AUTENTIFIKACIJA

Autentifikacija poruka

- Za informaciju se kaže da je autentična ako je:
  - Nepromenjena
  - Zaista potiče od navodnog izvora

Autentifikacija poruka moguća je:

- Pomoću konvencionalnog šifrovanja
- Bez šifrovanja
  - Kod za autentifikaciju poruke
  - Heš funkcija za jednosmerno šifrovanje

Autentifikacija pomoću konvencionalnog šifrovanja

- Simetrično šifrovanje nije pogodno za autentifikaciju – u nekim režimima šifrovanja moguće je da napadač promeni redosled blokova šifrata
- Svaki blok će se uspešno dešifrovati, ali promena redosleda može da promeni značenje celokupne informacije

Autentifikacija bez šifrovanja poruke

- Najčešće se autentifikacija obezbeđuje zasebno od šifrovanja poruka

Kod za autentifikaciju poruke

- Pomoću tajnog ključa generiše se mali blok podataka koji se zove kod za autentifikaciju poruke (MAC-message authentication cod)
- Poruka i kod se prenose
- Pošiljalac i primalac znaju tajni ključ
- Primalac iz primljene poruke izračunava kod pomoću tajnog ključa i poredi izračunati kod sa primljenim kodom
- Uobičajeno kod sadrži 16 ili 32 bita

Autentifikacija bez šifrovanja poruke

Heš funkcija za jednosmerno šifrovanje

- Od poruke promenljive dužine M, pravi izvod poruke H(M), bez korišćenja tajnog ključa
- Izvod poruke (heš) može se šifrovati tajnim ili javnim ključem, a može se i pri pravljenju izvoda koristiti tajna vrednost koja se ne prenosi već se doda na izlazu zbog izračunavanja kontrolnog izvoda koji se poredi sa prenetim izvodom
- Ako se izvod (heš) šifruje, na prijemu se porede heš koji se dobija računanjem iz pristigle poruke i heš koji je primljen i dešifrovan

Heš funkcija za jednosmerno šifrovanje

- Kada se koristi javni ključ:
  - Osim autentifikacije dobija se i digitalni potpis
  - Nema distribucije ključeva

Elektronsko potpisivanje

- E-mail
- E- trgovina
- Finansijske transakcije

- Provera identiteta pošiljaoca
- Integritet poruke
- Neporecivost

Bezbedne Hash funkcije

- Šifarski sistemi sa javnim ključevima, kao i sistemi za digitalni potpis mogu biti veoma spori.
- Takođe, u nekim slučajevima, dužina digitalnog potpisa može biti veća ili jednaka dužini same poruke koja se potpisuje.
- Da bi se rešili ovi problemi koriste se hash funkcije.

- Hash funkcija je izračunljiva funkcija koja primenjena na poruku m promenljive dužine daje njen rezime (otisak) fiksne dužine koja se naziva njenom hash vrednošću - $H(m)$
- Hash funkcije se definišu na sledeći način:

$$H : M \rightarrow M \qquad H(m) = h$$

- U opštem slučaju, $H(m)$ je mnogo manjih dimenzija od *m*. Na primer, *m* može da ima dužinu od jednog megabajta, dok $H(m)$ može imati svega 128 ili 160 bita.

- Upotrebom hash funkcija, problem dužine poruke ili digitalnog potpisa se rešava tako što se umesto da se šifruje ili digitalno potpisuje cela poruka m, potpisuje se ili šifruje samo rezime poruke – $H(m)$
- Hash funkcije koje su se najviše koristile u kriptografske svrhe su MD2, MD4 i MD5 (Message Digest), koje je predložio Rivest. Ove funkcije daju rezimee dužine 128 bita.
- Pored njih, bila je popularna i SHA-1 (Secure Hash Algorithm) funkcija (NIST) koja daje otisak poruke dužine 160 bita.
- Sve navedene funkcije su razbijene od strane kriptoanalitičara

MD5 algoritam

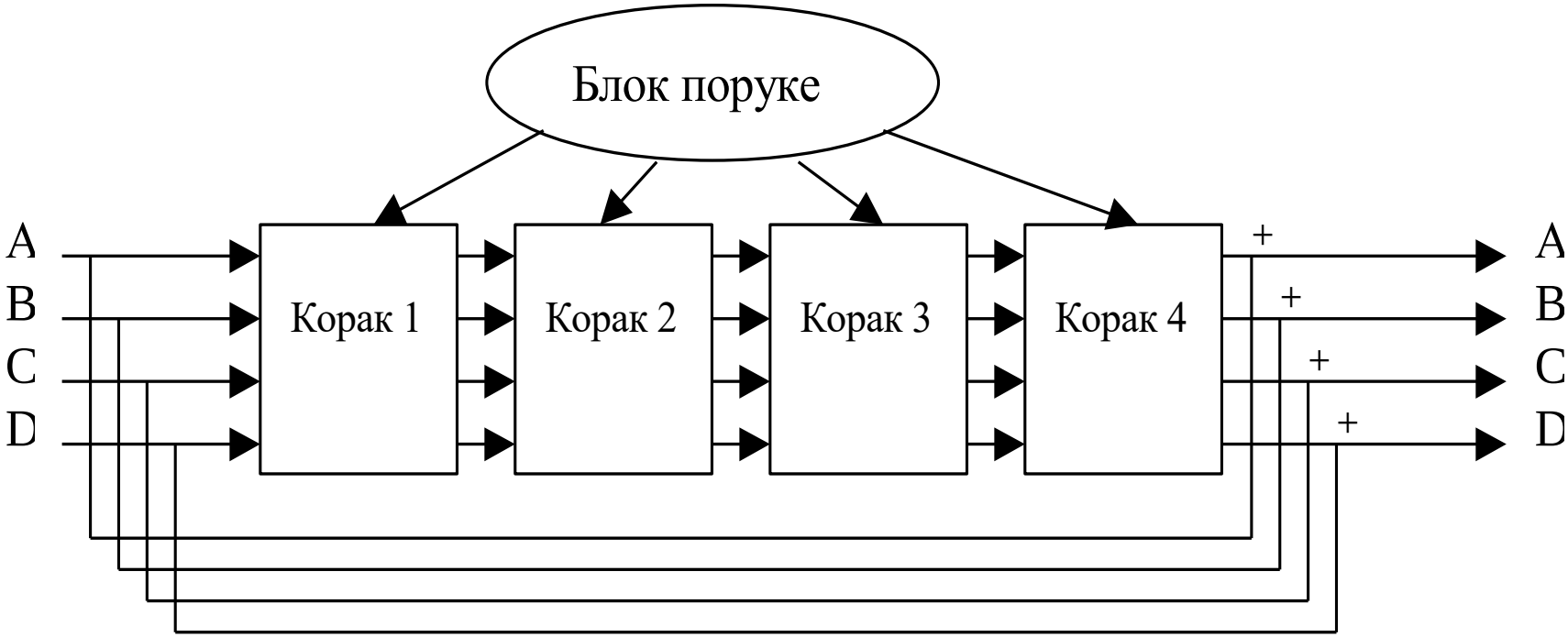
- Proizvodi 128-bitnu heš vrednost poruke

- Ulazna poruka se deli na blokove od 512 bita
- Blokovi se dele u podblokove po 32 bita (16)
- Sažetak poruke se postavi na 4 podbloka dužine po 32 bita i formira se 128-bitna heš vrednost

MD5 algoritam

- Dodavanje bita za dopunjavanje
  - Poruka se proširuje tako da bude tačno za 64 bita kraća od celobrojnog multipla 512 bita – prvo se na kraj poruke doda jedan bit jedinice, a onda se doda potreban broj nula
- Dopunjavanje dužine
  - 64 bita koji predstavljaju dužinu poruke, dodaju se na kraju
- Inicijalizacija MD bafera
  - Inicijalizuju se 4 promenljive A, B, C i D dužine po 32 bita
  - AA, BB, CC i DD
- Definisanje 4 nelinearne funkcije
  - F, G, H i I - ulaz su 4 32-bitne reči, izlaz takođe
- Transformacija za 4 kruga
  - Glavna petlja algoritma izvršava se za sve blokove poruke u 4 kruga (faze)
- Proračun 4 kruga (64 koraka)
  - Svaki krug ima različite operacije - FF, GG, HH i II, 16 puta
- Rezultat su 4 bloka po 32 bita

MD5 algoritam



MD5 algoritam**(detaljnije)**

- Četiri inicijalne promenljive se kopiraju u promenljive *a*, *b*, *c* i *d*.
- Glavna petlja se sastoji od 4 faze koje su veoma slične. Svaka faza koristi različitu operaciju 16 puta, koja se sastoji od primene određene nelinearne funkcije nad tri od četiri promenljive *a*, *b*, *c* ili *d*. Zatim se tako dobijeni rezultat dodaje četvrtoj promenljivoj, podbloku poruke i jednoj konstanti. Dobijeni rezultat se rotira ulevo promenljivi broj bita i dodaje se jednoj od četiri promenljive *a*, *b*, *c* ili *d*.
- Na kraju rezultat zamenjuje jednu od promenljivih *a*, *b*, *c* ili *d*.

MD5 algoritam**(detaljnije)**

- Postoje četiri nelinearne funkcije, po jedna se koristi u svakoj operaciji (različita u svakoj fazi):

  $F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$

  $G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$

  $H(X,Y,Z) = X \oplus Y \oplus Z$

  $I(X,Y,Z) = Y \oplus (X \vee (\neg Z))$

- gde navedeni funkcijski znaci predstavljaju ($\oplus$ - *XOR* funkcija, ^ - *AND* funkcija, v - *OR* funkcija, $\neg$ - *NOT* funkcija).

MD5 algoritam**(detaljnije)**

Ako Mj predstavlja j-ti podblok poruke, j =0, …, 15, a <<< s predstavlja funkciju cirkularnog šiftovanja za s bita, tada se pomenute četiri operacije mogu predstaviti na sledeći način:

$FF(a,b,c,d,M_j,s,t_i)$ označava: $a = b + ((a + F(b,c,d) + M_j + t_i) <<< s)$

$GG(a,b,c,d,M_j,s,t_i)$ označava: $a = b + ((a + G(b,c,d) + M_j + t_i) <<< s)$

$HH(a,b,c,d,M_j,s,t_i)$ označava: $a = b + ((a + H(b,c,d) + M_j + t_i) <<< s)$

$II(a,b,c,d,M_j,s,t_i)$ označava: $a = b + ((a + I(b,c,d) + M_j + t_i) <<< s)$

Nakon prethodno opisanog postupka, *a*, *b*, *c* i *d* se dodaju na *A*, *B*, *C* i *D*, respektivno, i algoritam nastavlja sa narednim blokom podataka. Krajnji rezultat se formira konkatenacijom od dobijenih *A*, *B*, *C* i *D*.

SHA-1 algoritam

- SHA-1 algoritam takođe procesira ulaznu poruku u blokovima od 512 bita, podeljenim u 16 podblokova dužine 32 bita.
- Prvo se poruka proširuje na isti način kao i u MD5 algoritmu, tako da se dobije poruka koja je po dužini tačno 64 bita kraća od odgovarajućeg multipla od 512 bita. Naime, prvo se na kraj poruke doda jedan bit jedinice, praćen zahtevanim brojem nula. Zatim se 64-bitna reprezentacija dužine poruke priključi rezultatu.
- Izlaz algoritma predstavlja skup od 5 32-bitna bloka, spojena tako da jednoznačno formiraju 160-bitnu hash vrednost.

SHA-1 algoritam (nastavak)

Algoritam se sastoji od sledećih koraka:

1. Prvo se poruka obradi tako da je njena dužina tačno multipl od 512 bita
2. Zatim se inicijalizuju 5 32-bitne promenljive
3. Zatim počinje glavna petlja algoritma koja se izvršava za sve blokove dužine 512 bita date poruke.

SHA-1 algoritam (nastavak)

- 5 inicijalnih promenljivih se kopira u promenljive *a*, *b*, *c*, *d* i *e*.
- Glavna petlja se sastoji od 4 faze koje su veoma slične. Svaka faza koristi različitu operaciju 20 puta, koja se sastoji od primene određene nelinearne funkcije nad tri od pet promenljivih *a*, *b*, *c*, *d* ili *e*. Zatim se tako dobijeni rezultat procesira slično kao i u MD5 algoritmu.

Dopuniti poruku 1010 1000 1110 1111 0011 0010 1101 1111 0110 0101 do dužine 512 bita, kao što je definisano u SHA-1 algoritmu za dobijanje sažetka poruke?

Ova poruka ima dužinu 40 bita tj L=40 u heksadecimalnom sistemu to je 28 (tj. 0000 0000 0000 0028)

1010 1000 (A8)

1110 1111 (EF)

0011 0010 (32)

1101 1111 (DF)

0110 0101 (65)

1000 0000 (80)

0 (Nulama se popunjava do 512 bita)

.

.

0

0

0000 0000

0000 0028 (40 u heksadecimaln sistemu)

SHA - 2

- Ovo je revidirana verzija standarda SHA-1
- Definisane su tri nove SHA-256, SHA-384 i SHA-512 sa dužinama heša 256, 384 i 512 bita
- Koriste se blokovi od po 1024 bita
- Dodaje se 128 bita koji sadrže dužinu originalne poruke
- Izlaz je 512 bita za haš vrednost
- Svaki bit heš koda je funkcija svakog ulaznog bita
- SHA 3 zvanično objavljen 2012. god

Kodovi za autentifikaciju poruka

Postoji značajno zanimanje za razvoj MAC-a izvedenog iz heš koda poput SHA-1 iz bar dva razloga:

- Heš funkcije se uglavnom izvršavaju brže od konvencionalnih algoritama za šifrovanje
- Kod za heš funkcije je svuda dostupan

Heš funkcija ne može da se direktno koristi kao MAC jer se ne oslanja na tajni ključ, ali moguća je ugradnja tajnog ključa u postojeći heš algoritam

HMAC

- Koristi se u raznim Internet protokolima:
  - IP security
  - TLS (Transport Layer Security)
  - SET (Secure Electronic Transaction)
- Koristi raspoložive heš funkcije bez modifikacija (kao modul)
- Ugrađenu heš funkciju je lako zameniti bržom ili bezbednijom, ako je potrebno
- Ključ dopunjen potrebnim brojem nula XOR-uje se sa nekim definisanim nizom bita, pa se to provlači kroz heš funkciju i generiše se ključ

Шифарски системи

- Šifarski sistem tajnim ključem je familija parova funkcija (E_k, D_k) za svaki ključ *k* iz skupa ključeva *K*, definisana na sledeći način:

$$E_k : M \to X$$

$$D_k : X \to M$$

- M i X su skupovi otvorenih tekstova i šifrata, respektivno
- Za svaki otvoreni tekst m iz M važi:

$$D_k\left(E_k\left(m\right)\right) = m$$

- Da bi se koristio ovakav sistem, korisnici A i B se dogovore da uzmu tajni ključ *k* iz *K*. Ako A želi da pošalje poruku *m* iz *M* korisniku B, šifruje je pomoću funkcije *Ek*, $E_k(m) = c$

  i rezultat *c* se šalje korisniku B.
- Da bi rekonstruisao originalnu poruku, B dešifruje primljeni šifrat *c* pomoću funkcije D_k,

$$D_k(c) = D_k(E_k(m)) = m$$

- U kriptografiji se smatra “lakim" proračun koji se može izvršiti u kratkom vremenu.
- Za probleme koji se ne mogu rešiti u prihvatljivom vremenskom periodu, koristeći najbolji poznati algoritam i najbolju raspoloživu tehnologiju koristi se termin “teški" ili “intraktabilni".
- Parovi funkcija (E_k, D_k) moraju biti “laki" za izračunavanje za korisnike i morali bi biti “teški" za izračunavanje za kriptoanalitičara koji poznaje samo *c*, tako da ne može da rekonstruiše ni *m* ni *k*.

Problemi u kriptografiji sa tajnim ključevima

- Distribucija ključeva – dva korisnika moraju da izaberu tajni ključ pre početka komunikacije i da za njegovo prenošenje koriste siguran kanal. Ovakav siguran kanal nije uvek na raspolaganju.

Problemi u kriptografiji sa tajnim ključevima

- **Manipulacija ključevima** – U mreži sa n korisnika, svaki par korisnika mora da ima svoj sopstveni tajni ključ, što čini ukupno $n(n-1)/2$ ključeva za tu mrežu.
- **Nemogućnost realizacije procedure digitalnog potpisa**– U šifarskim sistemima sa tajnim ključevima nema mogućnosti, u opštem slučaju, za digitalno potpisivanje poruka, tako da onaj koji prima poruku ne može da bude siguran da je onaj koji mu je poslao poruku zaista njen autor.