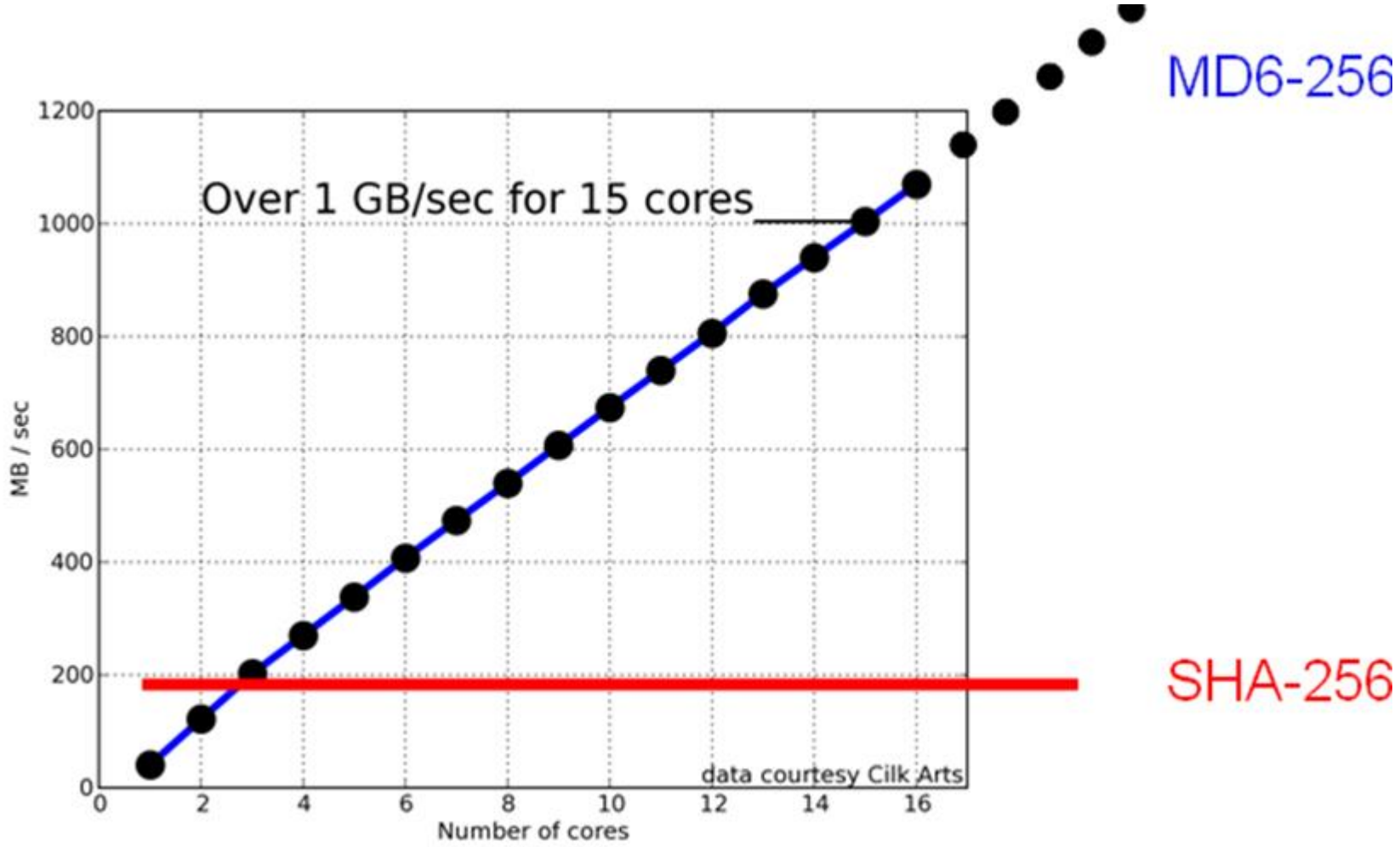
MD6

- MD6 je predložen kao kandidat za novi standard (NIST SHA-3).
- 1. jula 2009. godine Ron Rivest je poslao komentar da još nije spreman da kandiduje konačnu verziju.
- Veliki ulazni blok podataka: 512 bajtova (ne 512 bita)
- Koncipiran za primenu sa procesorima koji imaju više jezgara.
- Ima kompresiju 4:1 u svakom nivou.



MD6

- Efikasnost.



MD6

- Prilagođen je za rad na 64-bitnim vrednostima.
- Koristi jednostavne operacije:
  - XOR
  - AND
  - *Shift*
- Izlaz (heš vrednost) može da ima različitu dužinu: 160, 224, 256, 384, 512.
- Ima promenljivi broj rundi (svaka runda 16 koraka).

Heš funkcije – trenutno stanje

- Ne koristiti MD5, MD4.
- SHA-1 se ne preporučuje za upotrebu od 2009. godine.
- Može se koristiti SHA-2.
- Detaljnija kriptoanaliza SHA-3 se očekuje.

Potreba za heš funkcijama

- Integritet
- Autentifikacija
- Digitalni potpis

Provera integriteta i heš funkcije

- Heš funkcija može da se koristi za proveru integriteta.

Bob

Alisina banka

$(M, h(M))$ ⟶ $(M', h(M)')$

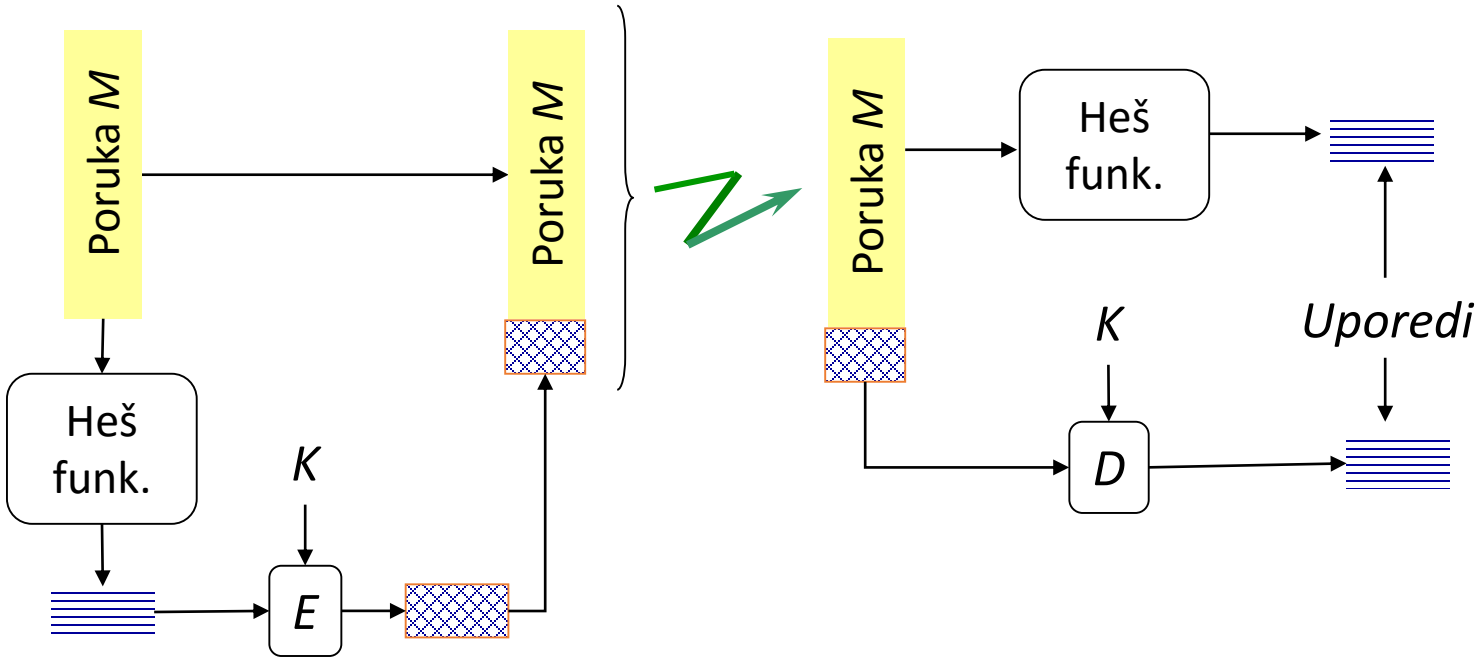
- Banka proverava da li je $h(M') = h(M)'$
- Eliminacija grešaka u prenosu.
- Problem: napad tipa **čovek u sredini**.
  - Trudi može da napiše novu poruku i da pošalje ispravnu heš vrednost!

Zaštita heš vrednosti

- Heš vrednost može da se zaštiti:
  - Primenom algoritama sa simetričnim ključem
  - Primenom algoritama sa javnim ključem
  - Koristeći tajnu vrednost.

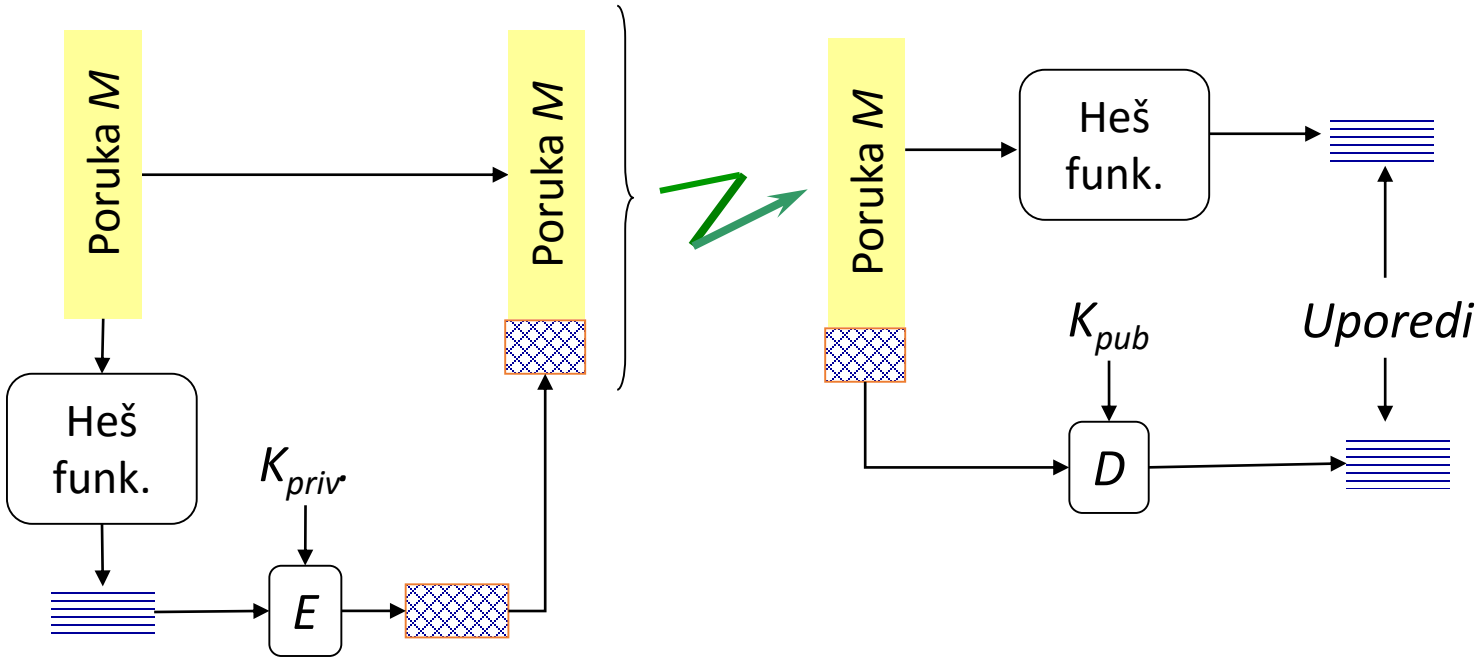
Zaštita heš vrednosti

- Heš vrednost može da se šifruje:
  - Primenom algoritama sa **simetričnim ključem**.



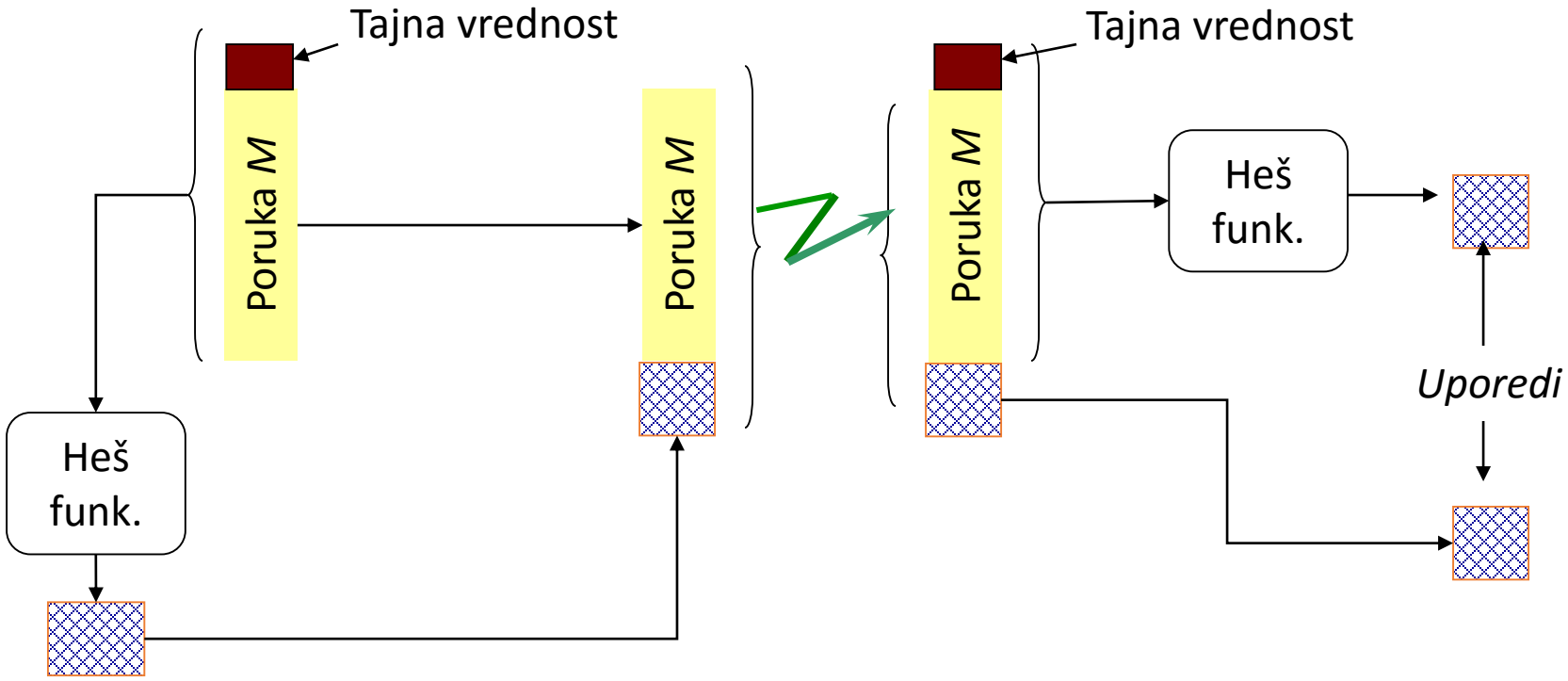
Zaštita heš vrednosti

- Heš vrednost može da se digitalno potpiše:
  - Primenom algoritama sa **javnim ključem**.



Zaštita heš vrednosti

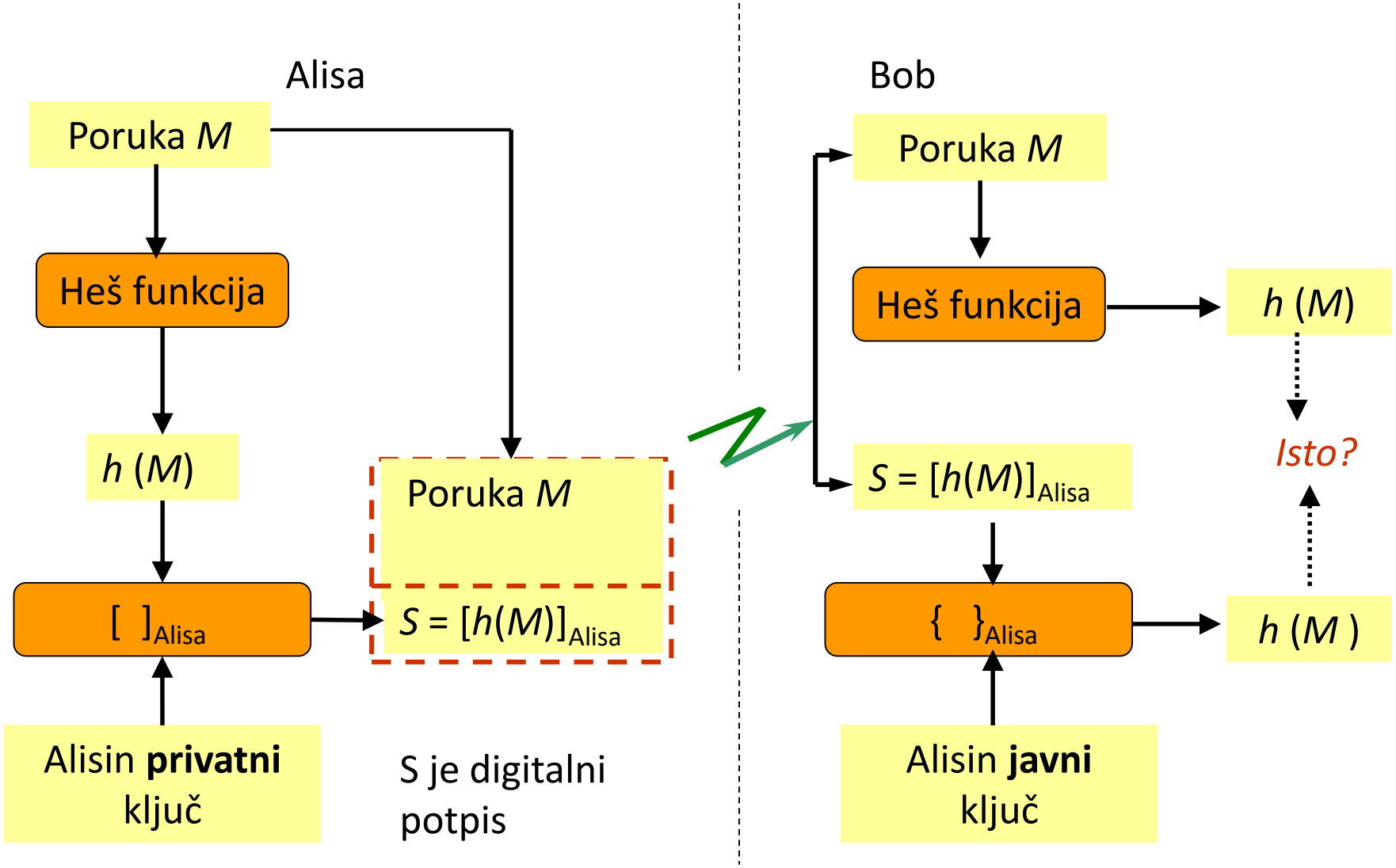
- Heš vrednost može da se zaštiti:
  - Primenom **tajne vrednosti**.
  - U ovom slučaju nema šifrovanja!



Digitalni potpis i heš funkcije

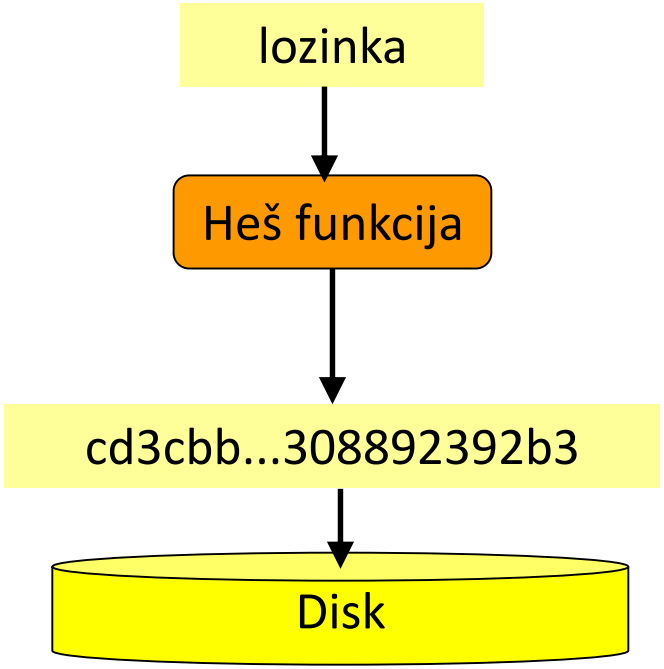
- Neka Alisa digitalno potpiše poruku M.
  - Alisa šalje Bobu M i $S=[M]_{\text{Alisa}}$.
  - Bob proverava da li je $M=\{S\}_{\text{Alisa}}$.
  - Ako je M veliko, $[M]_{\text{Alisa}}$ je računski zahtevno.
- Umesto toga, neka je digitalni potpis rezultat javne funkcije $h(M)$, gde je dužina izlaza funkcije $h(M)$ mnogo manja od dužine M.
  - Alisa šalje Bobu M i $S=[h(M)]_{\text{Alisa}}$.
  - Bob proverava da li je $h(M)=\{S\}_{\text{Alisa}}$.

Digitalni potpis i heš funkcije



Još ponešto o primeni heš funkcija

- **Provera integriteta**.
  - Preuzmete datoteku sa mreže, kako možete da znate da nije izmenjen u toku prenosa?
  - Pored datoteke može da se nalazi njena heš vrednost (digitalno potpisana).
  - Nakon prijema izračunate heš vrednost datoteke i uporediti je.
- **Čuvanje lozinke**.
  - Na disku se čuva samo heš vrednost.
  - Prilikom provere lozinke proverava se samo podudaranje heš vrednosti.



Još ponešto o primeni heš funkcija

- Neka su Alisa, Bob i Čarli ponuđači.
- Alisa daje ponudu *A*, Bob *B*, a Čarli *C*.
- Oni ne veruju da će ponude ostati tajne.
- Rešenje?
  - Alisa, Bob i Čarli dostavljaju heš vrednosti ponuda $h(A)$, $h(B)$, $h(C)$.
  - Sve heš vrednosti se javno publikuju.
  - Ponude se ne mogu menjati nakon što su poslate heš vrednosti.
  - U zakazano vreme se otkrivaju ponude A, B i C.
  - Heš vrednosti ne otkrivaju ponude (jednosmernost).
- Kolizije?

HMAC

- Predstavlja MAC dobijen upotrebom kriptografske heš funkcije (MD5, SHA, ...).
  - Tradicionalni MAC je zasnovan na simetričnim blok šiframa.
- **Motivacija**:
  - Kriptografske heš funkcije se izvršavaju brže od simetričnih šifarskih algoritama (DES, ...).
  - Kod kriptografskih heš funkcija je javno dostupan.
  - Nema izvoznih ograničenja (iz SAD).
- **Zahtevi za dizajn**:
  - Da koriste, bez modifikacije, dostupne heš funkcije.
  - Da sačuvaju originalna svojstva heš funkcije.
  - Da koriste ključeve na jednostavan način.
  - Da analiza kriptografskih svojstava bude jasna.

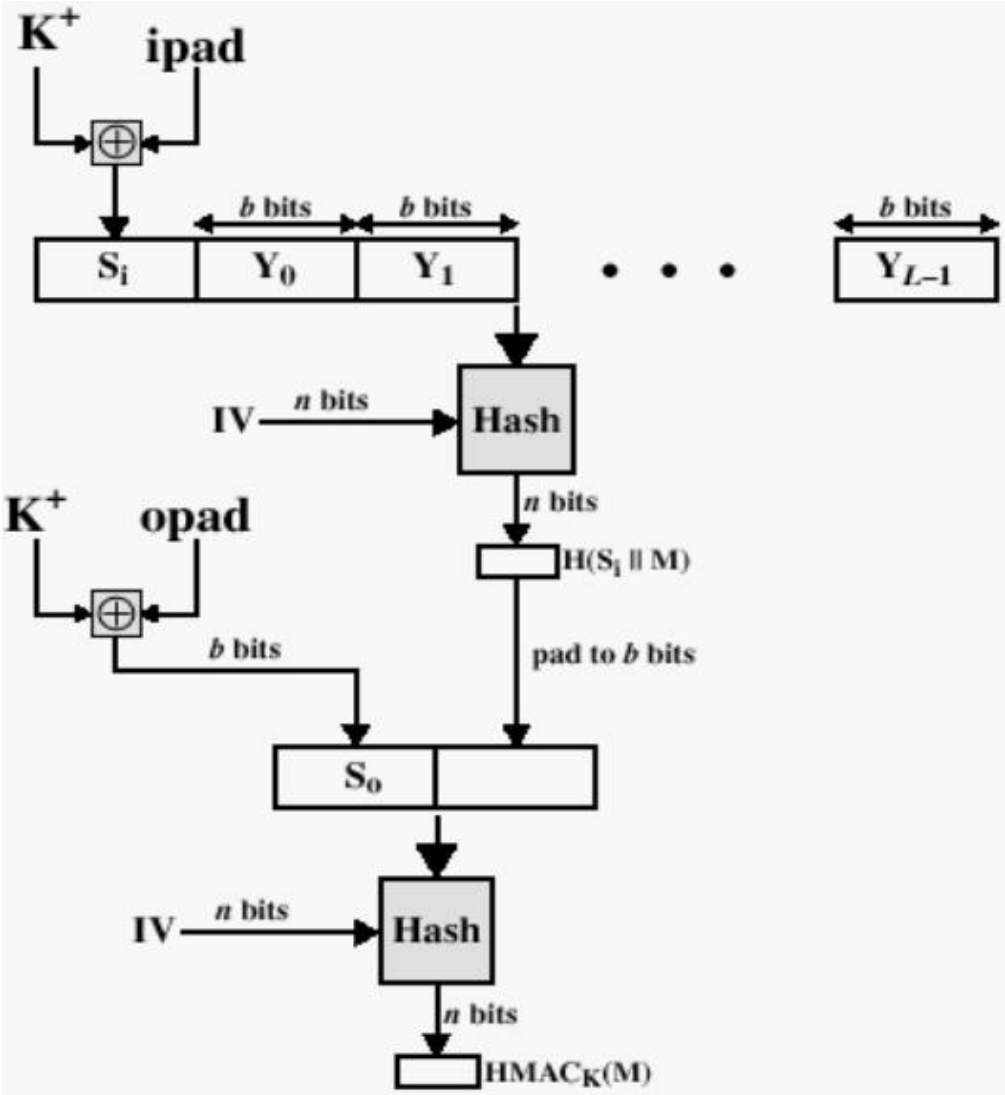
HMAC

- Neka je *h* bilo koja heš funkcija.
  - Poruka je bilo koje dužine a heš vrednost uvek ima dužinu *N*.
  - *N* zavisi od izbora *h*.
- Neka je *K* tajni ključ, koji se koristi za autentifikaciju, poznat obema stranama u komunikaciji.
  - Ključ ne treba da ima veću dužinu 64 bajta, tj. od dužine bloka heš funkcije.
  - Ako je ključ manje dužine, dopunjava se nulama.
- Dva fiksna niza dužine po 64 bajta:
  - ipad = 0x36 0x36.....0x36 (64 bajta)
  - opad= 0x5C 0x5C.....0x5C (64 bajta)

HMAC

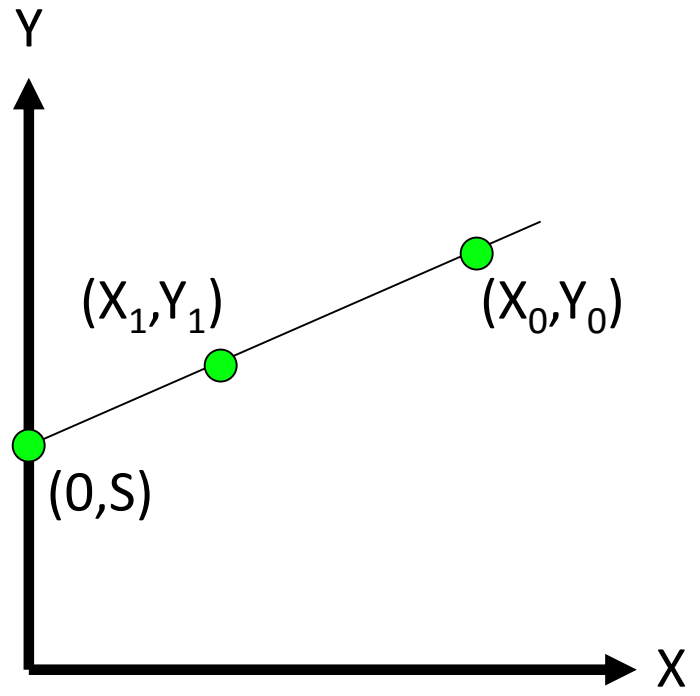
- Funkcija HMAC ima dva parametra: poruku *M* i ključ *K*.
- HMAC(*M*, *K*) = *h*(*K* ⊕ opad || *h*(*K* ⊕ *ipad* || *M*))
- Obajšnjenje algoritma:
  1. Po potrebi dopisati nule da bi *K* imao 64 bajta.
  2. Izračunati K ⊕ ipad.
  3. Na niz dobijen u koraku 2 dopisati poruku *M*.
  4. Izračunati heš vrednost niza dobijenog u koraku 3.
  5. Izračunati K ⊕ opad.
  6. Dopisati heš vrednost dobijenu u koraku 4 na 64 bajta dobijena u koraku 5.
  7. Izračunati heš vrednost niza dobijenog u koraku 6.

HMAC


K+
ipad
b bits
b bits
b bits
Si
Y0
Y1
YL–1
IV
n bits
Hash
K+
opad
n bits
H(Si || M)
b bits
pad to b bits
So
IV
n bits
Hash
n bits
HMACK(M)

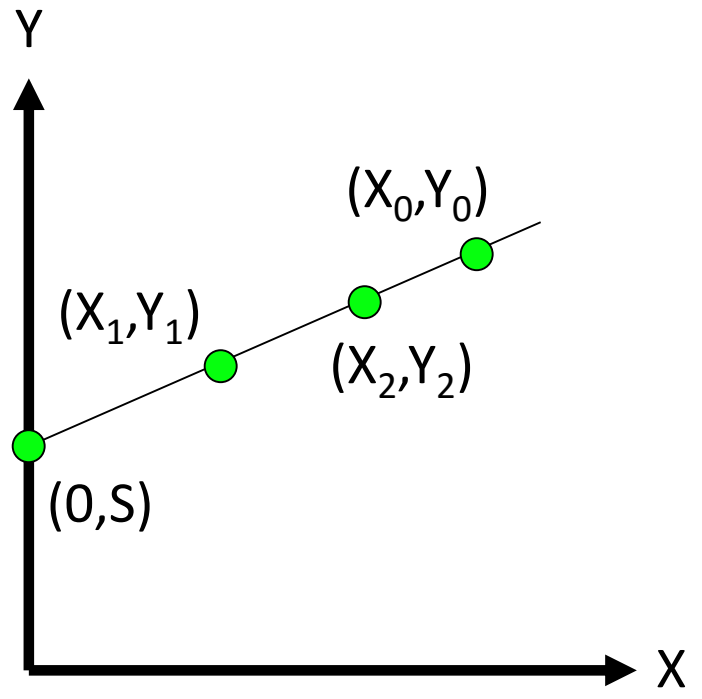
Deljenje tajni

- Shamir-ov sistem deljenja tajni, šema 2 od 2.
  - Dve tačke određuju liniju.
  - Dati (X_0, Y_0) Alisi.
  - Dati (X_1, Y_1) Bobu.
  - Bob i Alisa moraju da sarađuju kako bi došli do tajne S.



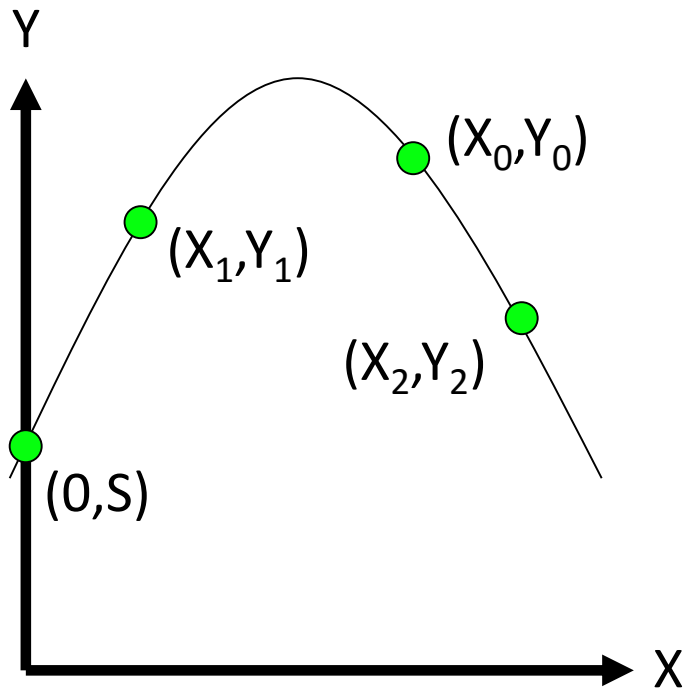
Deljenje tajni

- Shamir-ov sistem deljenja tajni, šema 2 od 3.
  - Dve tačke određuju liniju.
  - Dati (X_0, Y_0) Alisi.
  - Dati (X_1, Y_1) Bobu.
  - Dati (X_2, Y_2) Čarliju.
  - Bilo kojih dvoje od Alise, Boba i Čarlija može da sasrađuje kako bi došli do tajne *S*.
  - Niko do njih, sam, ne može da nađe tajnu *S*.



Deljenje tajni

- Shamir-ov sistem deljenja tajni, šema 2 od 3.
  - Tri tačke određuju parabolu.
  - Dati (X_0, Y_0) Alisi.
  - Dati (X_1, Y_1) Bobu.
  - Dati (X_2, Y_2) Čarliju.
  - 3 tačke određuju parabolu.
  - Alisa Bob i Čarli moraju da sarađuju kako bi došli do tajne *S*.
  - Da li možete da predložite šemu “3 od 4”?

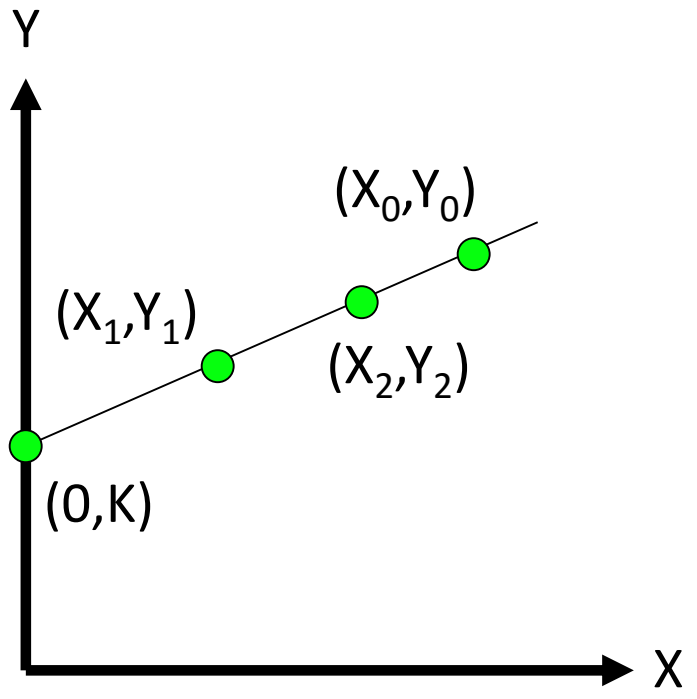


Deljenje tajni

- Primena: **deponovanje ključa** (*key escrow*).
  - Zahtev da tajni ključ mora biti negde deponovan.
  - Ključ se može upotrebiti na osnovu naloga suda.
    - Ali vlasnik ključa ne mora da ima apsolutno poverenje u neku instituciju koja će čuvati ključeve.
  - Može se koristiti deljenje tajni.
    - Na primer: tri zvanične agencije.
    - Dve moraju da sarađuje da bi se rekonstruisao ključ.
    - Samnjuje se verovatnoća zloupotrebe.

Deljenje tajni

- Neka je K vaš privatni ključ.
- Tačka (X0,Y0) – MUP.
- Tačka (X1,Y1) – ministarstvo pravde.
- Tačka (X2,Y2) – ministarstvo odbrane.
- Da bi se rekonstruisao ključ, dve od tri agencije moraju da sarađuju.
- Ni jedna od agencija ne moće sama da dođe do K.



Slučajni brojevi

- Slučajni brojevi služe za generisanje ključeva.
  - Simetrični ključevi
  - Diffie Hellman: tajna vrednost
  - ..
- Kriptografski slučajni brojevi moraju biti statistički slučajni i nepredvidivi.
  - Server generiše simetrične ključeve:
    - Alisa: K_A
    - Bob: K_B
    - Čarli: K_C
    - Dejv: K_D.
  - Pretpostavićemo da Alisa, Bob i Čarli ne vole Dejva.
  - Alisa, Bob i Čarli radeći zajedno ne bi smeli da otkriju K_D.

Primer loših slučajnih generatora

- Online verzija Texas Hold ‘em pokera
  - ASF Software, Inc.
- Slučajni brojevi se koriste za mešanje karata.
- Program nije proizvodio dovoljno slučajna mešanja karata.
- Bili su u satnju da razbiju ovaj sistem tako da predvide mešanje karata i **nekoliko nedelja unapred**!

Mešanje karata

- Ima $52! > 2^{225}$ mogućih mešanja.
- Poker program koristi “slučajan” 32-bitni ceo broj kojim je određeno mešanje.
  - Stoga se može desiti samo 2^{32} različitih mešanja.
- U tu svrhu je korišćen Pascal *pseudo-random number generator* (PRNG): Randomize().
- Vrednost *seed* za PRNG je funkcija od broja milisekundi brojano od ponoći.
  - Ima manje od 2^{27} milisekundi u danu.
    - Stoga ima manje od 2^{27} mogućih mešanja.
  - *Seed* se zasniva na milisekundama brojano od ponoći.
  - PRNG se puni novim početnim stanjem prilikom svakog mešanja.
    - Sinhronizacijom sata sa serverom broj mešanja koje treba testirati je $< 2^{18}$.
    - Mogu se oprobati svih 2^{18} u realnom remenu.
      - Testirati sva moguća mešanja za svaku kartu koja se vidi na talonu.
    - Napadač unapred zna svaku kartu koja će pojaviti, nakon prvih pet rundi igre!

Prediktabilost i pseudo-tajnost

- Svi **PRNG su prediktibilni**.
  - Samo je pitanje koliko izlaznog niza moramo poznavati da bi smo otkrili celokupni mehanizam generisanja.
- **Kriptološki slučajni nizovi nisu jednostavno prediktibilni**!
- Pitanje generisanja “seed-a” (ili unutrašnjih ključeva) ostaje problem!
  - Kako generisati početne slučajne vrednosti?
- Izvori slučajnosti u okviru softvera.
  - Softver je deterministički sistem.
    - Stoga se mora osloniti na eksterne “slučajne” dogadjaje: kretanje miša, mrežna aktivnost, itd.
    - Kvalitetan slučajan niz se može dobiti softverski, ali je količina ovakvih nizova ograničena.
  - Posledica: “korišćenje pseudo-slučajnih procesa u generisanju tajnih vrednosti rezultuje u pseudo-tajnosti”.

Literatura

1. M. Stamp: *Information Security*. John Wiley and Sons.
2. M. Veinović, S. Adamović: Kriptologija 1. Univerzitet Singidunum, Beograd. *

* Može se besplatno preuzeti sa portala: www.singipedia.com

Hvala na pažnji

Pitanja su dobrodošla.