IV - Zaštita informacija i podataka

S A D R Ž A J

4.1-Savremeni algoritmi kriptovanja

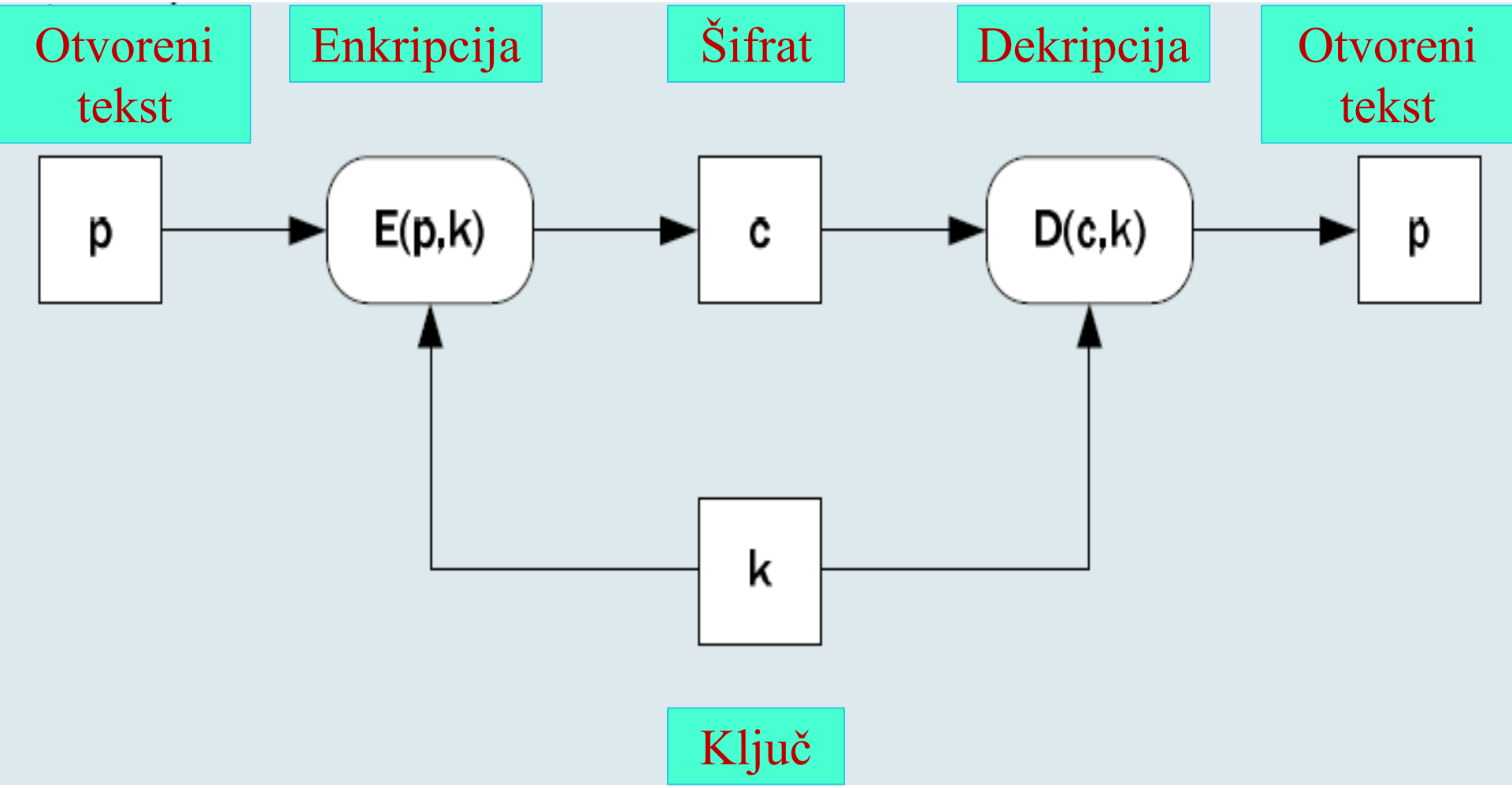
- Danas je život bez kriptografije teško zamisliv.
- Kriptografija se koristi u svakodnevnim radnjama:
  - elektronska pošta,
  - povezivanju na internet stranice,
  - elektronsko poslovanje,
  - SIM kartice,
  - zaštita od kopiranja,
  - autorska prava pesama ili knjiga i td.
- Pojava računara uslovila je razvoj novih algoritama kriptografije koji su svoje principe zasnivali na mogučnosti računara
- Metode klasične kriptografije zasnivaju se na tajnom pisanju, odnosno matematičkim metodama/algoritmima kako bi se neka poruka šifrirala
- Metode moderne kriptografije zasnivaju se na tajnosti ključa.
- U modernoj kriptografiji važnija je **tajnost ključa** od tajnosti metode
- **Dve metode** definišu kako funkcioniše ključ u kriptosistemu:

1. **Simetrični algoritmi** – koriste jedan odnosno privatni ključ
2. **Asimetrični algoritmi** - koriste dva različita (javni i privatni) ključ.

4.2-Simetrični algoritmi kriptovanja

- Simetrični algoritmi kod enkripcije i dekripcije koriste isti ključ.
- Sistem kriptovanja se bazira na tajnosti upotrebljenog ključa.
- Često se simetrični sistemi u kriptografiji nazivaju: sistemi sa deljenim ključem, sistemi sa privatnim ključem i sistemi sa jednim ključem.
- Ranije se mislilio da kada imamo dve funkcije, jednu za enkripciju i jednu za dekripciju da je siguran način prenosa poruke jedino ako način šifriranja i dešifriranja držimo u tajnosti.
- To je važilo sve dok Kerckhoff 1883 g. nije promenio način šifriranja.
- Za princip Kerckhoff-og šifriranja važi da je sistem u kriptografiji siguran i u slučaju kada su uljezu poznati svi detalji šifriranja gde se misli na algoritam enkripcije i dekripcije, osim **kriptografskog ključa**
- Simetrični sistem u kriptografiji se koristi i za **autentifikaciju**.
- Najpoznatiji primer takve tehnike su kodovi autentifikacije porukom (***message authentification codes***, ili skraćeno **MAC**).
- Problem je što se i sam ključ prenosi, pa postoji opasnost da se otkrije.
- Potreban je dogovor između pošiljaoca i primaoca poruke o ključu
- Simetrični sistemi imaju puno veću brzinu prenosa od asimetričnih

4.2-Simetrični algoritmi kriptovanja



Kriptosistem se definiše:

- **P** - skup poruka,
- **C** - skup šifrata,
- **K** - skup ključeva,
- **E(P,K)→C** - funkcija šifrovanja,
- **D(C,K)→P** - funkcija šifrovanja.

4.2-Simetrični algoritmi kriptovanja

- Koristi se isti ključ za šifrovanje i za dešifrovanje podataka
- Šifriranje simetričnim ključem ***E*** na osnovu ključa ***k*** i ulaznih podataka ***p*** proizvodi šifrat **c=E(p,k)**
- Funkcija dešifrovanja ***D*** na osnovu istog ključa ***k*** i šifrata ***c*** proizvodi orginalnu poruku **p=D(c,k)**
- Simetrični algoritmi su brzi pa se mogu koristiti za šifrovanje većih datoteka ili implementaciju u kriptosistema datoteka.
- Simetrične kriptografske algoritme delimo na:
  1. Algoritme koji kriptuju **tokove podataka**
  2. Algoritme koji kriptuju **blokove podataka**
  3. **Kodovi autentifikacije poruke** (**MAC**)
- Najpoznatiji simetrični kriptoalgoritmi su:
  1. **DES** (*Data Encryption Standard*),
  2. **AES** (*Advanced Encryption Standard*),
  3. IDEA (*International Data Encryption Algorithm*),
  4. Blowfish, Twofish i drugi.

4.2.1 - Šifre toka podataka

- **Šifre toka** (***Stream Ciphers***) šifriraju svaki bit posebno tj. bit po bit.
- Za ključ ove šifre se koristi niz koji se sastoji od niza bitova.
- Prikaz osnovnih funkcija u **šifri toka**:

  Enkripcija: $yi = e\ xi = xi + si \bmod 2$

  Dekripcija: $xi = e\ yi = yi + si \bmod 2$
- Prvo što možemo zaključiti da su funkcije enkripcije i dekripcije iste.
- U dosadašnjim primerima kriptografije uvek je postojala razlika jer se u dekripciji koristio suprotni operator od enkripcije.
- Prikaz jednakosti funkcija možemo prikazati na primeru f-je dekripcije

$$d\ yi = yi + si \bmod 2$$
$$d\ yi = (xi + si) + si \bmod 2$$
$$d\ yi = xi + 2si \bmod 2$$
$$d\ yi = xi$$

Objašnjenje: yi se u prvom koraku zameni metodom supstitucije odgovarajučem ekvivalentu iz funkcije enkripcije. Uvek će nam ispasti da imamo 2 nizovna ključa odnosno $2si$ što po modulu2 rezultira 0 i tako nam **funkcija dekripcije** ispada **jednaka** kao **funkcija enkripcije**.

4.2.1 - Šifre toka podataka

- Primenu šifre toka možemo objasniti u telefonskom pozivu sa jednog mobilnog uređaja ka drugom.
- Poziv se ostvaruje tako da se mobilni uređaj prvo spoji preko vlastitog kanala na najbližu centralu mrežnog provajdera.
- Niko nema pristup tom kanalu osim osobe koja vrši pozivanje.
- Kada se komunikacija između centrale uspostavi vrši se enkripcija nizom ključeva (s_i),tako da se glas pozivaoca kvantizira po vremenskoj amplitudi, na taj način da se ostvari kanal između njega i osobe koju je pozvao.

Primer: Prikaz komunikacije gde se koriste šifre toka (znak $\oplus$ = mod2)

$$e_k(x)=y \xrightarrow{\text{y}} d_k(\text{y}) = x$$

$$\ldots x1,x2,x3 \oplus \ldots s1,s2,s3 \xrightarrow{y_i} \oplus \ldots x1,x2,x3 \ldots s1,s2,s3$$

x_i	Si	y_i	Si	xi
1	1	0	1	1
0	1	1	1	0
0	0	0	0	0
0	1	1	1	0
0	0	0	0	0
0	0	0	0	0
1	1	0	1	1

4.2.1 - Šifre toka podataka

- Jačina šifre direktno zavisi od **niza ključeva** koji koristimo.
- Da bi se sprečilo pronalaženje niza ključa napadom prisilne metode koristi se **generator slučajnih brojeva** (*random generator number)*
- Sve generatore slučajnih brojeva delimo na **3 tipa**:

1. **Primenjeni generator** (*True random number generator*)-svrstavamo sve slučajne procese iz života: bacanje novčiča, lutrija, rulet i td.
2. **Generator prividno** (*Pseudo random number generator*) – ovaj generator slučajnih brojeva nije slučajan već je određen u stvarnosti. Slučajni niz bitova je dobijen izračunavanjem što znači da je određen i da neko ponovnim računanjem može dobiti potpuno isti niz bitova. U šifri toka ovi generatori su imali razvijenu složenost i gledano sa matematičke strane najbolje su rešavali problem niza ključeva, ali nedostatak je što nisu imali svojstvo nepredvidljivosti.
3. **Kriptografski generator** (*Cryptograhycally Secure pseudo random number generator*) – ovi generatori su u stvarnosti generatori slučajno prividnih brojeva sa dodatnim svojstvom nepredvidljivosti.

4.2.2 –Šifre bloka podataka

- Blokovske šifre (*Block Ciphers*) šifriraju deo otvorenog teksta po **blokovima** tj. manjim delovima otvorenog teksta, po određenoj dužini.
- Najpoznatiji blokovski algoritmi su **DES i AES algoritmi**
- Kod DES algoritama blokovi su dužine 64 bita, a kod AES 128 bita.
- Ovi algoritmi koriste matematičku metodu permutacije, pa ćemo u sledečem delu prikazati princip rada ovih algoritama.

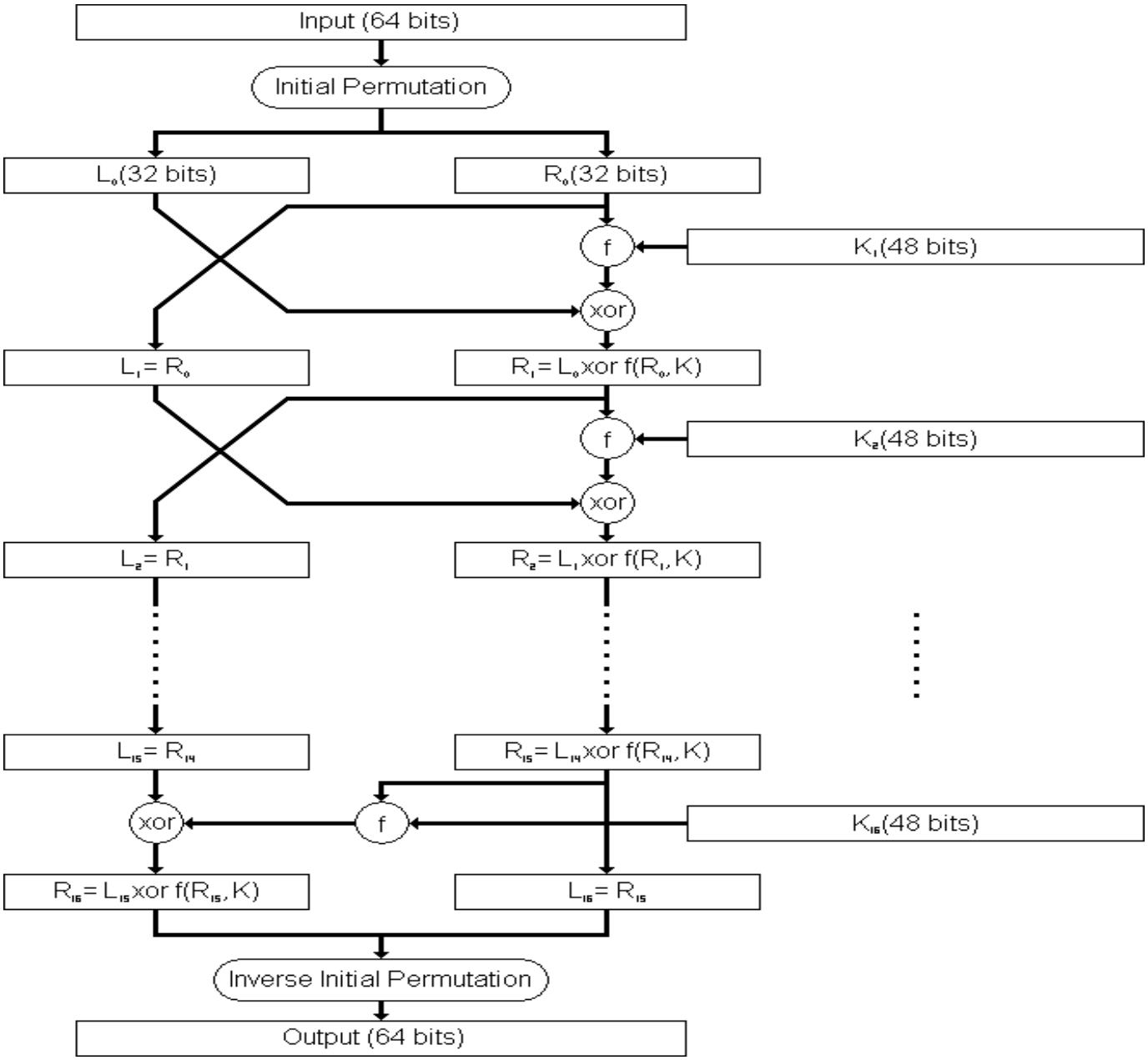
DES algoritam

- DES (*Data encryption standard*) algoritam je među najraširenijim i najpoznatijim algoritmima simetričnog sistema kriptovanja
- Spada u moderne i novije sisteme jer se koristi i danas.
- Krajem 60-tih prošlog veka dolazi do naglog razvoja finansijskih transakcija, pa i potreba za zaštitom tih sadržaja kriptografijom.
- Tokom 70-tih sistem kriptovanja razvija **IBM** a kasnije to dorađuje Agencija za nacionalnu sigurnost, NSA (*National Security Agency*).
- 1976 se standardizuje pod nazivom ***Data encryption standard***

4.2.2 – DES algoritam

- DES šifrira ulaznu poruku blokom dužine **64** bita, na izlazu dobijemo blok šifrata dužine **64b**, dok je ključ ***k*** koji koristimo blok dužine **56b**.
- Ključ se često pojavljuje u bloku dužine 64 bita, tada se svaki osmi bit zanemaruje, odnosno otpada i mi ga koristimo za proveru partiteta.
- DES je simetričan algoritam koji šifruje tekst u blokovima dužine, koristeći ključ **k** dužine **56 bitova**.
- Tako se dobija šifrat dužine **64 bita** (56 + 8 bita za paritet).
- Tri osnovna koraka u algoritmu su:
  1. **Inicijalna permutacija IP**,
  2. **16 rundi** obrade podataka (proširenje,XOR sa ključem, supstitucija)
  3. **Završna inverzna permutacija**.
- Sigurnost DES algoritma je rizična zbog dužine ključa.
- Zbog strukture algoritma i dužine ključa smatramo da ovaj algoritam ne spada u **kompletno sigurne sisteme** u kriptografiji, ali sa malim izmenama se može smatrati dovoljno sigurnim.
- Koristi se trostruki DES koji povećava sigurnost (ključ dužine 168 b).

4.2.2 – Algoritam DES



4.2.2 – AES algoritam

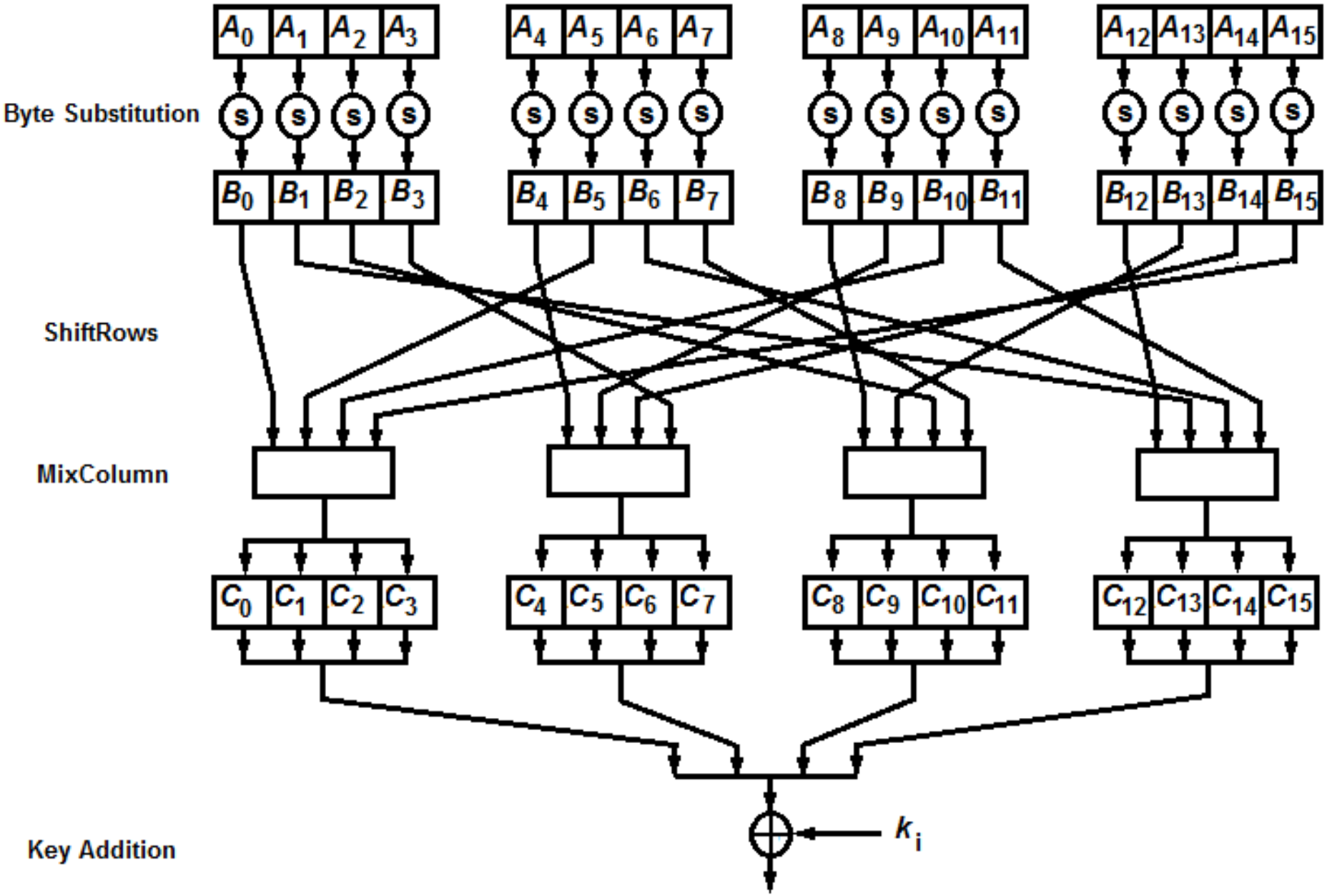
- AES (*Advanced encryption standard*) odnosno napredni kriptografski standard spada u simetrične algoritme.
- Nedostatak DES algoritma, dužina ključa, ovde je sada izmenjena.
- Ovde dužina blokova otvorenog teksta i šifrata iznosi **128 b**. dok dužina bloka ključa može biti 128, 192, ili 256 bita.
- AES algoritam je najpopularniji i najkorišćeniji simetrični algoritam.
- Iza funkcionalnosti AES algoritma stoji niz pravila i matemat.metoda koje se koriste a to je primena modularne aritmetike u kriptografiji
- U AES algoritmu takođe koristimo određene skupove.
- Osnovne algebarske sktukture možemo podeliti na sledeći način:
  1. grupa – koristimo operacije + , −
  2. prsten – koristimo operacije + , − , ·
  3. **polja – koristimo operacije + , − , · , /**
- AES algoritam koristi polja što znači da su na raspolaganju sve računske operacije.
- Za postizanje navedenih izmena kod DES-a bili su potrebni polinomi.

4.2.2 – AES algoritam

- Kao i u DES algoritmu imamo određeni broj koraka u algoritmu, ali ovde on zavisi od dužine bloka ključa tako da je:
  1. 128 bita – 10 koraka
  2. 192 bita – 12 koraka
  3. 256 bita – 14 koraka
- Unutar AES algortima sve operacije se vrše na dvodimenzionalnom nizu okteta, odnosno matricama.
- Enkripcija i dekripcija se izvode tako što se ulazni blok podataka kopira u matricu stanja nad kojom se izvode razne operacije.
- Matrica stanja se transformiše 10,12 ili 14 puta, zavisi od dužine ključa
- Svaki od navedenih dužina ključa odnosno okteta ima svoje korake, a svaki od koraka predstavlja funkciju koja sadrži četiri transformacije:

1. **zamena bloka** na bazi suptistucijske tablice
2. **šiftovanje redova** u matrici stanja
3. **mešanje podataka** unutar svake kolone matrice stanja,
4. **dodavanje podključa** u matricu stanja.

4.2.2 – AES algoritam



4.2 – Algoritam AES

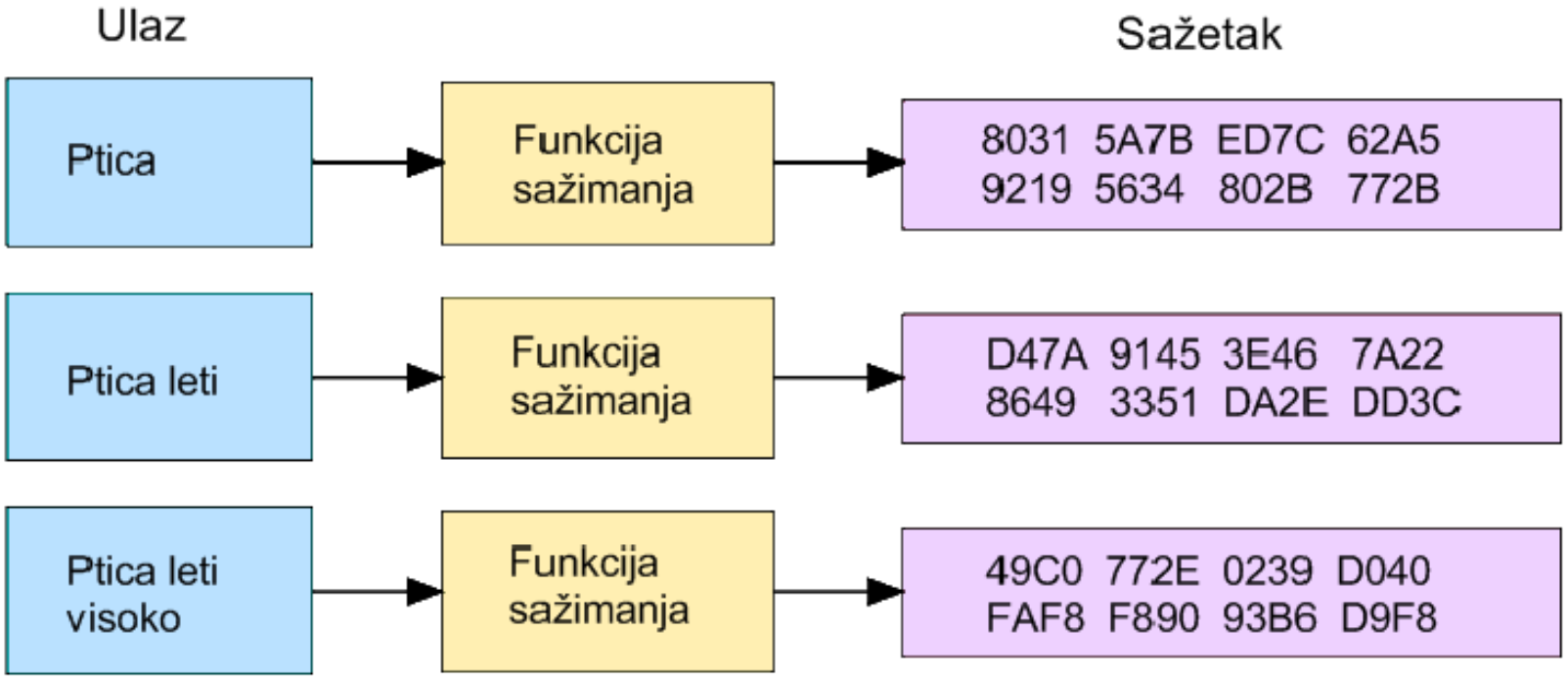
- Način na koji se AES razvio fascinira još dan danas.
- Tražio se naslednik DES-a zbog nedostatka dužine ključa, te je napravljen javi konkurs kako bi se napravio novi kriptografski sustav.
- Pobednik konkursa bili su Joan Daemen i Vincent Rijmen, a ime algoritma je složeno od početna 3 slova njihovih prezimena-**Rijendael**.
- U praksi se može reči da današnji AES nije isti kao **Rijendael** algoritam ali je nasledio strukturu i većinu osobina ovog algoritma
- Danas AES koristimo u svakom pretraživaču, elektronskoj pošti, raznim softverima koji koriste enkripciju kao štu su BitLocker, WinRAR, WinZip, Windows Office 2007, te sigurnost bežičnih mreža standarda 802.11 ili WPA2.
- Teoretski su pronađene metode za razbijanje AES enkripcije primenom grube sile, ali trenutno takve napade još nije moguće izvesti.
- Dobra strana AES algoritma što uvek postoji produženje dužine ključa, što bi otežalo još više kriptoanalizu kao i vreme potrebno za razbijanje ovog algoritma.

4.3 – Heš funkcije

- Hash funkcija(*hash*) predstavlja jednosmernu funkciju koja pretvara ulazni podatak promenljive dužine u izlazni podatak fiksne dužine Jednosmerna funkcija jeste funkcija oblika y=f(x) takva za koju važi:
  1. za dato x, f(x) se određuje relativno lako i efikasno,
  2. za dato y=f(x), $x=f^{-1}(y)$ određuje se relativno teško.
- Heš funkcije se dele na:
  1. **Jednoparametarske** (ulazni argument je samo poruka),
  2. **Dvoparametarske** (ulazni argumenti su poruka i ključ).
- U praksi se pojavila i druga podela na:
  1. Mehanizme za **uočavanje promena** (proveru integriteta poruke),
  2. Mehanizme za **proveru identiteta** poruka (CHAP).
- Kolizija je pojava kada dve različite ulazne poruke (ili više njih) rezultiraju istim izlazom.
- To je veliki problem ukoliko se heš funkcije koriste u okviru mehanizma provere identiteta.
- Značajne heš funkcije su algoritmi ***MD2, MD4, MD5 128 bit, SHA i RIPEMD 160 bit.***

4.3 - Karakteristike Heš funkcija

1. Računaju fiksne sažetke iz ulaznog niza podataka proizvoljne dužine.
2. Ireverzibilne su, iz sažetka se ne može izračunati izvorni niz podataka.
3. Postoji mogućnost sukoba (*collision*)–zbog fiksne dužine sažetka, dva različita ulazna niza podataka mogu rezultovati istim vrednostima sažetka
4. Kako bi se izbegli predvidivi sukobi, podaci koji se malo razlikuju rezultiraju potpuno različitim vrednostima sažetka.



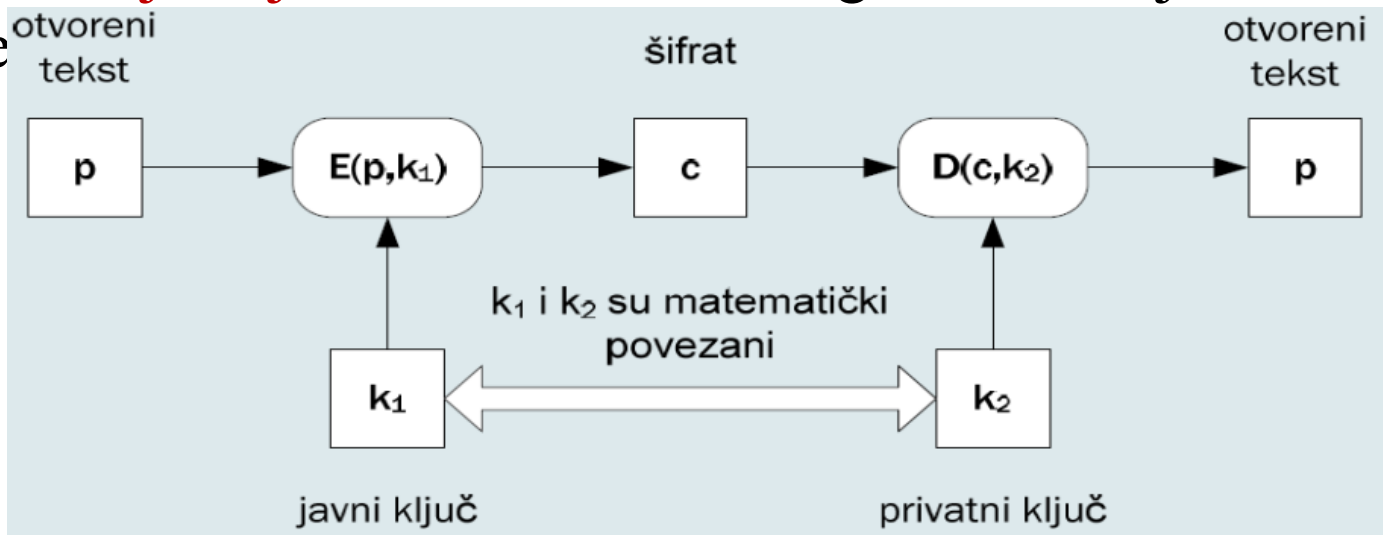
4.3 – Heš funkcije

Algoritam	Veličina sažetka (u bitovima)	Otpornost na sudare (složenost)	Otpornost na inverziju (složenost)
HAVAL	256/224/192/160/128	Da	
MD2	128	Skoro	
MD4	128	Da (2^8)	S greškama (2^102)
MD5	128	Da (2^5)	Ne
PANAMA	256	Da	
RadioGatún	Proizvoljne veličine	Ne	
RIPEMD	128	Da	
RIPEMD-128/256	128/256	Ne	
RIPEMD-160/320	160/320	Ne	
SHA-0	160	Da (2^39)	
SHA-1	160	S greškama (2^63)	Ne
SHA-256/224	256/224	Ne	Ne
SHA-512/384	512/384	Ne	Ne
Tiger(2)-192/160/128	192/160/128	Ne	
WHIRLPOOL	512	Ne	

4.4-Asimetrični algoritmi šifrovanja

2. **Algoritmi sa javnim ključem** - podaci se šifruju **javnim** ključem a dešifruju **privatnim** koji se razlikuje od javnog ključa.

- Šifrovanjem sa javnim ključem ***E*** priozvodi šifrat $\mathbf{c=E(p, k_1)}$ na osnovu javnog ključa (*public key*) k_1 i otvorenog teksta ***p***.
- Funkcija dešifrovanja ***D*** na osnovu privatnog ključa (*private key*) k_2 i šifrata ***c***, proizvodi orginalnu poruku $\mathbf{p=D(c, k_2)}$
- **Javni ključ** je poznat osobama sa kojima korisnik želi da komunicira, dok **privatni ključ** poznat samo korisniku koji je ovlašćen da dešifruje
- Asimetrični algoritmi su sporiji i primenjuju se za digitalno potpisivanje i šifrovanje ključeva simetričnih algoritama kojima su šifrovane datoteke



- Najpoznatiji algoritmi sa javnim ključem su:
  1. **RSA**
  2. **ElGamal**.

4.4 – RSA algoritam

- RSA je verovatno najpopularniji asimetrični kriptosistem.
- Sigurnost RSA zasniva se na složenosti faktorizacije velikih brojeva.
- **Javni** i **tajni ključ** određeni su parom velikih prostih brojeva (200 dekadnih cifara i više).
- Smatra se da je težina određivanja otvorenog teksta na osnovu šifrata, bez adekvatnog privatnog ključa, jedanaka težini faktorizacije proizvoda dva velika prosta broja.
- Par ključeva se generiše na sledeći način:

1. Najpre se generišu dva prosta broja p i q (oba preko 100 decimalnih cifara) i izračunavaju vrednosti **n=p·q i r=(p-1)·(q-1)**,
2. Bira se slučajan broj **e** u intervalu [**1,r-1**] koji je uzajamno prost sa **r** (jedini zajednički faktor za **e** i **r** je 1),
3. Izračunava se **d** tako da važi: **e·d=1 mod r**.
4. Vrednost **p**, **q**, **r** se čuvaju ili brišu.

- Privatni ključ (**d,n**) čuva se u tajnosti, dok je javni ključ (**e,n**) dostupan svima onima s kojima vlasnik privatnog ključa želi da komunicira.

4.4 - RSA algoritam

Primer: *Postupak izračunavanja parametra ključa Baneta nasumično bira dva velika prosta broja (oko 100 decimalnih mesta) ali zbog jednostavnosti koristićemo manje brojeve p = 3, i q = 11.*

Pratimo korake kao u postupku:

1. Izabrani p = 3,i q = 11
2. $n=p\cdot q=33$
3. $\varphi\, n = p-1 \cdot q-1 = 2\cdot 10 = 20$
4. Izaberemo javni ključ $e=3$
5. $d\equiv e-1\equiv 7 \bmod 20$

- Sada možemo krenuti sa enkripcijom i dekripcijom

1. Ana šalje poruku Banetu i poruku pretvara u broj x ($x=4$)
2. **$y=x^e \bmod n=4^3 \bmod 33 = 64 \bmod 33\equiv 31$**

- Ovime je izvršena enkripcija, odnosno poslana je poruka
- Sada je potrebna dekripcija kako bi se poruka pročitala.

$$x=y^d=31^7\equiv 4 \bmod 33$$

- Iz primera zaključujemo da su parametri koji definišu sigurnost RSA sistema p i q koji su u praksi definisani veličinom $p,q\geq 2512$.

4.5 – Digitalni potpis

- Digitalni potpis(*digital signature*) jeste elektronska verzija potpisa, na osnovu kog se može identifikovati pošiljalac i dokazati verodostojnost
- To je 51-bitni niz koji se dobija primenom RSA algoritma na hash vrednost generisanu iz bloka podataka koji se štiti i dodaje se na njegov kraj neposredno pre slanja.
- Digitalni potpisi su usko povezani sa jednosmernom heš funkcijom
- Prilikom potpisivanja, pošiljalac najpre jednosmernom heš funkcijom računa heš $\mathbf{h_1}$ poruke, koju nakon toga potpisuje svojim privatnim ključem (uslovno se može shvatiti kao šifrovanje privatnim ključem).

$$\mathbf{h_1=H(p)}$$
$$\mathbf{s_{1=}E(k_1, h_1).}$$

- Pošiljalac šalje orginalnu poruku i digitalni potpis primaocu.
- Primalac određuje **heš** $\mathbf{h_2}$ primljene poruke i proverava primljeni potpis $\mathbf{s_1}$ javnim ključem pošiljaoca (uslovno se može shvatiti kao dešifrovanje javnim ključem).

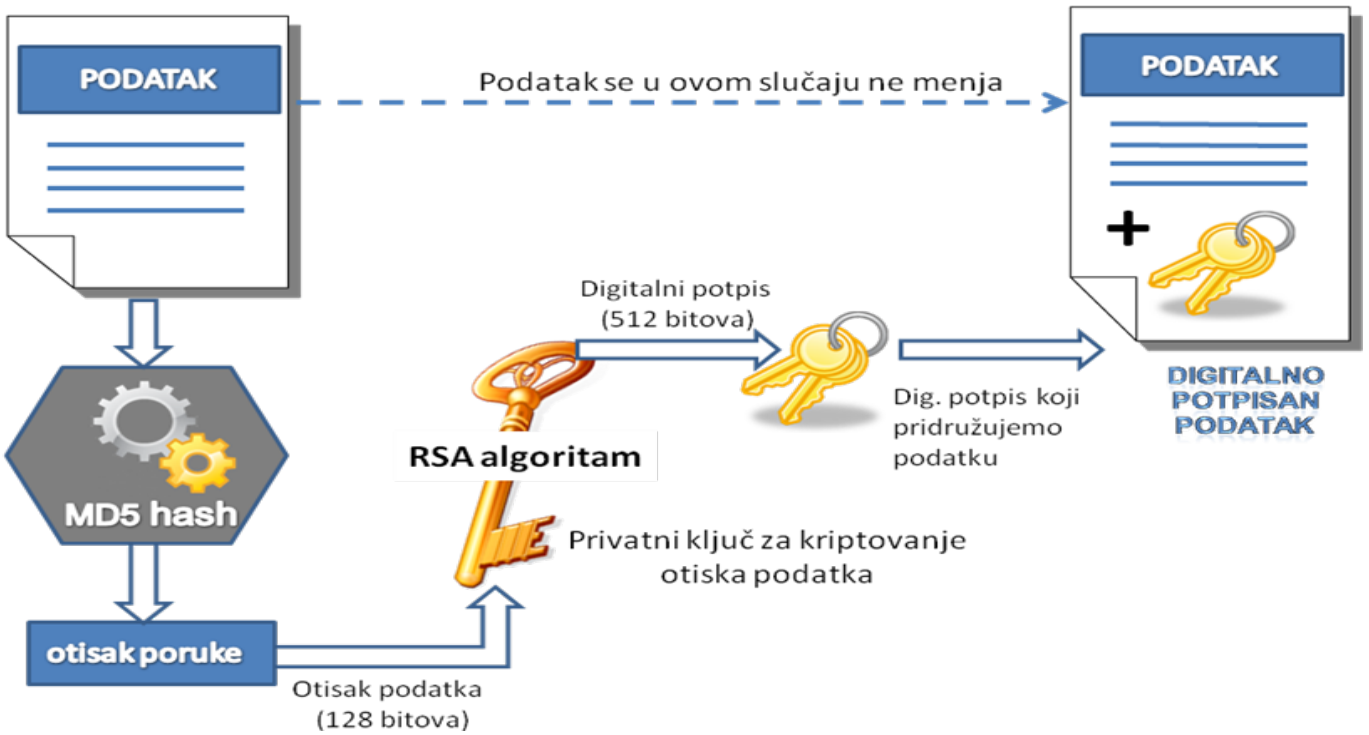
$$\mathbf{h_2=H(p),\ h_1=D(k_1, s_1)}$$

- Upoređivanjem vrednosti $\mathbf{h_1}$ i $\mathbf{h_2}$ proverava se identitet pošiljaoca.

4.5 – Digitalni potpis

Postupak kreiranja digitalnog potpisa se sastoji iz dve faze:

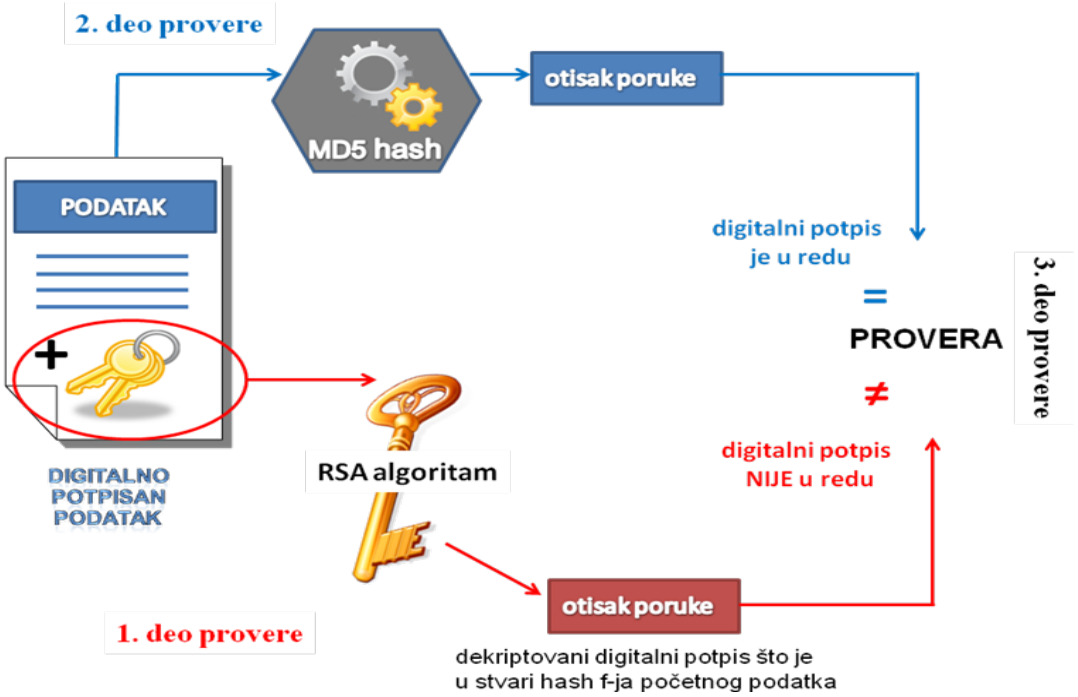
1. u prvoj fazi se primenom odgovarajuće kriptografske kompresione funkcije (MD5 hash) određuje otisak poruke *(message digest)*,
2. u drugoj fazi potpisnik poruke šifruje dobijeni otisak svojim tajnim (privatnim) ključem, primenom odgovarajućeg asimetričnog algoritma (RSA). Šifrovani otisak poruke predstavlja njen digitalni potpis i pridružuje joj se iza poslednjeg bajta.



4.5-Postupak provere digitalnog potpisa

Postupak verifikacije digitalnog postupka sastoji se iz 3 faze:

1. u prvoj fazi se iz dobijene poruke izdvaja digitalni potpis i dešifruje javnim ključem pošiljaoca
2. u drugoj fazi primalac formira hash funkciju dobijene poruke na isti načina kao što je to uradio pošiljaoc
3. u trećoj fazi vrši se poređenje, i ako je dobijeni otisak poruke identičan sa dešifrovanim otiskom, verifikacija je uspešna.



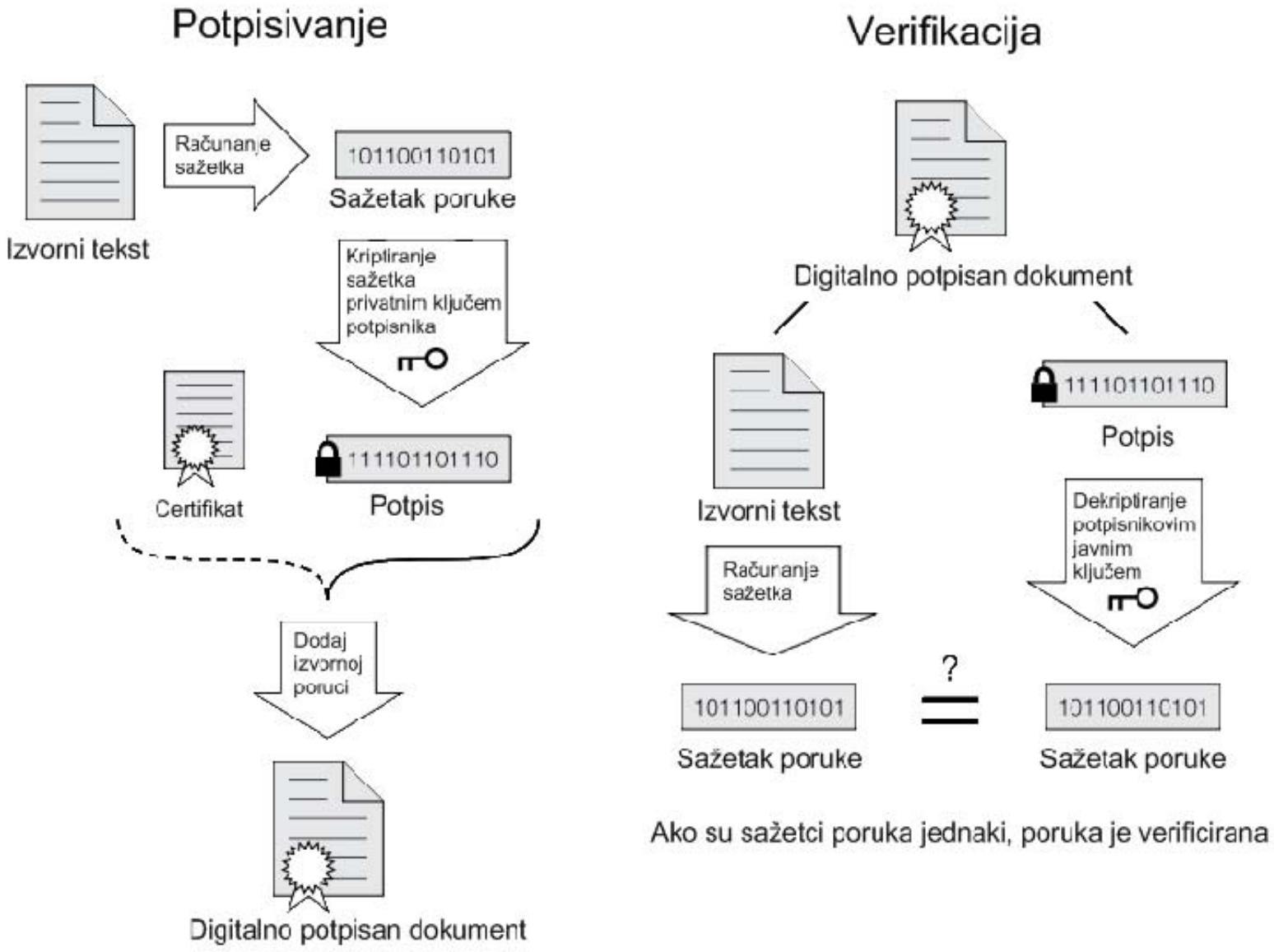
4.5-Postupak provere digitalnog potpisa

- Na osnovu iznetog može se zaključiti da je za funkcionalnost digitalnog potpisa potrebno izvršiti dva procesa, od kojih jedan sprovodi potpisnik, a drugi primalac.
- Uspešnom proverom digitalnog potpisa garantuje se:

1. ***Autentičnost***, pouzdanost identiteta pošiljaoca je posledica činjenice da je otisak poruke koji je šifrovan tajnim ključem, moguće uspešno dešifrovati samo primenom odgovarajućeg javnog ključa.
2. ***Integritet***, upoređivanjem izračunatog i dešifrovanog otiska poruke utvrđuje se da poruka nije modifikovana.
3. ***Neporecivost***, pošiljalac ne može da porekne slanje poruke pošto je potpisana njegovim tajnim ključem.

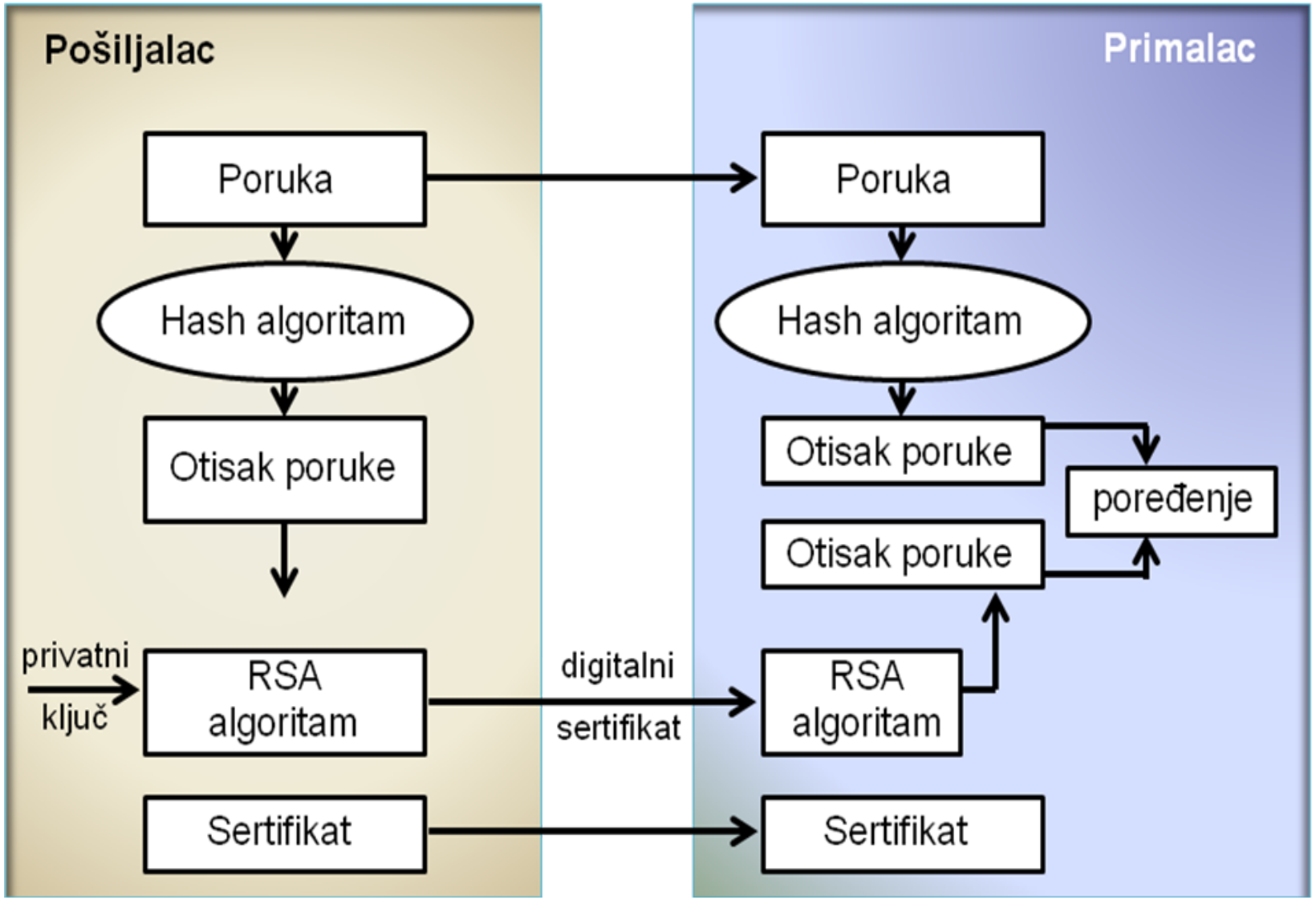
- Važno je pomenuti da elektronski potpisi uopšte, pa tako ni digitalni potpis ne pružaju zaštitu *Tajnosti* podataka od neovlašćenog čitanja, jer se svi podaci šalju u svom originalnom (nepromenjenom) obliku.
- Postupci kreiranja i verifikacije digitalnog potpisa prolaze kroz postupke modifikacije vec čitavu deceniju, i mogu se automatizovati toliko da je ljudska interakcija potrebna samo u izuzetnim slučajevima

4.5 – Digitalni potpis



Digitalno potpisivanje dokumenta i verifikacija

4.5 - Postupak provere digitalnog potpisa



4.5 - Digitalni potpis - primer

Primer: *Pretpostavimo da Ana, čiji je privatni ključ(**d,n**) a javni ključ (**e,n**), šalje Banetu poruku **m**, ali Bane zahteva od Ane da na neki način dokaže kako je baš ona poslala tu poruku.*

- U ovom slučaju, komunikacija se odvija po sledećem protokolu:
  1. Ana računa potpis **s** pomoću svog privatnog ključa: $\mathbf{s=m^d \bmod n}$,
  2. Ana šalje Banetu poruku i potpis, to jest uređeni par (**m,s**),
  3. Bane dešifruje potpis **s** koristeći Anin javni ključ: $\mathbf{m_1 = s^e \bmod n}$,
  4. ako je $m_1=m$, Bane prihvata poruku zato što jedino Ana zna svoj privatni ključ kojim je poruka potpisana.
- Ako se za komunikaciju koristi prethodno opisan protokol, Ana mora da pošalje dvaput veću poruku (dužina RSA šifrata=otvoreni tekst).
- U slučaju da je poruka dužine 10MB, potpis će takođe biti dugačak 10MB, pa se primaocu šalje 20MB.
- Ovo premašenje se može smanjiti ukoliko se pre slanja ne potpisuje sama poruka, već njen heš.

4.6 – Digitalni sertifikat

- Korisnici sve više koriste Internet za razmenu osetljivih podataka
- Razmena podataka zahteva da korisnik ima osiguranje(uverenje) da je Web stranica koju posećuje valjana stranica organizacije/banke
- Digitalni certifikati pružaju to uverenje.
- Digitalni certifikat je elektronski dokument koji utvrđuje identitet i autentifikuje korisnika kada obavlja određene transakcije na Internetu.
- Certifikati koriste digitalne potpise za povezivanje javnih ključeva sa podacima o identitetu vlasnika, kao što su ime osobe ili organizacije, adresa i sl., i time sprečavaju neovlašćenu izmenu podataka.
- Certifikat sadrži identitet i javni ključ i povezuje ih u digitalni potpis.
- Jedno od rešenja izdavanja sertifikata mogao bi da bude jedan centar koji bi na zahtev slao javni ključ (nešto slično kao DHCP server).
- Nepraktičnost ovakvog rešenja se ogleda u velikom broju zahteva koje ne bi mogla da opsluži ni jedna familija servera.
- Zgodno bi bilo da takav distribucioni centar uopšte ne mora da bude na mreži dostupan 24/7, već da je moguće da na zahtev korisnika se izda sertifikat koji ima verodostojnost i validnost.

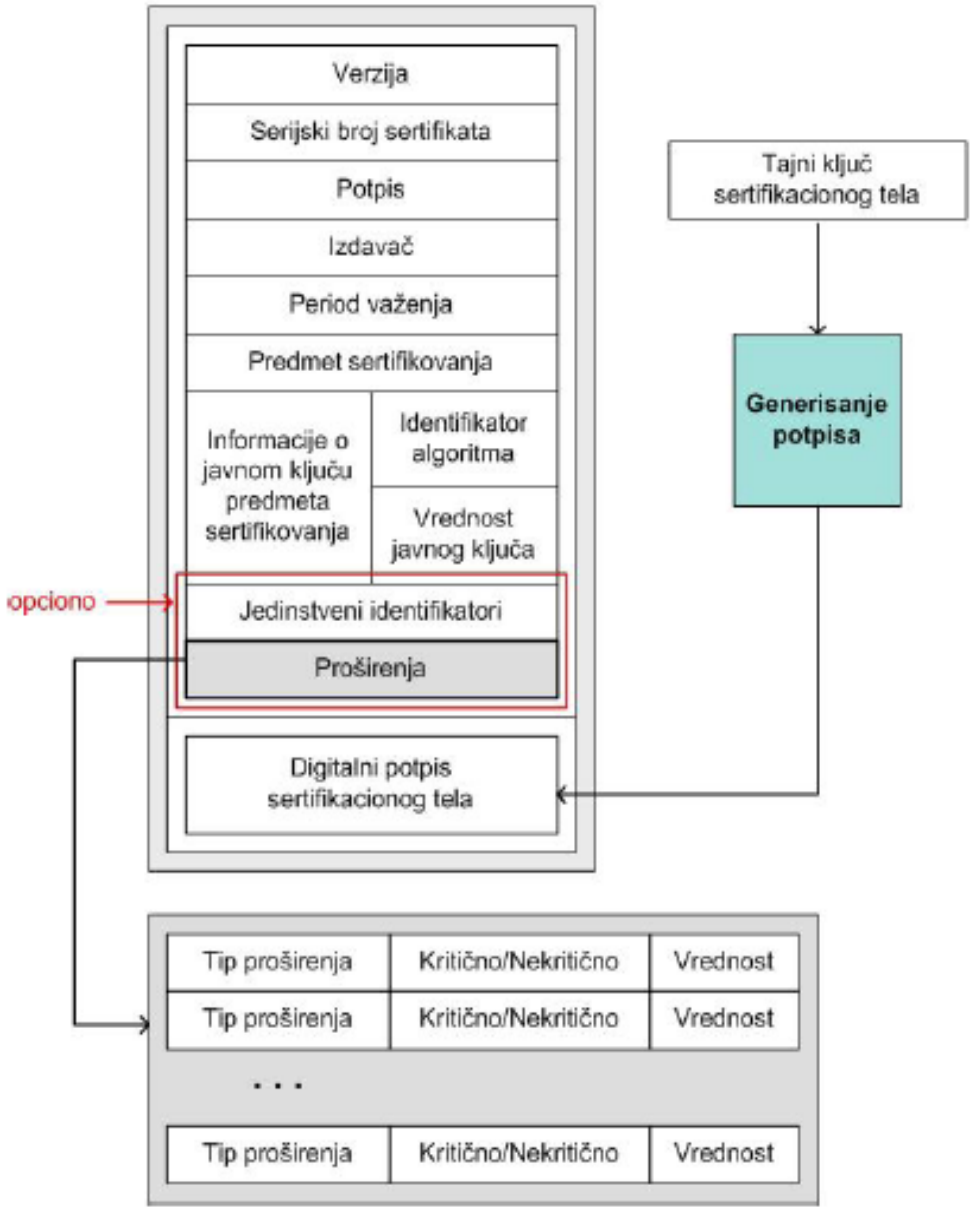
4.6 – Digitalni sertifikat

- Ovakvi centri postoje i to su **ovlašćene organizacije za izdavanje sertifikata** (*CA – Certification Authority*).
- Kada im se korisnik obrati sa zahtevom one izdaju sertifikat koji će sadržati budući javni ključ korisnika

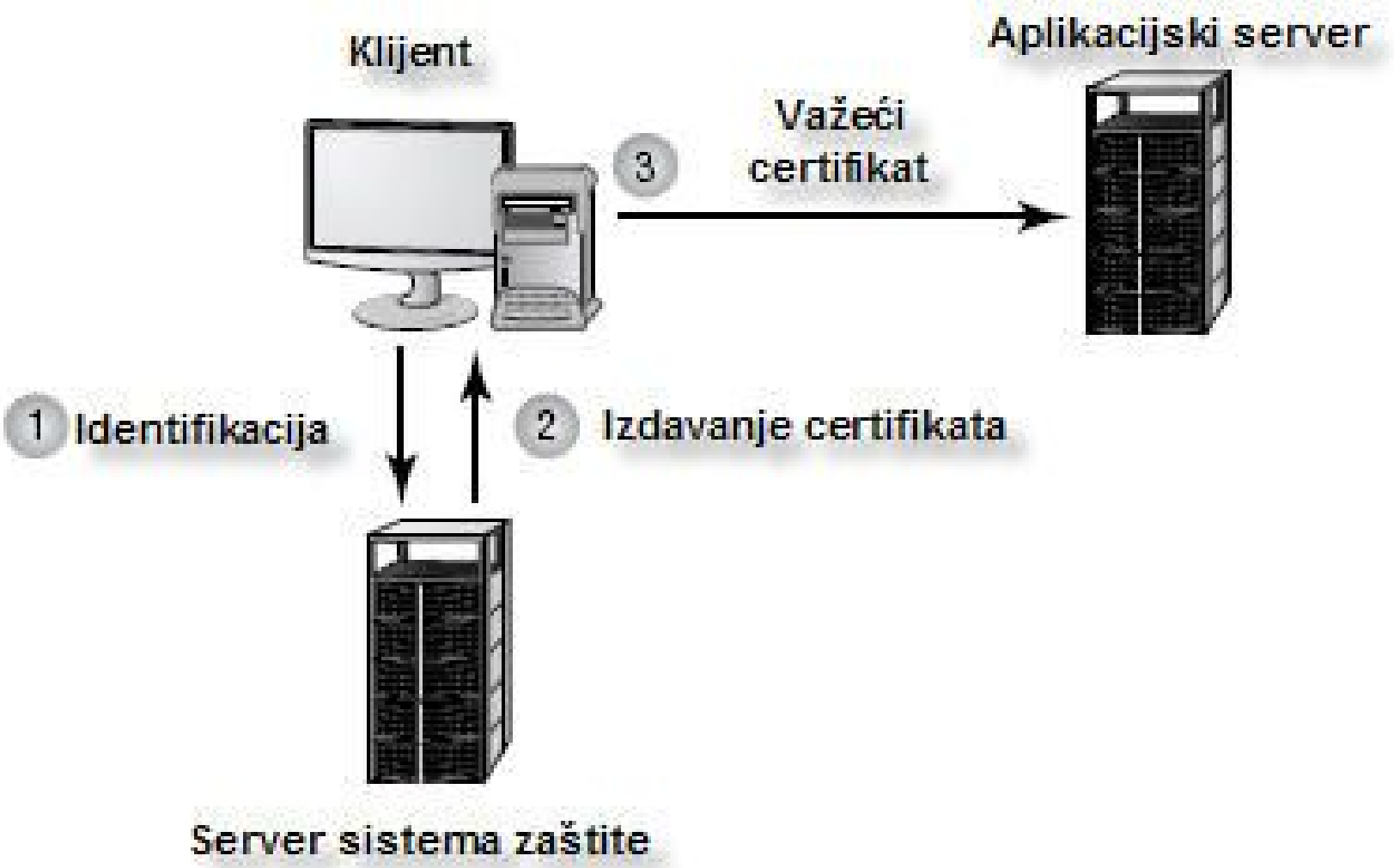
Elementi koji čine **strukturu digitalnog sertifikata** su:

- verzija formata sertifikata- ima oznaku strukture digitalnog sertifikata
- serijski broj sertifikata - sadrži vrednost koju dodeljuje sertifikacioni autoritet u trenutku kreiranja digitalnog sertifikata
- identifikator algoritma - sadrži podatke o algoritmu koji je pirmenjen
- naziv sertifikacionog tela - identifikuje izdavača digitalnog sertfikata
- rok važnosti sertifikata - vremenski period u kome važi izdati sertifikat
- vlasnik sertifikata - složena struktura koja obuhvata nekoliko podataka
- polje dodatnih atributa - sadrži vrednosti koje identifikuju vlasnika sertifikata a nisu sadržane u polju vlasnik sertifikata
- informacija o javnom ključu vlasnika – sadrži javni ključ i identifikator asimetričnog algoritma sa kojim se dati ključ primenjuje
- digitalni potpis sertifikata - od strane ustanove koja je izdala sertifikat

4.6 – Digitalni sertifikat



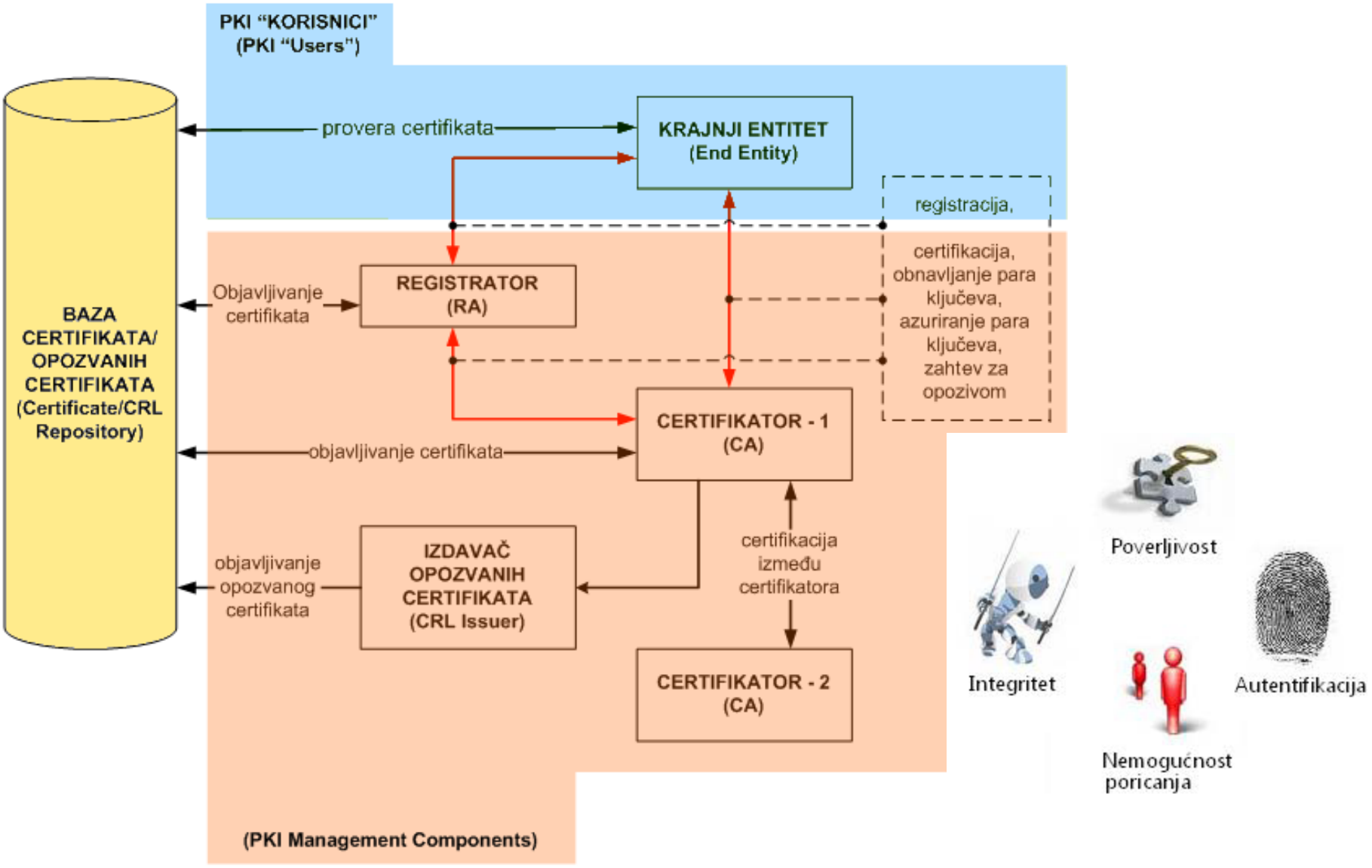
4.6 - Izdavanje sertifikata



4.7 – PKI sistem

- Infrastruktura sistema javnog ključa (*Public Key Infrastructure - PKI*) je složen sistem koji se temelji na asimetričnoj kriptografiji.
- PKI je skup programskih paketa, ljudi, politika i procedura koje su potrebne za stvaranje, upravljanje, spremanje, distribuciju i opozivanje digitalnih certifikata.
- PKI sistem koristi i digitalne potpise pri stvaranju certifikata koje uključuje upotrebu sažetka poruke.
- PKI je sporazum koji veže javne ključeve sa njihovim korisničkim identitetima preko sertifikacionog tela (sertifikatora).
- Korisnički identitet mora biti jedinstven za svakog sertifikatora.
- Namena PKI sistema je sigurna komunikacija preko nesigurnih kanala, a objedinjuje sertifikate, sertifikaciono telo (sertifikator), skladište sertifikata i opozvanih sertifikata, korisnike sertifikata i sve njihove međusobne interakcije-interakcije između pojedinih elemenata sistema
- PKI sistem omogućuje autentifikaciju i pruža brojne usluge, kao što su poverljivosti podataka, njihovog integriteta i upravljanje ključevima (*key managament*), odnosno sertifikatima.

4.7 – PKI sistem



Komponente PKI sistema

Hvala na pažnji !!!

Pitanja

? ? ?