

## 1. Simetrični i asimetrični algoritmi?

Simetrični algoritmi su vrsta kriptografskih algoritama koji koriste isti ključ za šifrovanje i dešifrovanje podataka. To znači da pošiljalac i primalac moraju imati isti ključ kako bi uspešno komunicirali. U simetričnoj komunikaciji, pošiljalac koristi dogovoreni algoritam i ključ za šifrovanje poruke, a primalac koristi isti algoritam i ključ za dešifrovanje poruke .

Asimetrični algoritmi, s druge strane, koriste par ključeva - javni ključ i tajni ključ. Javni ključ se deli sa drugima, dok se tajni ključ čuva tajnim. Pošiljalac koristi javni ključ primaoca za šifrovanje poruke, a primalac koristi svoj tajni ključ za dešifrovanje poruke. Ova vrsta algoritama omogućava sigurnu komunikaciju bez potrebe za deljenjem tajnog ključa .

Ukratko, simetrični algoritmi koriste isti ključ za šifrovanje i dešifrovanje, dok asimetrični algoritmi koriste par ključeva - javni i tajni ključ - za sigurnu komunikaciju.

## 2. Tipovi napada?

- 1) known-ciphertext - Napadač poseduje sadržaj šifriranih poruka i pokušava da dođe do otvorenog sadržaja poruka ili da izračuna ključ za dešifrovanje .
- 2) known-plaintext - Napadač poseduje sadržaj otvorenih poruka i odgovarajućih šifriranih poruka. Cilj mu je da dođe do ključa .
- 3) chosen-plaintext - Slično prethodnom, ali napadač može i da bira koji tekst će biti šifrovan .
- 4) adaptive-chosen-plaintext - Slično prethodnom, ali napadač može i da bira tekst za šifrovanje na osnovu rezultata prethodnih pokušaja .
- 5) chosen-ciphertext - Napadač može da bira različite šifrirane tekstove i može da ih dešifruje. Cilj mu je da dođe do ključa .
- 6) chosen-key - Napadač poseduje informacije o odnosima između različitih ključeva .
- 7) rubber-hose - Napadač pokušava da dođe do ključa pretnjama, ucenom, podmićivanjem ili mučenjem

## 3. Šifre zamene i šifre premestanja?

Šifre zamene i šifre premestanja su dva osnovna tipa šifri koji se koriste u kriptografiji.

Šifre zamene su šifre koje zamenjuju svaki karakter otvorenog teksta nekim drugim znakom u šifrovanom tekstu. Postoje različite vrste šifri zamene, uključujući monoalfabetske šifre, homofonske šifre, poligramske šifre i polialfabetske šifre .

Monoalfabetske šifre su šifre u kojima se svaki karakter otvorenog teksta zamenjuje jednim znakom u šifrovanom tekstu. Homofonske šifre su šifre u kojima jednom karakteru otvorenog teksta odgovara više karaktera u šifrovanom tekstu. Poligramske šifre su šifre u kojima se zamena vrši nad grupama karaktera. Polialfabetske šifre koriste više monoalfabetskih šifara koje se smenjuju sa svakim šifriranim znakom .

Šifre premestanja su šifre koje premeštaju karaktere u tekstu. Postoje različite vrste šifri premestanja, uključujući šifre koje menjaju redosled karaktera u tekstu, šifre koje menjaju redosled reči u tekstu i šifre koje menjaju redosled blokova teksta .

U modernoj kriptografiji, mnogi algoritmi kombinuju zamenu i premestanje kako bi se postigla veća sigurnost.

## 4. Rotorska masina?

Rotorska mašina je kriptografski uređaj koji se koristio za šifrovanje i dešifrovanje poruka tokom Drugog svetskog rata. Ova mašina se sastoji od tastature i niza rotora. Svaki rotor predstavlja permutaciju alfabeta

i rotori su međusobno povezani. Rotori se okreću u različitim koracima, a period ponavljanja za nrotorsku mašinu je  $26^n$ .

Jednokratna sveska, s druge strane, je vrsta šifre koja se smatra savršenom šifrom. Ova šifra koristi veliki neponavljajući niz slučajnih slova, koji se naziva beskonačna traka za teleprintere. Svaki znak otvorenog teksta se šifruje sabiranjem sa odgovarajućim znakom iz niza po modulu 26. Nakon što se znak iz niza upotrebi za šifrovanje, ne može se više koristiti. Sigurnost ove šifre zavisi od toga da beskonačna traka sadrži zaista slučajan niz.

## 5. Protokol?

Protokol je serija postupaka ili koraka koje učesnici moraju slediti kako bi obavili određeni zadatak. Protokoli se koriste u različitim oblastima, uključujući informacionu bezbednost, komunikaciju, mreže i druge.

Postoje različiti tipovi protokola, uključujući:

- i. Arbitrirani protokoli : Ovi protokoli uključuju treću osobu, arbitra, kojoj svi učesnici veruju. Arbitar je nepristrasna strana koja pomaže u rešavanju konflikata i donošenju odluka. Primeri arbitriranih protokola uključuju protokole koji se koriste u pravnim sistemima, bankarstvu i slično .
- ii. Adjudicirani protokoli : Ovi protokoli takođe uključuju arbitra, ali angažovanje arbitra može biti skupo. Stoga se protokol deli na dva podprotokola: jedan koji se sprovodi uvek (nearbitrirani) i drugi koji se sprovodi samo kada su učesnici u konfliktu (arbitrirani) .
- iii. Samoregulišući (self-enforcing) protokoli : Ovi protokoli su dizajnirani tako da sami garantuju ispravnost i integritet, bez potrebe za posrednikom ili arbitrom. Oni se oslanjaju na matematičke algoritme i mehanizme kako bi osigurali sigurnost i pravilno funkcionisanje. Primeri samoregulišućih protokola uključuju neke kriptografske protokole .

Važno je da protokoli budu nedvosmisleni, što znači da koraci moraju biti jasno definisani i da ne sme biti prostora za nesporazume. Takođe, protokoli moraju biti kompletni

## 6. Komunikacija pomoću simetričnih algoritama?

Komunikacija pomoću simetričnih algoritama se odnosi na proces šifrovanja i dešifrovanja poruka između dve strane, koristeći isti ključ za obe operacije. Ovi algoritmi se nazivaju simetričnim jer koriste isti ključ za šifrovanje i dešifrovanje.

Proces komunikacije pomoću simetričnih algoritama obično uključuje sledeće korake:

1. Dogovor algoritma : Strane se dogovaraju o kojem simetričnom algoritmu će koristiti za šifrovanje i dešifrovanje poruka.
2. Dogovor ključa : Strane se dogovaraju o zajedničkom ključu koji će koristiti za šifrovanje i dešifrovanje poruka. Taj ključ mora biti tajan i poznat samo stranama koje učestvuju u komunikaciji.
3. Šifrovanje poruke : Pošiljalac koristi dogovoreni algoritam i ključ za šifrovanje poruke. Poruka se transformiše u šifrirani oblik koji je nerazumljiv za neovlaštene osobe.
4. Slanje šifrirane poruke : Šifrirana poruka se šalje preko nezaštićenog kanala komunikacije, kao što je internet ili bežična mreža.

5. Dešifrovanje poruke : Primalac koristi isti dogovoreni algoritam i ključ za dešifrovanje primljene šifrirane poruke. Poruka se vraća u originalni oblik.

Važno je napomenuti da simetrični algoritmi zahtevaju siguran prenos ključa između strana kako bi se osigurala tajnost komunikacije. Takođe, simetrični algoritmi su brzi i efikasni, ali zahtevaju prethodni dogovor o ključu.

## 7. Jednosmerne funkcije?

Jednosmerne funkcije su matematičke funkcije koje su relativno jednostavne za izračunavanje u jednom smeru, ali su teške za obrnuti proces, odnosno za izračunavanje inverzne vrednosti.

Konkretno, za datu jednosmernu funkciju  $f(x)$ , lako je izračunati  $f(x)$  za bilo koju vrednost  $x$ . Međutim, teško je izračunati  $x$  ako je poznata vrednost  $f(x)$ . Ova svojstva jednosmernih funkcija čine ih korisnim u kriptografiji, gde se koriste za šifrovanje i zaštiti podataka.

Važno je napomenuti da ne postoji matematički dokaz da jednosmerne funkcije postoje, ali za neke funkcije možemo reći da su jednosmerne jer ne znamo lak način da izračunamo inverznu funkciju. Jedan primer jednosmerne funkcije je kvadratna funkcija  $f(x) = x^2$ , gde je lako izračunati  $f(x)$  za bilo koju vrednost  $x$ , ali je teško izračunati  $x$  ako je poznata vrednost  $f(x)$ .

Postoje i posebne vrste jednosmernih funkcija koje se nazivaju "trapdoor" jednosmerne funkcije. Ove funkcije imaju dodatno svojstvo da se inverzna vrednost može lako izračunati ako se zna određena tajna informacija, poznata kao "trapdoor". Ovo svojstvo omogućava da se efikasno koriste u kriptografiji za generisanje ključeva i digitalne potpise.

Jednosmerne hash funkcije su matematičke funkcije koje generišu kratke, fiksne izlazne vrednosti (hash vrednosti) za ulazne podatke bilo koje dužine. One su lako izračunljive, ali je teško obrnuto izračunati ulaznu vrednost na osnovu hash vrednosti. Ove funkcije se koriste za proveru integriteta podataka, generisanje digitalnih potpisa i skladištenje lozinki. Hash funkcije su otporne na obrnuti inženjering i obezbeđuju jedinstvenost, što znači da različiti podaci ne mogu proizvesti istu hash vrednost.

## 8. Komunikacija pomoću asimetričnih algoritama?

Komunikacija pomoću asimetričnih algoritama se odnosi na proces šifrovanja i dešifrovanja poruka između dve strane, koristeći različite ključeve za obe operacije. Ovi algoritmi se nazivaju asimetričnim jer koriste različite ključeve za šifrovanje i dešifrovanje.

Proces komunikacije pomoću asimetričnih algoritama obično uključuje sledeće korake:

1. Dogovor algoritma : Strane se dogovaraju o kojem asimetričnom algoritmu će koristiti za šifrovanje i dešifrovanje poruka.
2. Generisanje ključeva : Svaka strana generiše par ključeva - javni ključ i tajni ključ. Javni ključ se može slobodno deliti sa drugima, dok se tajni ključ čuva tajnim.
3. Razmena javnih ključeva : Strane razmenjuju svoje javne ključeve.
4. Šifrovanje poruke : Pošiljalac koristi javni ključ primaoca za šifrovanje poruke. Poruka se transformiše u šifrirani oblik koji je nerazumljiv za neovlaštene osobe.

5. Slanje šifrirane poruke : Šifrirana poruka se šalje preko nezaštićenog kanala komunikacije, kao što je internet ili bežična mreža.
6. Dešifrovanje poruke : Primalac koristi svoj tajni ključ za dešifrovanje primljene šifrirane poruke. Poruka se vraća u originalni oblik.

Važno je napomenuti da asimetrični algoritmi su sporiji od simetričnih algoritama, ali su korisni u situacijama gde nije moguće unapred dogovoriti zajednički tajni ključ. Asimetrični algoritmi se takođe koriste za digitalne potpise, gde pošiljalac koristi svoj tajni ključ za šifrovanje poruke, a primalac koristi pošiljačev javni ključ za proveru da li je poruka zaista poslata od strane pošiljaoca. Asimetrični algoritmi se takođe koriste za generisanje ključeva za simetrične algoritme, gde se javni ključ koristi za šifrovanje simetričnog ključa koji se zatim koristi za šifrovanje i dešifrovanje poruka.

## 9. Hibridni kriptosistemi?

Hibridni kriptosistemi su kombinacija asimetričnih i simetričnih kriptografskih algoritama koji se koriste za sigurnu razmenu podataka. Ovi sistemi koriste prednosti oba tipa algoritama kako bi obezbedili efikasnu i sigurnu komunikaciju.

U hibridnim kriptosistemima, asimetrični algoritmi se koriste za razmenu ključeva za simetrične algoritme. Ovo rešava problem razmene tajnih ključeva koji je prisutan kod čistih simetričnih algoritama.

Proces komunikacije u hibridnom kriptosistemu obično uključuje sledeće korake:

1. Inicijalna razmena ključeva: Početna razmena ključeva se obavlja pomoću asimetričnih algoritama. Bob šalje svoj javni ključ Alice, koju ona koristi za generisanje slučajnog ključa za sesiju.
2. Šifrovanje ključa za sesiju: Alice šifrue generisani ključ za sesiju Bobovim javnim ključem i šalje mu šifriranu poruku.
3. Dešifrovanje ključa za sesiju: Bob dešifrue primljenu poruku koristeći svoj tajni ključ, čime dobija ključ za sesiju.
4. Komunikacija sa simetričnim algoritmom: Nakon uspešne razmene ključeva, Alice i Bob mogu nastaviti komunikaciju koristeći simetrični algoritam i ključ za sesiju. Simetrični algoritam je brži od asimetričnih algoritama, što omogućava efikasnu razmenu podataka.

Hibridni kriptosistemi pružaju sigurnu i efikasnu komunikaciju, kombinujući prednosti asimetričnih i simetričnih algoritama. Asimetrični algoritmi se koriste za sigurnu razmenu ključeva, dok se simetrični algoritmi koriste za brzo šifrovanje i dešifrovanje samih poruka. Ovaj pristup kombinuje efikasnost simetričnih algoritama sa sigurnošću asimetričnih algoritama, pružajući tako optimalno rešenje za sigurnu komunikaciju.

Hibridni kriptosistemi su široko korišćeni u praksi, posebno u sistemima kao što su sigurna komunikacija putem interneta, elektronska pošta i elektronsko bankarstvo. Kombinacija asimetričnih i simetričnih algoritama omogućava efikasnu i pouzdanu razmenu podataka, čime se obezbeđuje privatnost i integritet informacija.

## 10. Digitalni potpisi?

Digitalni potpis je kriptografski mehanizam koji se koristi za proveru autentičnosti i integriteta digitalnih dokumenata. Digitalni potpis se generiše pomoću jednosmernih hash funkcija i asimetričnih kriptografskih algoritama. Imaju osobine običnih potpisa.

Digitalni potpisi se obično generišu pomoću asimetričnih algoritama, jer oni pružaju veću sigurnost i pouzdanost u odnosu na simetrične algoritme. Međutim, u nekim situacijama se koriste i simetrični algoritmi za generisanje digitalnih potpisa.

Jedan od načina da se koriste simetrični algoritmi za generisanje digitalnih potpisa je korišćenje HMAC (Hash-based Message Authentication Code) funkcija. HMAC funkcije koriste simetrične ključeve za generisanje hash vrednosti poruke, koja se zatim koristi za generisanje potpisa. HMAC funkcije su brže od asimetričnih algoritama, ali pružaju manju sigurnost.

Drugi način da se koriste simetrični algoritmi za generisanje digitalnih potpisa je korišćenje tajnih ključeva za simetrične algoritme. Ovaj pristup se obično koristi u situacijama gde je potrebna brza i efikasna generacija potpisa, ali gde nije potrebna visoka sigurnost. Tajni ključ se koristi za generisanje potpisa, koji se zatim šalje zajedno sa dokumentom.

Međutim, u većini slučajeva se koriste asimetrični algoritmi za generisanje digitalnih potpisa, jer pružaju veću sigurnost i pouzdanost. Asimetrični algoritmi se koriste za generisanje ključeva za digitalne potpise, kao i za potpisivanje hash vrednosti dokumenata. Asimetrični algoritmi se obično koriste u kombinaciji sa jednosmernim hash funkcijama, koje se koriste za generisanje hash vrednosti dokumenata.

#### **Proces generisanja digitalnog potpisa obično uključuje sledeće korake:**

1. Izračunavanje hash vrednosti : Pošiljalac izračunava hash vrednost dokumenta pomoću jednosmerne hash funkcije.
2. Potpisivanje hash vrednosti : Pošiljalac potpisuje hash vrednost pomoću svog privatnog ključa, koristeći asimetrični kriptografski algoritam.
3. Slanje potpisanog dokumenta : Pošiljalac šalje dokument i potpis primaocu.
4. Provera potpisa : Primalac koristi javni ključ pošiljaoca za dešifrovanje potpisa i dobijanje hash vrednosti dokumenta. Zatim, primalac izračunava hash vrednost primljenog dokumenta pomoću iste jednosmerne hash funkcije. Ako se hash vrednosti poklapaju, to znači da je dokument autentičan i da nije izmenjen u prenosu.

Digitalni potpisi se koriste u različitim aplikacijama, uključujući elektronsko bankarstvo, elektronsku poštu, elektronsko glasanje i druge. Oni obezbeđuju autentičnost i integritet digitalnih dokumenata, čime se sprečava neovlašćen pristup i manipulacija podacima.

## **11. Sertifikati?**

Sertifikat je digitalni dokument koji se koristi za autentifikaciju identiteta osobe, organizacije ili uređaja. Sertifikati se obično koriste u kombinaciji sa asimetričnim kriptografskim algoritmima, kao što su RSA ili ECC, kako bi se obezbedila sigurna razmena podataka.

Sertifikati se izdaju od strane Certificate Authority (CA), koji je poverljivo lice koje potvrđuje identitet osobe, organizacije ili uređaja. CA koristi svoj privatni ključ za potpisivanje sertifikata, čime se garantuje da je sertifikat autentičan i da je izdao CA.

Sertifikati obično sadrže sledeće informacije:

- Ime vlasnika sertifikata
- Javni ključ vlasnika sertifikata
- Ime Certificate Authority (CA) koji je izdao sertifikat
- Datum izdavanja sertifikata
- Datum isteka sertifikata
- Potpis CA

Sertifikati se koriste u različitim aplikacijama, uključujući elektronsko bankarstvo, elektronsku poštu, elektronsko glasanje i druge. Oni obezbeđuju autentičnost i integritet digitalnih dokumenata, čime se sprečava neovlašćen pristup i manipulacija podacima.

Kada se koristi sertifikat, aplikacija koja koristi sertifikat proverava da li je sertifikat autentičan i da li je izdao CA. Ako je sertifikat autentičan, aplikacija koristi javni ključ vlasnika sertifikata za šifrovanje ili dešifrovanje podataka, čime se obezbeđuje sigurna razmena podataka.

## 12. Razmena ključeva?

Simetrična razmena ključeva zahteva prethodni dogovor o zajedničkom tajnom ključu koji se koristi za enkripciju i dekripciju poruka. Tajni ključ se generiše i šalje preko sigurnog kanala. Ova vrsta razmene ključeva je brza i efikasna.

Asimetrična razmena ključeva koristi par ključeva - javni ključ i tajni ključ. Javni ključ se deli javno, dok je tajni ključ poznat samo vlasniku. Razmena se odvija tako što jedna strana šalje svoj javni ključ, druga strana generiše svoj par ključeva i enkriptuje svoj tajni ključ sa primljenim javnim ključem. Nakon dekripcije, obe strane mogu koristiti tajni ključ za sigurnu komunikaciju. Ova vrsta razmene ključeva omogućava sigurnu komunikaciju bez prethodnog dogovora o zajedničkom tajnom ključu, ali je sporija od simetrične razmene ključeva.

Važno je napomenuti da se asimetrični algoritmi koriste za razmenu ključeva, dok se simetrični algoritmi koriste za enkripciju/dekripciju samih poruka. Nakon uspešne razmene ključeva, dalja komunikacija može se odvijati korišćenjem simetričnih algoritama, koji su brži od asimetričnih algoritama.

Postoje četiri vrste razmene ključeva:

1. Interlok protokol : Alice i Bob razmenjuju svoje javne ključeve. Ova razmena omogućava Alice i Bobu da imaju javne ključeve jedno o drugom.
2. Razmena sesijskog ključa : Alice generiše slučajni sesijski ključ, enkriptuje ga Bobovim javnim ključem i šalje ga Bobu. Nakon ove razmene, dalja komunikacija između Alice i Boba koristi sesijski ključ, što omogućava sigurnu komunikaciju.
3. Komunikacija bez prethodne razmene ključeva : Alice generiše sesijski ključ, enkriptuje ga Bobovim javnim ključem i šalje enkriptovani ključ zajedno sa enkriptovanom porukom Bobu. Bob dešifruje sesijski ključ koristeći svoj tajni ključ i koristi ga za dekripciju poruke. Ova vrsta razmene ključeva omogućava sigurnu komunikaciju bez prethodne razmene ključeva, ali zahteva korišćenje asimetričnih algoritama i digitalnih potpisa.
4. Digitalni potpisi sa šifrovanjem dokumenata : Alice potpisuje poruku svojim privatnim ključem, šifruje poruku Bobovim javnim ključem i šalje poruku Bobu. Bob dešifruje poruku svojim tajnim ključem i proverava potpis Alicinim javnim ključem. Ova vrsta razmene ključeva omogućava sigurnu komunikaciju i proveru autentičnosti poruke korišćenjem digitalnih potpisa i asimetričnih algoritama.

## 5. Dužina ključeva?

Dužina ključeva je jedan od faktora koji utiče na sigurnost kriptografskih algoritama. Dužina ključeva se meri u bitovima i veća dužina ključa pruža veću sigurnost. Dužina ključa određuje koliko mogućih kombinacija postoji i koliko vremena je potrebno za probijanje ključa metodom iscrpne pretrage.

Kod simetričnih algoritama tipične dužine ključa su 128, 192 ili 256 bitova. Što je ključ duži, to je veća sigurnost, ali i veći utrošak resursa za enkripciju i dekripciju.

Kod asimetričnih algoritama dužina ključa se često meri u bitovima za javne i privatne ključeve. Što je dužina ključa veća, to je teže probiti kriptografsku zaštitu, ali je i veće opterećenje za generiranje i obradu ključeva.

Važno je napomenuti da tehnološki napredak može smanjiti sigurnost kraćih ključeva s vremenom, pa se preporučuje korištenje dužih ključeva kako bi se osigurala adekvatna sigurnost u skladu s trenutnim standardima i preporukama stručnjaka.

## 6. Block vs stream?

Block cipheri i stream cipheri su dva osnovna tipa kriptografskih algoritama koji se koriste za šifrovanje podataka.

### 1. Block cipheri:

- Block cipheri operišu nad blokovima otvorenog i šifriranog teksta .
- Blok je tipično 64 bita .
- Block cipheri primenjuju fiksnu transformaciju na blok otvorenog teksta kako bi generisali šifrirani blok
- Primeri popularnih block ciphera su DES (Data Encryption Standard) i AES (Advanced Encryption Standard).

### 2. Stream cipheri:

- Stream cipheri operišu nad tokom otvorenog/šifriranog teksta po 1 bitu/bajtu/reči istovremeno .
- Stream cipheri generišu niz pseudoslučajnih bitova, poznat kao keystream, koji se kombinuje sa otvorenim tekstom koristeći XOR operaciju kako bi se generisao šifrirani tekst .
- Keystream generator generiše keystream na osnovu ključa i internog stanja . ☐ Primeri popularnih stream ciphera su RC4 i Salsa20.

Ukratko, block cipheri operišu nad blokovima podataka, dok stream cipheri operišu nad tokom podataka. Oba tipa algoritama imaju svoje karakteristike i primene u kriptografiji.

## 7. Rezimi rada?

### 1. ECB (Electronic Codebook):

- ECB je najjednostavniji režim rada .
- Otvoreni tekst se deli na blokove fiksne dužine, a svaki blok se zatim šifrjuje nezavisno od drugih blokova .
- ECB nije bezbedan za šifrovanje velikih količina podataka, jer identični blokovi otvorenog teksta daju identične blokove šifriranog teksta .

### 2. CBC (Cipher Block Chaining):

- CBC je režim rada koji koristi prethodni šifrirani blok kao ulaz za šifrovanje sledećeg bloka .
- CBC koristi inicijalizaciju vektora (IV) kako bi se obezbedila jedinstvenost šifriranog teksta .

- CBC je jedan od najčešće korišćenih režima rada .

### 3. CFB (Cipher Feedback):

- CFB je režim rada koji koristi prethodni šifrirani blok kao ulaz za keystream generator .
- Keystream se zatim kombinuje sa otvorenim tekstom kako bi se generisao šifrirani tekst .
- CFB se može koristiti sa bilo kojim blok cipherom ili stream cipherom .

### 4. OFB (Output Feedback):

- OFB je režim rada koji koristi keystream generator za generisanje keystreama .
- Keystream se zatim kombinuje sa otvorenim tekstom kako bi se generisao šifrirani tekst . - OFB se može koristiti sa bilo kojim blok cipherom ili stream cipher

## 8. Autentifikacija?

U autentifikaciji, učesnici su:

1. Prover (claimant): Osoba ili entitet koji se predstavlja kao korisnik i želi da potvrdi svoj identitet.
2. Verifier (recipient): Osoba ili entitet koji proverava identitet i autentičnost korisnika.

Alati za autentifikaciju uključuju:

- Šifrovane poruke: Korišćenje enkripcije za zaštitu poruka kako bi se potvrdila autentičnost pošiljaoca i poruke.
- Checksum funkcije: Izračunavanje kontrolnog broja (checksum) nad porukom i tajnim ključem radi autentifikacije.
- Heš funkcije: Korišćenje heš funkcija za generisanje jedinstvenog identifikatora poruke radi autentifikacije.

Protokol za autentifikaciju je proces u realnom vremenu kojim se utvrđuje identitet korisnika. Rezultat može biti da je korisnikov identitet autentičan ili da nije autentičan.

Karakteristike protokola za autentifikaciju uključuju:

- Zanemarljiva verovatnoća da treći učesnik, predstavljajući se kao prover, može da navede verifier-a na pozitivan rezultat autentifikacije.
- Verifier ne može koristiti informacije koje je dostavio prover kako bi se predstavio kao prover trećem učesniku.

Sredstva za autentifikaciju uključuju:

- Identifikator korisnika: Korisnik se identifikuje pomoću određenog identifikatora kao što je korisničko ime.
- Višefaktorska autentifikacija: Korisnik potvrđuje svoj identitet korišćenjem više faktora, kao što su nešto što zna (lozinka, PIN), nešto što ima (smart kartica, ključ) i nešto što je njegova karakteristika (otisak prsta, prepoznavanje lica).
- Lozinka: Najčešći metod autentifikacije koji zahteva unos tajne lozinke koja se upoređuje sa heširanim vrednostima.
- Jednokratne lozinke: Korišćenje različitih lozinki za svako prijavljivanje radi povećanja sigurnosti.
- PIN (personal identification number): Kratak niz cifara koji se koristi za autentifikaciju i često se skladišti na fizičkim uređajima kao što su kartice ili tokeni.

Klasifikacija šema za autentifikaciju uključuje:

- Slabe (weak) šeme: Jednostavne za implementaciju, ali podložne napadima. Primeri su lozinke i PINovi.
- Jake (strong) šeme: Zasnovane su na challenge-response protokolu i uključuju kriptografske tehnike radi povećanja sigurnosti.



Autentifikacija pomoću lozinke je najrašireniji metod autentifikacije koji se koristi. Lozinka se obično hešira i poredi sa heširanom vrednošću koja je skladištena u sistemu, čime se osigurava da otvoreni tekst lozinke nije trajno skladišten.

Autentifikacija pomoću PIN-a podrazumeva korišćenje kratkog niza cifara koji se skladišti na fizičkom uređaju, kao što je kartica ili token.

Ukratko, autentifikacija koristi različite alate, protokole i sredstva kako bi potvrdila identitet korisnika i osigurala sigurnost sistema.

## 9. Challenge-response?

Challenge-response je tehnika autentifikacije koja se koristi za proveru identiteta korisnika ili entiteta. Ova tehnika se zasniva na razmeni izazova (challenge) i odgovora (response) između proveravača (verifier) i korisnika (claimant).

U challenge-response postupku, proveravač generiše izazov (challenge) koji se šalje korisniku. Izazov može biti nasumični niz podataka ili neka druga vrednost koja se zahteva od korisnika da obradi ili odgovori na nju. Korisnik zatim generiše odgovor (response) na osnovu izazova koristeći određeni algoritam ili pravilo. Taj odgovor se zatim šalje nazad proveravaču.

Proveravač upoređuje pristigli odgovor sa očekivanim rezultatom. Ako se odgovor poklapa sa očekivanim, korisnik je uspešno autentifikovan. U suprotnom, autentifikacija nije uspela.

Challenge-response tehnika se često koristi u različitim kontekstima, kao što su pristup računarskim sistemima, bankovne transakcije, digitalni potpisi i druge situacije gde je potrebno proveriti identitet korisnika ili entiteta. Ova tehnika pruža dodatni sloj sigurnosti jer zahteva da korisnik poseduje odgovarajući algoritam ili znanje za generisanje ispravnog odgovora na izazov.

## 10. Napadi na mehanizme za autentifikaciju?

1. Napadi sa lažnim predstavljanjem (impersonation attacks):
  - Napadač dolazi u posed tajne informacije (lozinka, PIN, tajni ključ) i koristi je za autentifikaciju . - Napadač se predstavlja kao legitimni korisnik kako bi dobio pristup zaštićenim resursima .
2. Napadi sa ponovljenim porukama (replay attacks):
  - Napadač prisluškuje komunikaciju u toku autentifikacije i ponavlja je .
  - Napadač može ponoviti transakciju koju je već izvršio legitimni korisnik .
  - Anti-replay mere, kao što su timestamp, slučajan broj i slično, mogu se koristiti kako bi se sprečili ovakvi napadi .
3. Napadi sa uvođenjem kašnjenja (forced delay attacks):
  - Napadač presretne poruku, sačeka određeni period vremena, i prosledi je na odredište . - Korisnik dobije timeout, misli da je transakcija otkazana, a ona je stigla na odredište .
4. Napadi sa preplitanjem (interleaving attacks):
  - Napadač koristi informacije iz više tekućih postupaka za autentifikaciju .
  - Primer protokola: korisnik šalje sistemu šifrovani slučajan broj, sistem odgovara dešifrovanim brojem i šalje šifrat novog, korisnik odgovara dešifrovanim novim brojem

## 11. Autentifikacija pomocu lozinke?

Autentifikacija pomoću lozinke je najrašireniji metod autentifikacije. Korisnik se prijavljuje pomoću para (korisničko ime, lozinka), a server proverava da li dati par postoji u registru postojećih korisnika. Lozinka se smatra tajnim ključem, a postupak autentifikacije se zasniva na tome da se otvoreni tekst lozinke nigde ne skladišti trajno. Umesto toga, prilikom prijavljivanja izračunava se heš lozinke i poredi sa onim koji je skladišten u fajlu. Heš lozinke se čuva umesto same lozinke kako bi se sprečilo čitanje lozinke od strane privilegovanih korisnika sistema ili čitanje fajla iz bekapa.

## 12.CA?

Certificate Authority (CA) je pravno lice od poverenja koje ima ključnu ulogu u infrastrukturi javnih ključeva. Njegova glavna funkcija je potpisivanje sertifikata . CA izdaje digitalne sertifikate koji potvrđuju identitet entiteta, kao što su veb stranice, korisnici ili uređaji. Kada CA potpiše sertifikat, to znači da je CA verifikovao identitet subjekta i garantuje da je javni ključ u sertifikatu povezan sa tom identifikacijom. Ovo omogućava sigurnu razmenu podataka i uspostavljanje poverenja u digitalnom okruženju.

## 13.X.509?

X.509 je standardni format za sertifikate koji se koriste u kriptografiji radi autentifikacije. Sertifikati sadrže informacije o korisnicima i omogućavaju formiranje hijerarhijskog direktorijumskog servisa za autentifikaciju.

Postoje tri tipa autentifikacionih procedura u okviru X.509:

### 1. One-way autentifikacija:

- Provera šalje informacije verifieru kako bi potvrdio svoj identitet.
- Informacije uključuju formiranje poruke, upućivanje poruke verifieru i sprečavanje modifikacija ili ponovnog slanja (replay) poruke.
- Poruka sadrži slučajan broj (rand), vremensku oznaku (timestamp - tmp), identitet verifiera i potpisane podatke sa sertifikatom proveru.
- Opciono, poruka može sadržati i podatke koji se šifruju javnim ključem verifiera i mogu se koristiti za uspostavljanje sesijskog ključa.

### 2. Two-way autentifikacija:

- Omogućava obostranu autentifikaciju.
- Verifier šalje dodatnu poruku kojom potvrđuje svoj identitet.
- Poruka sadrži primljeni slučajan broj (rand), novi slučajan broj (rand') i potpisane podatke sa sertifikatom proveru kako bi se potvrdio identitet verifiera.

### 3. Three-way autentifikacija:

- Uključuje treću poruku koju proveru šalje verifieru.
- Poruka sadrži slučajan broj (rand') i potpisane podatke koji služe za sinhronizaciju časovnika.
- One-way i two-way autentifikacija se često koriste na webu u okviru SSL protokola, koji se dalje oslanja na X.509 sertifikate.

## 14.Kerberos?

Osnovna svrha Kerberos je omogućavanje sigurne autentifikacije i autorizacije u računarskim mrežama. Kerberos je osmišljen da omogući Single Sign-On (SSO) pristup. To znači da se korisnik prijavljuje samo jednom, a zatim ima pristup svim resursima na mreži u skladu sa svojim pravima, bez potrebe za ponovnim unošenjem lozinke.

Kerberos se koristi za upravljanje velikim brojem korisničkih naloga i omogućava efikasan pristup pojedinačnim resursima. Ključna komponenta Kerberos je Key Distribution Center (KDC), koji se sastoji

od tri glave: baze podataka, servera za proveru identiteta i servera za izdavanje karata. Kerberos koristi koncept principala za jednoznačno identifikovanje učesnika. Principali su povezani sa tajnim ključem za autentifikaciju kod KDC-a i sastoje se od identiteta, instance (opciono) i realm-a.

Kerberos koristi dve vrste karata: Ticket-Granting Ticket (TGT) i Service Ticket (ST). TGT karta se izdaje prilikom prijave na sistem i koristi se za dobijanje ST karata koje omogućavaju pristup pojedinačnim resursima.

KDC ima ulogu centra za distribuciju ključeva i autentifikaciju u Kerberos sistemu. Baza principala čuva informacije o principalu i njegovim tajnim ključevima, dok AS izdaje TGT karte korisnicima prilikom prijave, a TGS izdaje ST karte za pristup resursima.

Kerberos ima mogućnost da se implementira na različitim platformama, kao što su Linux sa LDAP bazom principala ili Windows sa Active Directory.

Važno je napomenuti da je Kerberos efikasan mehanizam za sigurnu autentifikaciju i autorizaciju u računarskim mrežama, posebno u sistemima kao što je Windows. Detaljnije informacije o Kerberosu i njegovim specifičnostima možete pronaći u dokumentaciji i literaturi o kriptografiji i sigurnosti mreža.

## 15.HTTP?

HTTP autentifikacija omogućava identifikaciju korisnika prilikom pristupa web resursima. Postoje dva tipa autentifikacije po HTTP standardu:

1. Basic Authentication : Relativno često korišćena metoda autentifikacije. Klijenti se identifikuju na osnovu korisničkog imena i lozinke. Kada klijent traži pristup resursu, server proverava da li je pristup ograničen. Ako jeste, server šalje odgovor "401 Unauthorized" zajedno sa zaglavljem "WWW-Authenticate" koje zahteva unos korisničkog imena i lozinke. Klijent zatim ponovo šalje zahtev sa kredencijalima u zaglavlju "Authorization". Username:password se kodiraju Base64 algoritmom, ali ovaj metod ne pruža zaštitu od prisluškivanja i smatra se nisko sigurnim.
2. Digest Access Authentication : Vrlo retko korišćena metoda autentifikacije. Ova metoda ispravlja neke nedostatke Basic metode. Umesto slanja korisničkog imena i lozinke kao otvorenog teksta, samo se šalje hash kod tih informacija. Na serverskoj strani se takođe čuva samo hash kod. Komunikacija se odvija po principu challenge-response. Server šalje kodiranu informaciju (nonce) klijentu, koji zatim odgovara sa korisničkim imenom, lozinkom i kodiranom informacijom. Odgovor se računa koristeći MD5 hash funkciju i specifične vrednosti iz zahteva. Ovaj metod pruža nešto viši nivo sigurnosti od Basic metode, ali i dalje ne obezbeđuje zaštitu od prisluškivanja ili man-in-the-middle napada.

Ukratko, Basic Authentication je jednostavniji i češće korišćen, ali manje siguran, dok je Digest Access Authentication složeniji, ali pruža nešto veću sigurnost.

## 16.OAuth?

OAuth 2.0 je autorizacioni okvir koji omogućava aplikacijama da pristupaju resursima u ime vlasnika tih resursa. To je protokol za delegaciju, gde vlasnik resursa dozvoljava nekom drugom subjektu (klijentu) da pristupa resursu u njegovo ime. OAuth 2.0 nije samo protokol za autentifikaciju, već omogućava i delegiranje prava pristupa putem tokena.

OAuth 2.0 je specifično dizajniran za web sisteme, posebno one koji su bazirani na REST arhitekturi. On implicitno uključuje autentifikaciju, ali je njegov glavni fokus na autorizaciji i delegaciji prava pristupa.

Proces OAuth 2.0 protokola uključuje nekoliko koraka. Kada klijentu treba OAuth access token, on redirektuje vlasnika resursa (korisnika) na autorizacioni server. Autorizacioni server zatim zahteva autentifikaciju od klijenta i dobija od vlasnika resursa informacije o tome šta klijent može da delegira. Nakon toga, autorizacioni server redirektuje vlasnika resursa nazad na klijentsku aplikaciju, pružajući joj jednokratne kredencijale (authorization code) koje klijent kasnije koristi za dobijanje access tokena. Klijent zatim šalje authorization code autorizacionom serveru zajedno sa svojim kredencijalima kako bi dobio access token. Sa dobijenim access tokenom, klijent pristupa zaštićenim resursima tako što šalje token u zahtevu putem HTTP Authorization zaglavlja.

OAuth 2.0 omogućava i upotrebu refresh tokena koji se koristi za obnavljanje access tokena kada istekne, bez potrebe za interakcijom sa vlasnikom resursa. Autorizacioni token može biti u bilo kom formatu, obično u JSON formatu, i klijent ga prosleđuje u zahtevu za pristup resursima. OAuth 2.0 takođe definiše i koncept scope-a, koji predstavlja skup prava za zaštićene resurse.

Postoje različite vrste autentifikacionih grantova u OAuth 2.0, kao što su eksplicitni (authorization code), implicitni, client credentials, resource owner credentials i assertion grant. Svaka vrsta granta ima svoje specifičnosti i koristi se u određenim situacijama.

OAuth 2.0 podržava različite vrste klijenata, kao što su javni klijenti koji ne zahtevaju autentifikaciju i poverljivi klijenti koji poseduju autentifikacione podatke.

Ukratko, OAuth 2.0 je autorizacioni okvir koji omogućava aplikacijama da pristupaju resursima u ime vlasnika resursa putem izdavanja i korišćenja tokena. To je fleksibilan protokol za delegaciju prava pristupa, posebno prilagođen web sistemima.

## **17.TCP/IP?**

Sigurnost TCP/IP steka se može implementirati na različitim nivoima, uključujući aplikacijski, transportni, mrežni i nivo veze.

Na aplikacijskom nivou, sigurnost se implementira u samim aplikacijama, pružajući fleksibilnost i potpuni pristup podacima. Međutim, zahteva izmenu svake aplikacije posebno.

Na transportnom nivou, sigurnost se postiže putem protokola poput TLS, koji obezbeđuje usluge provere identiteta, integriteta i poverljivosti preko TCP protokola. Ovde se izmene vrše na nivou transportnog protokola, dok aplikacije ostaju nepromenjene.

Na mrežnom nivou, sigurnost se ostvaruje kroz protokole kao što je IPSec. IPSec omogućava proveru identiteta, integritet i poverljivost preko IP protokola. Ova implementacija omogućava sigurnost za sve transportne veze koje koriste određeni protokol.

Na nivou veze, sigurnost se primenjuje na namenskim vezama između uređaja, poput bankomatskih mreža. Ova sigurnost obično zahteva hardverske uređaje za šifrovanje kako bi se osigurala brza obrada podataka na vezi.

Izbor odgovarajućeg nivoa implementacije sigurnosti zavisi od specifičnih zahteva i okruženja u kojem se primenjuje sigurnost TCP/IP komunikacije.

## **18.PGP?**

PGP se koristi za šifrovanje i digitalno potpisivanje poruka. Kod šifrovanja poruka putem PGP-a, otvoreni tekst se komprimuje i generiše se simetrični ključ za svaku poruku. Poruka se zatim šifrue simetričnim

algoritmom, a simetrični ključ se šifruje asimetričnim algoritmom pomoću javnog ključa primaoca. Tako se osigurava poverljivost poruka.

Digitalno potpisivanje poruka putem PGP-a uključuje računanje heša otvorenog teksta i potpisivanje tog heša privatnim ključem pošiljaoca. Pošiljalac šalje otvoreni tekst i potpisani heš primaocu, koji može proveriti autentičnost poruke.

PGP takođe koristi repozitorijume javnih ključeva, poznate kao key servere, gde korisnici mogu publikovati svoje javne ključeve i pronaći javne ključeve drugih korisnika. Ovo omogućava izgradnju "web of trust" mreže poverenja, gde korisnici mogu verovati ključevima drugih ljudi na osnovu reputacije i preporuka.

PGP je standardizovan kroz RFC 4880 u okviru IETF (Internet Engineering Task Force), što je omogućilo razvoj različitih implementacija PGP-a u aplikacijama, pluginovima za e-poštu i browserima. Neke od popularnih implementacija PGP-a su GnuPG, eM Client, The Bat!, Outlook, Thunderbird, i Enigmail.

Ukratko, PGP je programski alat koji pruža šifrovanje i digitalno potpisivanje poruka, omogućava izgradnju mreže poverenja putem repozitorijuma javnih ključeva i obezbeđuje sigurnu elektronsku komunikaciju.

## 19.SSL/TLS?

SSL (Secure Sockets Layer) je komunikacioni protokol koji je razvijen sa ciljem pružanja kriptografske bezbednosti i interoperabilnosti u komunikaciji putem mreže. SSL je prvobitno razvijen od strane kompanije Netscape i kasnije je standardizovan kao TLS (Transport Layer Security) u okviru IETF (Internet Engineering Task Force).

SSL/TLS omogućava sigurnu komunikaciju između klijenta i servera putem šifrovanja podataka i autentifikacije uključujući:

- SSL/TLS Record Protocol: Koristi simetrične algoritme za šifrovanje i hash funkcije za proveru integriteta poruka.
- SSL/TLS Handshake Protocol: Koristi asimetrične algoritme za autentifikaciju klijenta i servera, dogovor oko korišćenih algoritama i generisanje session ključa.

Tok komunikacije u SSL/TLS obuhvata sledeće korake:

- Klijent inicira vezu slanjem zahteva serveru sa spiskom podržanih šifara i heš funkcija.
- Server bira najjaču kombinaciju šifara i obaveštava klijenta.
- Server šalje svoju identifikaciju u vidu sertifikata.
- Klijent može proveriti validnost sertifikata kontaktiranjem sertifikacione autoritete (CA).
- Klijent enkriptuje slučajan broj javnim ključem servera i šalje ga serveru.
- Na osnovu slučajnog broja, klijent i server generišu session ključ koji se koristi za šifrovanje podataka tokom sesije.

SSL/TLS takođe koristi Alert Protocol za slanje obaveštenja o različitim stanjima veze, kao što su upozorenja o potencijalnim problemima ili fatalne greške koje mogu ugroziti komunikaciju.

Ukratko, SSL/TLS je protokol koji omogućava sigurnu komunikaciju putem mreže kroz kriptografsko šifrovanje i autentifikaciju, čime se osigurava privatnost i integritet podataka između klijenta i servera.

## 20.IPSEC?

IPSec je proširenje za IPv4 i integralni deo IPv6 koje pruža privatnost, integritet, proveru identiteta i neporecivost. Koristi se na mrežnom sloju TCP/IP steka i uključuje Authentication Header (AH) i Encapsulated Security Payload (ESP) protokole. AH pruža integritet i proveru identiteta, dok ESP pruža

poverljivost i integritet podataka. IPSec koristi različite algoritme za razmenu ključeva, šifrovanje i proveru identiteta. Može raditi u transportnom režimu (šifruje samo podatke) i tunelovanju (enkapsulira originalne IP pakete). Za uspostavu veze, koriste se protokoli kao što su ISAKMP i IKE.

## 21.IKE?

IKE (Internet Key Exchange) je protokol koji omogućava uspostavu sigurnosne veze i razmenu ključeva u IPSec protokolu. Postupak uspostave veze preko IKE se sastoji od uspostavljanja IKE SA parametara i IPSec SA parametara. IKE SA parametri definišu algoritme, heš funkcije i metode provere identiteta. Postoje dva režima rada: main mode (zaštićena identifikacija) i aggressive mode (brža veza bez identifikacije). Nakon uspostavljanja IKE SA, dolazi do uspostavljanja IPSec SA u quick mode. IPSec SA omogućava sigurnu komunikaciju i pruža poverljivost, integritet i autentičnost podataka.

## 22.Reference monitor?

Reference monitor je apstraktan pogled na podsistem za kontrolu pristupa i koristi informacije o pravima pristupa subjekata i objekata. Implementacija reference monitora se zasniva na principima kompletnosti, izolacije i proverivosti. Kompletnost zahteva da se zaštite svi objekti, čak i one koji nisu očigledni, dok izolacija osigurava da napadač ne može narušiti funkcionalnost reference monitora. Proverivost se postiže testiranjem i minimalizacijom funkcionalnosti sistema. Ova implementacija može uključivati i upotrebu sigurnosnog kernela, koji pruža usluge operativnom sistemu i omogućava razdvajanje koda i podataka aplikacija. Važno je i formalno modeliranje kernela i korišćenje formalnih metoda za proveru korektnosti.

## 23.Modeli kontrole pristupa?

Modeli kontrole pristupa predstavljaju apstraktan pogled na mehanizme za sprovođenje kontrole pristupa i omogućavaju jednostavniju analizu i izbor različitih implementacija. Neki od poznatih modela su Lampson model, Bell-LaPadula model, Clark-Wilson model i role-based access control (RBAC).

Lampson model se zasniva na konceptu "matrice pristupa" koja ima jedan red po subjektu i jednu kolonu po objektu. Matrica definiše dozvoljene operacije subjekata nad objektima. U implementaciji se često koriste mehanizmi poput lista sposobnosti (capability lists) i lista kontrole pristupa (access control lists).

Bell-LaPadula model je formalizacija vojnih pravila za kontrolu pristupa. Objekti imaju nivo poverljivosti, a korisnici imaju nivo pristupa. Osnovna pravila ovog modela su "no read up" i "no write down", što znači da subjekat može čitati samo objekte nižeg ili istog nivoa poverljivosti, dok može pisati samo u objekte višeg ili istog nivoa poverljivosti. Dodatno, model koristi kategorije kako bi ograničio pristup određenim objektima.

Važno je napomenuti da Bell-LaPadula model ima problem odlučivosti, što znači da ne može garantovati da će konfiguracija koja se smatra ispravnom ostati ispravna. Takođe, postoji mogućnost da korisnici nenamerno dodele prava pristupa kroz mehanizme delegiranja prava.

## 24.Discretionary Access Control?

Discretionary Access Control (DAC) je mehanizam kontrole pristupa koji omogućava korisnicima da delegiraju prava pristupa objektima na osnovu njihovog identiteta. Ovaj model se često koristi u fajlsistemima, relacionim bazama podataka i drugim informacionim sistemima.

U DAC modelu, korisnici imaju diskrecionu kontrolu nad dodeljivanjem i oduzimanjem prava pristupa objektima koje poseduju. Vlasnici objekata imaju posebnu ulogu jer imaju pravo da kontrolišu pristup svojim objektima. Oni mogu dodeliti prava pristupa drugim korisnicima, uključujući čitanje, pisanje, izvršavanje ili druge operacije nad objektom.

Jedan od najčešćih mehanizama za implementaciju DAC modela su Access Control Lists (ACLs), koje su liste koje povezuju subjekte (korisnike) sa objektima i definišu dozvoljene operacije. Svaki objekat ima svoju ACL listu koja sadrži informacije o korisnicima ili grupama korisnika i pravima koja im se dodeljuju.

Međutim, DAC model ima nekoliko slabosti. Jedna od njih je tranzitivnost prava čitanja, što znači da ako korisnik A dodeli pravo čitanja korisniku B, a zatim korisnik B kopira sadržaj objekta na drugo mesto ili ga deli sa korisnikom C, korisnik A možda neće biti svestan da korisnik C ima pristup objektu.

Još jedna slabost je ranjivost na napade trojanskim konjem. Kada korisnik pokrene program ili skriptu, program nasleđuje identitet korisnika koji ga je pokrenuo. To znači da zlonamerni korisnik može napisati program koji ima pristup objektima koje korisnik ne bi smeo da deli i izvršava akcije koje nisu u skladu sa namerama vlasnika objekta.

Unatoč ovim slabostima, DAC model i dalje se široko koristi u različitim sistemima zbog svoje jednostavnosti i fleksibilnosti u delegiranju prava pristupa.

## **25.Mandatory Access Control?**

Mandatory Access Control (MAC) je mehanizam kontrole pristupa koji se bazira na Bell-LaPadula modelu. U ovom modelu, korisnicima i objektima se dodeljuju bezbednosni atributi koji uključuju hijerarhijski bezbednosni nivo (clearance level) i nehijerarhijsku kategoriju.

Clearance level definiše hijerarhiju nivoa kao što su unclassified (U), confidential (C), secret (S) i top secret (TS). Kategorije su dodatni atributi koji se dodeljuju korisnicima i objektima kako bi se ograničio pristup samo određenim informacijama.

U MAC modelu se primenjuju pravila "no read up" i "no write down". Ovo znači da korisnik može pristupiti samo objektima sa istim ili nižim nivoom bezbednosti kao što je njegov. Korisnik ne sme pisati u objekte sa nižim nivoom bezbednosti od svog. Ova pravila su osmišljena da spreče napade trojanskim konjem i zloupotrebu privilegija.

MAC model je odlučiv i zahteva jasno razlikovanje između korisnika i subjekata (programa). Korisnik može imati određeni bezbednosni nivo, ali program koji koristi može imati drugačiji bezbednosni nivo. Kako bi pristupio objektima sa nižim nivoom bezbednosti, korisnik mora promeniti svoju sesiju na odgovarajući nivo.

MAC model se često primenjuje u okruženjima kao što su baze podataka i sistemi sa visokim nivoom poverljivosti. On pruža dodatni sloj bezbednosti i smanjuje rizik od neovlašćenog pristupa informacijama na osnovu strogo definisanih pravila i atributa bezbednosti.

## **26.Biba model?**

Biba model je mehanizam kontrole integriteta podataka koji se fokusira na očuvanje integriteta, dok poverljivost može biti manje važna. Koristi koncept nivoa integriteta za korisnike i objekte, gde se indikuje stepen poverenja u korisnika i osetljivost objekta na izmene. Pravila za kontrolu pristupa su inverzna u odnosu na Bell-LaPadula model: subjekt može čitati objekat samo ako je njegov bezbednosni nivo manji ili jednak nivou objekta, dok može pisati samo ako je njegov bezbednosni nivo veći ili jednak nivou objekta. Biba model je posebno koristan za sisteme sa kritičnim podacima gde je integritet od suštinske važnosti.

## **27.Clark-Wilson model?**

Clark-Wilson model je mehanizam kontrole integriteta podataka koji se fokusira na održavanje ispravnosti podataka u komercijalnim primenama. Glavni cilj je osigurati da podaci budu promenjeni samo na ispravan

način od strane autorizovanih korisnika. Model uvodi nove koncepte kao što su dobro formirane transakcije (WFT) i razdvajanje zaduženja (SoD) kako bi se osigurala konzistentnost promena u podacima. Osnovne jedinice kontrole pristupa su korisnik, transformaciona procedura (TP), podatak sa ograničenim pristupom (CDI) i procedura za proveru integriteta (IVP). Clark-Wilson model definiše devet pravila koja obezbeđuju integritet podataka, uključujući sertifikaciju transformacionih procedura, praćenje promena i autorizaciju korisnika. Model se često koristi u kontekstu baza podataka gde se tabele ne direktno pristupaju korisnicima, već samo putem uskladištenih procedura.

## 28. Kineski zid?

Politika "Kineski zid" ima osnovnu nameru sprečiti protok podataka koji mogu izazvati konflikt interesa. Ova politika se primenjuje u situacijama gde finansijski konsultanti imaju pristup privatnim podacima svojih klijenata. Kako bi se sprečile zloupotrebe, organizacije se klasifikuju u različite kategorije za konflikt interesa (COI). Svaka organizacija pripada samo jednoj COI koja sadrži bar dve organizacije koje se bave istom ili sličnom delatnošću. Prema politici "Kineski zid", konsultant ne može pristupiti privatnim podacima više od jedne organizacije iz iste COI. Ako pristupi privatnim podacima jedne organizacije iz određene COI, ne može pristupiti podacima drugih organizacija iz iste COI. Ovaj model kontrole se fokusira na operaciju čitanja podataka, dok pisanje podataka se bavi drugim modelom kao što je Brewer-Nash model.

## 29. Domain Type Enforcement?

Domain Type Enforcement (DTE) model je model kontrole pristupa koji se koristi za regulisanje pristupa subjekata (aktivnih entiteta kao što su procesi i programi) objektima (pasivni entiteti kao što su fajlovi, uređaji, delovi memorije) na osnovu njihovih dodeljenih domena i tipova. U ovom modelu, subjektima se dodeljuje domen, dok se objektima dodeljuje tip. Dozvole za pristup su vezane za kombinaciju domena i tipova i izražene su u tabelarnom obliku. Postoje dve vrste dozvola: domen-domen dozvole (DomainDomain Access Control Table - DDACT) i domen-tip dozvole (Domain-Type Access Control Table DTACT). U ćelijama ovih tabela nalaze se skupovi dodeljenih prava.

Na primer, u kontekstu fajl sistema, mogu postojati dozvole poput kreiranja (C) i gašenja (K) za domendomen interakciju, i dozvole poput čitanja (R), pisanja (W), izvršavanja (E) i pregledanja direktorijuma (T) za domen-tip interakciju. Prema DTE modelu, proces A može pokrenuti proces B samo ako postoji odgovarajuće pravo C u ćeliji tabele koja povezuje A i B. Ovaj model ima sličnosti sa Lampson modelom (matrica pristupa), ali je tabela značajno manja zbog grupisanja procesa u domene i objekata u tipove.

## 30. RBAC?

Role Based Access Control (RBAC), ili Kontrola pristupa bazirana na ulogama, je model koji se koristi za upravljanje pristupom informacijama u organizaciji. Ovaj model se zasniva na definisanju uloga koje su povezane sa određenim radnim mestima u organizaciji.

NIST-ova studija je pokazala da DAC nije najbolje prilagođen potrebama organizacija jer stvarni vlasnik podataka nije pojedinačni korisnik već sama organizacija, a takođe, diskreciona kontrola pristupa nije poželjna u mnogim slučajevima. Konvencionalni model kontrola pristupa baziran na pravima (MAC) takođe nije odgovarao potrebama organizacija, pa je bila potrebna nova paradigma koja bi omogućila kontrolu pristupa baziranu na kompetenciji.

Model koji je razvijen kao odgovor na ove potrebe naziva se Ferraiolo-Kuhn RBAC model. Ovaj model definiše tri osnovna pravila:

1. Dodela uloga : Subjekt može izvršiti transakciju samo ako mu je dodeljena odgovarajuća uloga ili je izabrao neku ulogu.
2. Autorizacija uloga : Subjekt može koristiti samo uloge za koje je autorizovan.



3. Autorizacija transakcija : Subjekt može izvršavati transakciju samo ako je ta transakcija autorizovana za aktivnu ulogu subjekta.

U RBAC modelu, uloga predstavlja skup dozvola koje korisnik može imati. Korisnicima se dodeljuje jedna ili više uloga, dok uloga nije isto što i grupa korisnika. Ova razlika je važna jer uloga predstavlja skup dozvola, dok grupa predstavlja skup korisnika.

RBAC omogućava jednostavniju administraciju u organizaciji, jer se uloge retko menjaju, dok se korisnici i dozvole mogu menjati češće. Na taj način se izbegava "kloniranje" korisnika, odnosno dupliciranje njihovih privilegija.

Administracija RBAC sistema obično je centralizovana, pri čemu se na jednom serveru definišu uloge, dozvole i korisnici. Postoje dva pristupa centralizaciji: "user pull" (korisnik se prijavljuje centralnom serveru i dobija informacije o svojim privilegijama) i "server pull" (aplikacija se autentifikuje kod centralnog servera i dobija informacije o privilegijama korisnika).

RBAC model se može prilagoditi organizacionim strukturama i odgovornostima, a implementacija se može prilagoditi specifičnim potrebama sistema.

### **31. Hijerarhijski RBAC?**

Hijerarhijski RBAC (Role Based Access Control) je proširenje osnovnog RBAC modela koje uvodi hijerarhijsku strukturu uloga. U hijerarhijskom RBAC-u, uloge su organizovane u hijerarhijske odnose, gde neke uloge imaju nadređene i podređene uloge.

U ovom modelu, svaka uloga može imati jednog ili više podređenih uloga, ali samo jednu nadređenu ulogu. Nadređena uloga ima više privilegija i odgovornosti od svojih podređenih uloga. To omogućava da se privilegije nasleđuju i prenose kroz hijerarhijsku strukturu.

Hijerarhijski RBAC omogućava fleksibilnost i efikasnost u upravljanju pristupom. Evo nekoliko ključnih karakteristika hijerarhijskog RBAC-a:

- Nasleđivanje privilegija: Podređene uloge nasleđuju privilegije nadređenih uloga.
- Prekidanje nasleđivanja: U nekim situacijama, može biti potrebno prekinuti nasleđivanje privilegija kako bi se ograničio pristup određenim resursima.
- Višestruko nasleđivanje: Hijerarhijski RBAC omogućava višestruko nasleđivanje, što znači da jedna uloga može biti podređena više nadređenih uloga.

Hijerarhijski RBAC pruža organizacijama fleksibilnost u definisanju i upravljanju ulogama, omogućavajući bolju kontrolu pristupa resursima na osnovu hijerarhijskih odnosa i nasleđivanja privilegija. Ovaj model se često koristi u organizacijama sa kompleksnim strukturama i različitim nivoima odgovornosti.

## 32. ABAC?

ABAC je skraćenica za Attribute Based Access Control (Kontrola pristupa zasnovana na atributima).

Model kontrole pristupa zasnovan na atributima (ABAC) definiše ovlašćenja koja predstavljaju uslove nad svojstvima (atributima) i subjekta i objekata.

Ključni koncepti i funkcionisanje ABAC-a:

1. Mehanizam i pravila: ABAC model kontroliše pristup objektima procenjujući pravila u odnosu na attribute entiteta (subjekat i objekat), operacija i okruženja relevantnog za zahtev. On se oslanja na procenu atributa subjekta, atributa objekta i formalnog odnosa ili pravila kontrole pristupa koji definišu dozvoljene operacije za kombinacije atributa subjekt-objekat u datom okruženju.

2. Fleksibilnost: ABAC dozvoljava neograničen broj atributa koji se kombinuju da bi se zadovoljilo bilo koje pravilo kontrole pristupa.

3. Ključni elementi modela:

- Atributi – predstavljaju karakteristike koje definišu specifične aspekte subjekta, objekta, uslova okruženja i/ili zahtevanih operacija.
- Model arhitekture – primenjuje se na polise koje vrše kontrolu pristupa.
- Model polisa – definiše ABAC polise.

### Tipovi Atributa

Atributi sadrže informacije koje ukazuju na klasu informacija koju daju atribut, ime i vrednost. Postoje tri tipa atributa:

- Atributi subjekata: Subjekt je aktivan entitet (npr. korisnik, aplikacija, proces ili uređaj). Atributi subjekta definišu identitet i karakteristike subjekta, uključujući njegov identifikator, ime, organizaciju, naziv posla. Uloga subjekta se takođe može posmatrati kao atribut.
- Atributi objekata (resursa): Objekat je pasivni entitet vezan za informacioni sistem (npr. uređaji, datoteke, zapisi, tabele). Atributi objekta se mogu iskoristiti za donošenje odluka o kontroli pristupa i često se mogu izdvojiti iz metapodataka objekta (npr. naslov, datum, autor).
- Atributi okruženja: Oni opisuju operativno, tehničko ili situaciono okruženje, ili kontekst u kojem se dešava pristup informacijama. Primeri uključuju trenutni datum i vreme ili nivo bezbednosti mreže (npr. Internet naspram intraneta).

### Model polisa

Polisa je skup pravila i odnosa koji regulišu dozvoljeno ponašanje unutar organizacije, na osnovu privilegija subjekata i načina zaštite resursa.

Pravilo polise se u opštem obliku može definisati kao Bulova funkcija atributa subjekta (s), objekta (o) i okruženja (e):  $\text{Rule:can\_access}(s,o,e) \leftarrow f(\text{ATTR}(s), \text{ATTR}(o), \text{ATTR}(e))$ . Ako je rezultat funkcije tačan, pristup resursu se odobrava, u suprotnom se odbija.

ABAC u poređenju sa RBAC

Glavna prednost ABAC modela je što eliminiše definiciju i upravljanje statičkim ulogama. Samim tim, uklanja se potreba za administrativnim zadacima ručne dodele veza korisnik-uloga i dozvola-uloga, što je obavezno kod Role-based Access Control (RBAC) modela.

## 32. SoD?

SoD je skraćenica za Separation of Duties (Razdvajanje zaduženja).

To je ključni koncept u modelima kontrole pristupa, a posebno je centralan u Clark-Wilson modelu i kao jedno od ograničenja (constraints) u Role-Based Access Control (RBAC) modelu.

Svrha i definicija

Razdvajanje zaduženja (SoD) je mehanizam čija je primarna svrha da spreči da autorizovani korisnici naprave pogrešne izmene u podacima.

Ovaj princip:

1. Osigurava konzistentnost izmena u podacima.

2. Zahteva da se kritične operacije obavljaju od strane dva ili više lica, tako da bezbednost ne zavisi isključivo od jedne osobe.

U komercijalnim primenama, Clark-Wilson model naglašava da je integritet daleko važniji od poverljivosti, a SoD je ključan za osiguravanje tog integriteta.

Primeri primene SoD-a

SoD obezbeđuje da se autorizovane izmene sprovode u skladu sa politikama:

- Bankarski sef: Otvaranje sefa zahteva učešće klijenta i službenika (primer da kritične operacije obavljaju dva ili više lica).
- Aktivacija nuklearnog oružja: Primer je pravilo "Two-man rule".
- Knjigovodstvo/Nabavka: Da bi se sprečile prevare, različite podoperacije obavljaju različite osobe. Na primer, nabavku zahteva korisnik A, odobravanje vrši korisnik B, a kontrolu (nadgledanje) vrši korisnik C.

Ako bi sve podoperacije radio isti korisnik, moguće su prevare. Čak i kada ih rade različite osobe, prevara je moguća jedino uz "zločinačko udruživanje".

SoD u RBAC modelu

SoD se implementira kao ograničenje (constraint) u naprednim RBAC varijantama (RBAC2 i RBAC3). Ograničenja se mogu postaviti na dva osnovna načina:

### 1. Statički SoD (Static Separation of Duties)

- Ograničenja se postavljaju u trenutku kada se korisniku dodeli uloga.
- Na primer, dodeljivanje jedne uloge može sprečiti dodeljivanje druge, u skladu sa SoD pravilima.
- SoD baziran na objektima: Ovo je varijanta statičkog SoD-a gde nijedna osoba ne sme imati obe kritične uloge za isti objekat (npr. isti zahtev za nabavku). (Ovo rešenje se koristi jer je nepraktično za male organizacije striktno zabraniti da ista osoba ikada ima obe uloge, pa se fokus prebacuje na zabranu konflikta nad istim objektom).

## 2. Dinamički SoD (Dynamic Separation of Duties)

- Ograničenja se postavljaju u toku sesije.
- Korisniku mogu biti statički dodeljene obe uloge, ali ih ne može imati istovremeno aktivne u jednoj sesiji.

## 3. SoD baziran na istoriji

- Korisniku su dozvoljene sve privilegije za obavljanje kritičnog zadatka, ali ne može obaviti sve delove tog zadatka nad istim objektom.

Pravila za dodeljivanje dozvola ulogama moraju biti pažljivo definisana kako bi se osigurao SoD princip, garantujući da nijedan korisnik ne može samostalno da izvrši kritičan zadatak.

pitanja sa prvog roka:

### **Razlika asimetričnih i simetričnih algoritama:**

Simetrični algoritmi koriste isti ključ za šifrovanje i dešifrovanje podataka. To znači da i pošiljalac i primalac moraju imati pristup istom ključu. Primjer simetričnog algoritma je AES (Advanced Encryption Standard). Asimetrični algoritmi koriste par ključeva: javni ključ za šifrovanje podataka i privatni ključ za dešifrovanje podataka. Javni ključ se može dijeliti s drugima, dok privatni ključ ostaje tajna. Primjer asimetričnog algoritma je RSA.

### **Hibridni pristup:**

Hibridni pristup kombinuje simetričnu i asimetričnu kriptografiju radi postizanja efikasnog i sigurnog prenosa podataka. Obično se koristi asimetrična kriptografija za razmjenu simetričnog ključa koji se zatim koristi za šifrovanje samih podataka. To omogućava brzu razmjenu ključeva uz manju računarsku složenost.

### **Kako potvrditi identitet osobe koja je poslala certifikat:**

Identitet osobe koja je poslala certifikat može se potvrditi korištenjem javnih ključeva i digitalnih potpisa. Certifikat obično sadrži informacije o identitetu vlasnika, a digitalni potpis garantuje da su te informacije autentične i nepromijenjene.

### **Kako napraviti lozinku sigurnijom:**

Da bi se lozinka učinila sigurnijom, trebalo bi razmotriti sljedeće korake:

Koristiti kombinaciju velikih i malih slova, brojeva i posebnih znakova.

Koristiti duže lozinke (najmanje 8-12 znakova).

Izbjegavati korištenje očiglednih informacija kao što su imena, datumi rođenja ili lozinke poput "password".

Redovno mijenjati lozinke i ne koristiti istu lozinku za više naloga.

Koristiti upravitelja lozinki koji će generisati i čuvati sigurne lozinke.

### **Logovi:**

Logovi su zapisi koji bilježe aktivnosti i događaje koji se dešavaju u sistemu ili aplikaciji. Oni sadrže informacije kao što su vremenska oznaka, identifikacija korisnika, vrsta događaja i druge relevantne detalje. Logovi se koriste za dijagnostiku, sigurnost, praćenje performansi i istraživanje problema u sistemu.

### **Povlačenje sertifikata:**

Povlačenje sertifikata je postupak u kojem izdavatelj sertifikata (Certification Authority - CA) povlači validnost sertifikata prije isteka datuma. Ovo se može dogoditi iz različitih razloga, kao što je otkrivanje kompromitovanog privatnog ključa, promjena identiteta vlasnika ili opoziv sertifikata zbog kršenja sigurnosnih politika.

### **Firewall:**

Firewall je sigurnosni mehanizam koji se koristi za kontrolu pristupa između računarske mreže i vanjskog svijeta. On radi na principu uspostavljanja pravila i filtriranja mrežnog prometa kako bi se spriječili

neovlašteni pristupi ili štetni napadi. Firewall može biti implementiran kao hardverski ili softverski, a koristi se za zaštitu mreže od različitih sigurnosnih prijetnji.

### **RBAC (Role-Based Access Control):**

RBAC je model upravljanja pristupom koji se koristi za kontrolu autorizacije u računarskim sistemima. Ovaj model temelji se na dodjeli uloga korisnicima, a svaka uloga ima pridružene privilegije i prava pristupa određenim resursima. RBAC omogućava fleksibilno upravljanje pristupom i olakšava administraciju sistema.

### **Dvofaktorska autentifikacija:**

Dvofaktorska autentifikacija je sigurnosni mehanizam koji zahtijeva dvije različite vrste identifikacije pri prijavi na računarski sistem ili aplikaciju. Obično se koristi kombinacija nečega što korisnik zna (npr. lozinka) i nečega što korisnik posjeduje (npr. mobilni uređaj koji generiše jednokratni kod). Ovaj pristup povećava sigurnost jer napadaču je potrebno da sazna više od jednog elementa kako bi dobio pristup.

### **Kako obezbeđujemo portove:**

Portovi se mogu obezbijediti primjenom sigurnosnih mjera kao što su:

Konfiguracija firewalla kako bi se blokirao pristup nepotrebnim portovima.

Upotreba enkripcije kako bi se osigurala privatnost i integritet podataka koji prolaze kroz portove. Redovno ažuriranje softvera i primjena sigurnosnih zakrpa kako bi se spriječile ranjivosti koje bi mogle biti iskorištene na određenim portovima.

Korištenje sigurnih protokola i autentifikacije prilikom pristupa portovima.

### **Hijerarhijski RBAC (Role-Based Access Control):**

Hijerarhijski RBAC je proširena verzija RBAC modela koji omogućava upravljanje pristupom na osnovu hijerarhije uloga. U ovom modelu, uloge su organizovane u hijerarhijsku strukturu, gdje više privilegirane uloge mogu naslijediti privilegije niže privilegiranih uloga. Ovo olakšava upravljanje pristupom i omogućava fleksibilnost u dodeljivanju privilegija korisnicima.

### **Mane dvofaktorske autentifikacije:**

Iako dvofaktorska autentifikacija pruža dodatni sloj sigurnosti, postoji nekoliko potencijalnih mana: Kompleksnost: Dvofaktorska autentifikacija može biti složena za postavljanje i korištenje, posebno za manje tehnički obučene korisnike.

Potreba za dodatnim uređajem: Korisnici moraju imati drugi uređaj (npr. mobilni telefon) kako bi primili jednokratne kodove ili provjerili identitet, što može biti nepraktično ili problematično.

Ovisnost o drugim faktorima: Ako korisnik izgubi pristup drugom faktoru autentifikacije (npr. mobilnom telefonu), može biti teško dobiti pristup računu.

### **Challenge-Response:**

Challenge-response je metoda autentifikacije u kojoj sistem postavlja izazov (challenge) korisniku, a korisnik mora pružiti ispravan odgovor (response) kako bi se autentifikovao. Izazov i odgovor mogu biti generisani na osnovu tajnog ključa ili neke druge informacije koja je poznata samo korisniku i sistemu.

### **CA (Certification Authority):**

CA je organizacija ili entitet koji izdaje digitalne certifikate. Njegova uloga je da potvrdi identitet entiteta (npr. osoba, organizacija, web stranica) i garantuje integritet podataka u certifikatu. CA također koristi digitalne potpise kako bi osigurao autentičnost certifikata. Monitoring Access Control:

Monitoring Access Control se odnosi na proces praćenja i nadzora pristupa korisnika sistemu ili resursima. To uključuje praćenje i analizu logova, nadzor pokušaja neovlašćenog pristupa, detekciju sumnjivih aktivnosti i reakciju na sigurnosne incidente. Cilj je otkriti i spriječiti neovlašćeni pristup i zloupotrebu sistema.

### Kako zaštititi server:

Zaštita servera uključuje primjenu sigurnosnih mjera kao što su:

Redovno ažuriranje i zakrpe softvera na serveru kako bi se ispravile poznate sigurnosne ranjivosti.

Konfiguracija firewalla i ograničavanje pristupa samo na potrebne usluge i portove.

Korištenje snažnih autentifikacijskih metoda poput lozinki sa snažnom kompleksnošću ili dvofaktorske autentifikacije.

Kriptovanje podataka koji se prenose između servera i klijenata kako bi se osigurala privatnost i integritet.

Redovno praćenje logova i nadzor servera kako bi se otkrile sumnjive aktivnosti ili napadi.

Ograničavanje privilegija korisnika i primjena principa najmanjih privilegija (least privilege principle) kako bi se smanjila površina napada i rizik zloupotrebe privilegija.

### Jos pitanja na osnovu discord poruka:

#### 1. Separation of Duties (SoD) u RBAC-u i Opšte (10 Pitanja)

Šta je Separation of Duties (SoD)?

**Separation of Duties (SoD)**, ili Razdvajanje zaduženja, je mehanizam čija je primarna svrha da **spreči da autorizovani korisnici naprave pogrešne ili zlonamerne izmene u podacima**. Ovaj princip osigurava konzistentnost izmena. SoD zahteva da se kritične operacije obavljaju od strane **dva ili više lica**, tako da bezbednost ne zavisi isključivo od jedne osobe.

**Primeri:** Nabavku zahteva korisnik A, odobrava korisnik B, a kontrolu vrši korisnik C. Ako bi sve podoperacije radio isti korisnik, moguće su prevare.

SoD u RBAC modelu (Ograničenja)

SoD se implementira kao **ograničenje** (Constraint) u RBAC modelu (konkretno u RBAC2 i RBAC3 varijantama). Cilj je definisati pravila za dodeljivanje dozvola ulogama tako da **nijedan korisnik ne može samostalno da izvrši kritičan zadatak**.

Postoje tri glavna tipa SoD ograničenja:

##### 1. Statički SoD (Static Separation of Duties):

- Ograničenja se postavljaju u trenutku **kada se korisniku dodeli uloga**.

- Npr. ako je korisniku dodeljena uloga A, ne sme mu biti dodeljena uloga B (međusobno isključive uloge).

- Statički SoD se definiše kao uređeni par (skup uloga,  $n$ ), gde nijedan korisnik ne može imati više od  $n$  uloga iz tog skupa (obično  $n=1$ ).

- Kada se primenjuje sa **hijerarhijom uloga**, dodeljivanje jedne uloge može sprečiti dodeljivanje druge uloge i svih njenih potomaka.

## 2. Dinamički SoD (Dynamic Separation of Duties):

- Ograničenja se postavljaju **u toku sesije** kada korisnik koristi sistem.

- Korisniku mogu biti statički dodeljene obe uloge (A i B), ali ih **ne može imati istovremeno aktivne** u jednoj sesiji/subjektu.

- Dinamički SoD se definiše kao uređeni par (skup uloga,  $n$ ) gde nijedna korisnička sesija ne može imati više od  $n$  uloga iz tog skupa.

## 3. SoD baziran na objektima (Object-Based SoD):

- Ovo je rešenje za male organizacije gde je nepraktično potpuno zabraniti da ista osoba ima dve uloge.

- Fokus je na zabrani konflikta **nad istim objektom**. Npr., nijedna osoba ne sme imati ulogu "zahtev za troškove" i ulogu "odobravanje troškova" **za isti zahtev za nabavku**.

## 4. SoD baziran na istoriji (History-Based SoD):

- Korisniku su dozvoljene sve privilegije za obavljanje kritičnog zadatka, ali **ne može obaviti sve delove tog zadatka nad istim objektom** (zasnovano na prethodnim akcijama).

## 2. Simetrični vs. Asimetrični Algoritmi (9 Pitanja)

Kriptografski algoritmi se dele na simetrične i asimetrične, zavisno od odnosa ključa za šifrovanje i ključa za dešifrovanje.

Simetrični Algoritmi (Tajni Ključ)

Kod simetričnih algoritama, ključ za dešifrovanje se može izračunati na osnovu ključa za šifrovanje, i obrnuto. Najčešće, ova dva ključa su jednaka.

Prednosti

Mane

**Brzina:** Simetrični algoritmi su **najmanje 1000 puta brži** od asimetričnih algoritama.

**Problem razmene ključeva (Key Distribution):** Pošiljalac i primalac se moraju **unapred dogovoriti o ključu** pre šifrovane komunikacije.

**Pouzdanost:** Dobri javni simetrični algoritmi su dovoljno sigurni, uz uslov da se ključ čuva u tajnosti.

**Tajnost ključa:** Ključevi se moraju distribuirati **sigurnim komunikacionim kanalom**. Ako je ključ kompromitovan, sve poruke šifrovane njime su kompromitovane.



**Manje ključeva:** Iako se za  $n$  učesnika koristi  $n(n-1)/2$  ključeva, to je i dalje efikasnije za samo šifrovanje podataka.

**Upravljanje ključevima:** Za mrežu od 100 korisnika potrebno je 4950 ključeva.

**Dužina ključa:** Uobičajena dužina ključa je 56–256 bita; preporuka od 2016. je da ključevi budu veći od 112 bita. Brute-force napad zahteva testiranje  $2n$  pokušaja.

### Asimetrični Algoritmi (Javni Ključ)

Asimetrični algoritmi koriste algoritme sa javnim ključem (public-key algorithms). Ključ za šifrovanje se razlikuje od ključa za dešifrovanje, a **ključ za dešifrovanje se ne može izračunati** (u razumnom vremenu) na osnovu ključa za šifrovanje. Ključ za šifrovanje je **javni ključ**, a ključ za dešifrovanje je **tajni (privatni) ključ**.

#### Prednosti

**Rešavanje problema razmene ključeva:** Nema problema sa razmenom tajnih ključeva, jer se tajni ključevi ne razmenjuju. Grupa učesnika može formirati javnu bazu podataka sa ključevima.

**Šifrovanje:** Bilo ko može uputiti šifrovanu poruku primaocu koristeći javni ključ, ali samo primalac može dešifrovati poruku koristeći svoj tajni ključ.

**Digitalni Potpis:** Omogućavaju realizaciju digitalnog potpisa.

#### Mane

**Sporost:** Asimetrični algoritmi su **najmanje 1000 puta sporiji** od simetričnih. Zbog toga se uglavnom ne koriste za šifrovanje velikih dokumenata, već za potpisivanje heševa.

**Ranjivost na chosen-plaintext napade:** Ako je tajna poruka jedna iz konačnog skupa  $n$  poruka, napadač može šifrovati (javnim ključem) svih  $n$  poruka i uporediti rezultat sa tajnom porukom.

**Kompleksnost ključa:** Brute-force napad zahteva izračunavanje tajnog ključa na osnovu javnog (npr. faktorisanje količnika), što je trenutno računski složeno.

### Hibridni Kriptosistemi

Zbog sporosti asimetričnih algoritama, oni se koriste prvenstveno za **razmenu ključeva za simetrične algoritme** (session key). Simetrični ključ (session key) se generiše slučajno i šifruje se javnim ključem primaoca, čime se obezbeđuje sigurna distribucija, a zatim se celokupna komunikacija odvija brzo, korišćenjem simetričnog ključa za sesiju.

## 3. Challenge-Response Protokol (8 Pitanja)

**Challenge-response** je princip na kome se zasnivaju jake šeme za autentifikaciju.

**Princip rada:** Korisnik potvrđuje svoj identitet tako što **demonstrira poznavanje tajne informacije**, pri čemu se ta tajna informacija **ne odaje sistemu**.

1. Sistem (Verifier) korisniku (Proveru) šalje podatak koji se menja tokom vremena (tzv. *challenge*), a koji može uključivati timestamp, slučajan broj (*random number*), ili sekvencijalni broj (*sequence number*).

2. Korisnik izračunava odgovor (*response*) na osnovu dobijenog *challenge* podatka i svoje tajne informacije.

3. Ako je odgovor ispravan, sistem smatra da je korisnik autentičan.

Realizacija Challenge-Response sa javnim ključem

Postupak pomoću asimetričnih algoritama (javni ključ):

1. **Sistem generiše slučajan broj** i računa njegov heš  $h(r)$ .

2. Sistem **šifruje** (i proizvoljnu poruku  $m$ ) **korisnikovim javnim ključem**  $c(r,m,pk)$ .

3. Sistem šalje korisniku poruku:  $[h(r),c(r,m,pk)]$ .

4. Korisnik **dešifruje šifrovani deo** poruke (koristeći svoj tajni ključ) i dobija  $r$ .

5. Korisnik izračunava svoj heš  $h(r)$  i poredi ga sa primljenim  $h(r)$ . Ako su jednaki, šalje **dešifrovani nazad sistemu**.

6. Sistem proverava da li je primljeni  $r$  jednak originalnom.

**Prednost:** Korišćenje javnog ključa servera za šifrovanje prenosa lozinke je jedan pristup za mitigaciju *password sniffing* napada (SSH, SSL/TLS). Samo server može da dešifruje informacije koristeći svoj privatni ključ.

**Alternativne realizacije:** Challenge-response se može realizovati i pomoću kriptosistema sa tajnim ključem (simetrični algoritam) ili korišćenjem *zero-knowledge* tehnike, kao što je Fiat-Shamir protokol. *Zero-knowledge* protokoli su dizajnirani da eliminišu mogućnost da sistemu otkriju bilo kakvu informaciju o tajni.

## 4. CA / Sertifikati i Hijerarhija (7 Pitanja)

Certificate Authority (CA) i Sertifikati

**Sertifikat (Digital Certificate)** je elektronski dokument koji vezuje **javni ključ** za identitet subjekta kome pripada. Sertifikat takođe sadrži podatke o tome ko ga je izdao (issuer), kome je izdat (subject), period važenja, i digitalni potpis izdavaoca.

**Certificate Authority (CA)** je (pravno) lice od poverenja koje **potpisuje sertifikate**. Javni ključ CA-a je opšte poznat. Celokupan sistem za kreiranje, upravljanje, distribuciju, skladištenje i opozivanje digitalnih sertifikata naziva se **Infrastruktura javnih ključeva (PKI)**.

CA Hijerarhija (Certificate Chaining)

Sertifikati se često organizuju hijerarhijski.

- **Root CA (Korenski CA):** Nalazi se na vrhu hijerarhije. **Root CA sertifikat je self-signed** (samopotpisan). Browseri sadrže ugrađene sertifikate Root CA-a kojima veruju.

- **Sub CA (Posrednički CA):** Root CA može izdati sertifikat sa naznakom da je primalac ovlašćen da izdaje dalje sertifikate.

- **Lanac Sertifikata (Certificate Chaining):** Predstavlja putanju od neposrednog CA-a pa sve do korenskog CA-a.

Sertifikat po X.509 standardu sadrži i "**CA flag**" (deo ekstenzija), koji govori da li vlasnik ima pravo da izdaje nove sertifikate, odnosno da li je i sam CA.

## 5. Role-Based Access Control (RBAC) (6 Pitanja)

Osnovni RBAC (RBAC0)

RBAC model kontroliše pristup baziran na **ulogama (roles)**, pri čemu uloga obično predstavlja radno mesto u organizaciji.

**Pet osnovnih koncepata:**

1. **Korisnici (Users)**

2. **Uloge (Roles)**

3. **Dozvole (Permissions):** Koje se sastoje iz **Operacija** dopuštenih nad **Objektima**.

4. **Korisničko-Uloga veze (User-Role Assignment):** Korisnik može imati više uloga, a uloga može imati više korisnika.

5. **Uloga-Dozvola veze (Role-Permission Assignment):** Uloga može imati više dozvola, a dozvola može biti dodeljena više uloga.

**Administrativna Ušteda:** Glavna prednost RBAC-a je pojednostavljena administracija i ušteda u broju veza. Dok direktno povezivanje korisnika i dozvola zahteva  $|U| \cdot |P|$  veza, u RBAC-u je potreban zbir veza korisnik-uloga i uloga-dozvola, odnosno  $|U| + |P|$  veza (pod uslovom da je  $|U|, |P| > 2$ ).

**Dinamička komponenta:** Korisnik može imati više uloga, ali subjekat (proces/program) može aktivirati samo **podskup** tih uloga u datoj sesiji radi ispunjenja principa minimalnih privilegija.

Hijerarhijski RBAC (RBAC1)

Hijerarhijski RBAC dodaje RBAC0 modelu **hijerarhiju uloga**.

- **Nasleđivanje dozvola:** Uloge mogu da nasleđuju dozvole od drugih uloga. Ako uloga A nasleđuje ulogu B, sve dozvole koje ima uloga B pripadaju i ulozi A. Tj., skup dozvola u B je podskup skupa dozvola u A.

- **Svrha:** Hijerarhije su prirodno sredstvo za strukturiranje uloga, odslikavajući organizaciju i raspodelu zaduženja. Iako je priprema složenija, administracija je dugoročno jednostavnija.

- **Tipovi hijerarhija:** Postoje **ograničene hijerarhije** (jednostruko nasleđivanje) i **opšte hijerarhije** (višestruko nasleđivanje), pri čemu su ograničene raširenije.

- **Tipovi uloga:** U složenim organizacijama mogu se uvesti **tipovi uloga** (npr. *kreditni savetnik*) koji imaju kvalifikatore (npr. lokacija, region) da bi se izbeglo definisanje identičnih uloga za svaku filijalu.

## 6. OAuth (OAuth 2.0) (5 Pitanja)

**OAuth 2.0** je **autorizacioni okvir** (ili protokol za delegaciju), a ne primarno protokol za autentifikaciju. On omogućava aplikacijama da pristupaju resursima u ime vlasnika resursa, a da pritom vlasnik **ne mora da deli svoje kredencijale** (lozinke) sa tom aplikacijom (klijentom).

### Karakteristike:

- **Delegacija pristupa:** Vlasnik resursa autorizuje klijentsku aplikaciju da delu prava pristupa resursu u njegovo ime.
- **Tokeni:** Za autorizaciju se koriste **access tokeni** (koji su obično JWT) umesto kredencijala. Token predstavlja delegirano pravo pristupa. Access token može biti povučen ili imati ograničeno vreme trajanja.
- **Uloge u OAuth-u:** Klijent (aplikacija), Vlasnik resursa (Resource Owner/Korisnik), Autorizacioni server (Authorization Server) i Server resursa (Resource Server).
- **Autentifikacija korisnika:** Autentifikacija se uvek dešava **između vlasnika resursa i autorizacionog servera**, pri čemu klijent ne učestvuje u tom procesu, niti dobija korisnikovu lozinku.

Razlika između OAuth i Kerberosa

Dok je **Kerberos** primarno dizajniran kao **protokol za autentifikaciju** (Single Sign-On unutar jedne domene/realm-a) koji se oslanja na simetričnu kriptografiju i tajne ključeve za distribuciju tiketa, **OAuth 2.0** je **okvir za autorizaciju/delegaciju pristupa**. OAuth 2.0 se primarno koristi u web sistemima (često REST baziranim).

| Karakteristika          | OAuth 2.0   | Kerberos   |
|-------------------------|---|--|
| <b>Primarna svrha</b>   | Delegacija autorizacije (Pristup resursima bez deljenja lozinke)                        | Autentifikacija i SSO unutar domene (realm-a)          |
| <b>Glavni mehanizam</b> | Access Tokeni (obično JWT)  | KDC (AS/TGS) i Tiketi (TGT, ST)                        |
| <b>Lozinka</b>          | Nikad se ne deli sa klijentom; koristi se za autentifikaciju kod Autorizacionog servera | Nikad se ne šalje kao otvoreni tekst                   |
| <b>Primarna okolina</b> | Web sistemi (REST bazirani API-ji)  | Korporativne/mrežne okoline (Windows/Active Directory) |

## 7. Kerberos Protokol (5 Pitanja)

**Kerberos** je protokol za autentifikaciju nastao na MIT-u 1980-tih, koji je široko rasprostranjen (npr. Windows implementacija). Omogućava **Single Sign-On (SSO)**—korisnik se prijavi samo jednom, a potom ima pristup svim resursima na mreži u skladu sa svojim pravima. Ključna prednost je da se **lozinke nikad ne šalju kao otvoreni tekst**.

**Glavne komponente Kerberos sistema:**

1. **Klijent (Client):** Korisnik ili servis koji traži pristup resursu.
2. **Server (Service Server):** Resurs kojem klijent želi da pristupi.
3. **Key Distribution Center (KDC):** Centralna tačka koja ima tri komponente:
  - **Baza principala:** Čuva evidenciju svih učesnika (principala) i njihovih tajnih ključeva.
  - **Authentication Server (AS):** Izdaje TGT kartu klijentima prilikom prijave na sistem.
  - **Ticket-Granting Server (TGS):** Izdaje ST karte namenjene pojedinim resursima.

**Tok komunikacije (Pojednostavljeno):**

1. **AS zahtev (KRB\_AS\_REQ):** Klijent šalje zahtev AS-u (šifrovan korisnikovom tajnom, što je slaba tačka).
2. **AS odgovor (KRB\_AS\_REP):** AS generiše **Ticket-Granting Ticket (TGT)** i session ključ za komunikaciju Klijent-TGS, šifrujući TGT tajnim ključem TGS-a.
3. **TGS zahtev (KRB\_TGS\_REQ):** Klijent, koristeći TGT iz keša, traži **Service Ticket (ST)** za pristup konkretnom resursu od TGS-a.
4. **TGS odgovor (KRB\_TGS\_REP):** TGS generiše ST za komunikaciju Klijent-Resurs, šifrujući ST tajnim ključem koji dele TGS i Resurs.
5. Klijent pristupa resursu koristeći ST kartu.

## 8. Discretionary Access Control (DAC) (4 Pitanja)

**Discretionary Access Control (DAC)** je model kontrole pristupa u kojem **vlasnici objekata dodeljuju prava pristupa** tim objektima. Koncept "diskrecije" znači da korisnik koji ima odgovarajuće pravo može delegirati pravo pristupa objektu drugim korisnicima.

**Osnovni mehanizmi implementacije:** Najčešći mehanizam su **Liste kontrole pristupa (ACLs)**.

**Primeri implementacije:**

- **UNIX/Linux Protection Bits:** Kontrola pristupa u fajl-sistemu pomoću zaštitnih bitova (r, w, x) definisanih za tri kategorije korisnika: Vlasnik (*Self*), Grupa (*Group*), i Ostali (*Other*).
- **Relacione baze podataka (SQL):** Koriste se naredbe **GRANT** (dodeljivanje prava) i **REVOKE** (uklanjanje prava) nad objektima kao što su tabele i pogledi, gde vlasnik ima pravo delegiranja.

Slabosti DAC modela:

1. **Tranzitivnost dodeljivanja prava čitanja:** Dodeljivanje prava čitanja je tranzitivno, što znači da Alice može dozvoliti Bobu da čita fajl, a Bob može iskopirati sadržaj u svoj fajl i dozvoliti Carol da ga čita. Alice time gubi kontrolu nad distribucijom podataka.
2. **Ranjivost na napade Trojanskog konja:** Programi nasleđuju identitet korisnika koji ih je pokrenuo. Korisnik sa visokim privilegijama (npr. Alice) može pokrenuti program (subjekat) podmetnut od napadača (npr. Boba). Taj program može čitati Alicine privatne fajlove i kopirati ih u fajl koji je dostupan Bobu, a za čije brisanje u audit logu piše da ga je izvršila Alice.

Osiguravanje lozinke (jačanje autentifikacije lozinkom)

Lozinka je najrašireniji metod autentifikacije, ali je slaba tačka podložna napadima rečnikom (*dictionary attack*).

**Načini osiguravanja/jačanja autentifikacije lozinkom:**

1. **Čuvanje heša umesto lozinke:** Server ne čuva lozinku u otvorenom tekstu, već njen heš. Prilikom prijavljivanja, izračunava se heš unete lozinke i poredi se sa skladištenim hešom.
2. **Korišćenje "salt" vrednosti:** Da bi se otežao napad rečnikom, lozinci se pre heširanja dodaje **salt** (slučajan niz znakova). Salt i heš se skladište zajedno. Na ovaj način, napadač mora heširati kompletan rečnik za svaku lozinku ponaosob, umesto da hešira rečnik samo jednom.
3. **Korišćenje jednokratnih lozinki (One-Time Passwords):** Pri svakom prijavljivanju koristi se različita lozinka. Naredna lozinka se može odrediti putem liste, sekvencijalnog ažuriranja, ili sekvencijalnog izračunavanja (npr. svaka sledeća lozinka je heš prethodne lozinke).
4. **SKEY postupak:** Koristi jednosmernu heš funkciju  $f$  i niz brojeva  $x_1, x_2, \dots, x_{100}$ , gde sistem skladišti  $x_{101} = f(x_{100})$ . Prilikom prijave, korisnik šalje  $x_{100}$ , a sistem proverava da li je  $f(x_{100})$  jednak skladištenom  $x_{101}$ . Svaki broj se koristi samo jednom, a zbog jednosmernosti funkcije  $f$ , čitanje  $x_{101}$  od strane napadača mu ne pomaže.
5. **Fizički uređaji (Multifaktorska autentifikacija):** Autentifikacija je pouzdanija ako se koristi više faktora, npr. nešto što korisnik zna (lozinka/PIN) i nešto što korisnik ima (smart kartica, token).

## 9. MAC (Mandatory Access Control) (3 Pitanja)

**Mandatory Access Control (MAC)** je višenivojski model kontrole pristupa zasnovan na **Bell-LaPadula modelu**. U MAC modelu, bezbednosni nivoi se dodeljuju i korisnicima i objektima. Ključno je da MAC model **jeste odlučiv**, jer subjekti ne mogu proizvoljno da daju prava drugima (za razliku od DAC-a).

**Bezbednosni atributi:** Sastoje se od:

1. **Hijerarhijske komponente (nivo poverljivosti):** Npr. Top Secret (TS)  $\geq$  Secret (S)  $\geq$  Confidential (C)  $\geq$  Unclassified (U).
2. **Nehijerarhijske komponente (kategorije):** Dodeljene kategorije, npr. NUCLEAR, NATO.

**Osnovna pravila (Bell-LaPadula):**

1. **No Read Up (Simple Security Property):** Subjekat može da čita samo dokumente čiji nivo poverljivosti je **manji ili jednak** njegovom nivou pristupa. (Sprečava curenje poverljivosti na niže nivoe).

2. **No Write Down (\* Property):** Subjekat može da piše samo u dokumente čiji nivo je **veći ili jednak** njegovom. (Sprečava da subjekat prenese informacije sa visokog nivoa na nizak nivo).

**Zaštita od Trojanskog konja:** *No Write Down* pravilo obezbeđuje zaštitu od trojanskog konja. Ako korisnik Alice ima nivo TS, program (trojanski konj) koji ona pokrene ne može pisati u fajlove nivoa U (koje napadač Bob može da čita), jer Alice ne sme da "piše dole".

## 10. CRL i OCSP (Validnost Sertifikata) (3 Pitanja)

**CRL (Certificate Revocation List) i OCSP (Online Certificate Status Protocol)** su mehanizmi koji se koriste za **proveru povučenosti (revokacije) digitalnih sertifikata**.

Certificate Revocation List (CRL)

CRL je lista povučenih sertifikata koje izdaje CA. Glavni nedostatak CRL-a je što liste mogu biti veoma dugačke, zahtevajući veće preuzimanje podataka i potencijalno zastarele informacije.

Online Certificate Status Protocol (OCSP)

OCSP omogućava proveru statusa sertifikata u realnom vremenu.

**Postupak provere (OCSP):**

1. Alice šalje svoj sertifikat Bobu.
2. Bob (koji želi da proveriti validnost) šalje **OCSP zahtev** koji uključuje **serijski broj** Alisinog sertifikata.
3. Sertifikaciono telo (CA) proverava svoju bazu podataka i utvrđuje status sertifikata (validan/povučeno).
4. Bob dobija odgovor, **potpisan od strane CA-a**, koji tvrdi da je Alisin sertifikat ispravan.
5. Bob proverava digitalni potpis CA-a i uspostavlja komunikaciju.

**Šta je bolje, CRL ili OCSP?** Iako izvori ne daju eksplicitnu ocenu šta je generalno "bolje", OCSP je dizajniran da reši probleme CRL-a pružajući **status u realnom vremenu** samo za traženi sertifikat, čime se izbegava preuzimanje velikih lista (CRL). Međutim, izvori napominju da browseri mogu, ali i **ne moraju proveravati validnost sertifikata preko CRL ili OCSP-a**.

## 11. Attribute Based Access Control (ABAC) (2 Pitanja)

**ABAC** je skraćenica za **Attribute Based Access Control** (Kontrola pristupa zasnovana na atributima).

Model i Funkcionisanje

ABAC definiše ovlašćenja kao **uslove nad svojstvima (atributima)** i subjekta i objekata. Kontrola pristupa se vrši procenom pravila u odnosu na attribute entiteta, operacija i okruženja relevantnog za zahtev. ABAC dozvoljava **neograničen broj atributa** koji se kombinuju da bi se zadovoljilo bilo koje pravilo kontrole pristupa.

**Pravilo Polise:** U opštem obliku, pravilo polise se definiše kao Bulova funkcija atributa subjekta ( $s$ ), objekta ( $o$ ) i okruženja ( $e$ ):  $Rule:can\_access(s,o,e) \leftarrow f(ATTR(s),ATTR(o),ATTR(e))$  Ako je rezultat funkcije tačan, pristup se odobrava.

Tipovi Atributa:

1. **Atributi subjekata:** Definišu identitet i karakteristike subjekta (aktivnog entiteta, npr. korisnika ili procesa), uključujući njegov ID, ime, organizaciju, naziv posla. Uloga subjekta se takođe može posmatrati kao atribut.
2. **Atributi objekata (resursa):** Definišu karakteristike pasivnog entiteta (npr. fajlovi, tabele). Atributi se često mogu izdvojiti iz metapodataka objekta (npr. naslov, datum, autor).
3. **Atributi okruženja:** Opisuju operativno, tehničko ili situaciono okruženje/kontekst u kojem se dešava pristup informacijama (npr. **trenutni datum i vreme** ili nivo bezbednosti mreže).

**Prednost u odnosu na RBAC:** ABAC eliminiše definiciju i upravljanje statičkim ulogama, čime se uklanja potreba za administrativnim zadacima ručne dodele veza korisnik-uloga i dozvola-uloga.

## 12. Ostala pitanja (Manja učestalost)

HTTPS (Secure HTTP)

**HTTPS** obezbeđuje sigurnu komunikaciju putem HTTP protokola tako što se HTTP oslanja na drugi protokol koji omogućava bezbednu komunikaciju, a to je **SSL/TLS (Secure Sockets Layer/Transport Layer Security)**. SSL/TLS se implementira na transportnom nivou TCP/IP steka.

**Funkcionalnost TLS-a:**

- TLS obezbeđuje usluge provere identiteta (**autentifikaciju**), **integriteta** i **poverljivosti** preko TCP protokola.
- Sastoji se od dva sloja: **Record Protocol** (koristi simetrične algoritme i heš funkcije za proveru integriteta) i **Handshake Protocol** (koristi asimetrične algoritme za autentifikaciju klijenta i servera, dogovor o algoritmima i ključevima).
- Tokom Handshake protokola, server šalje svoju identifikaciju u obliku **sertifikata** (X.509).

Digitalni Potpis

Digitalni potpis obezbeđuje:

- **Autentičnost** (potvrđuje potpisnika).
- **Nekrivotvorivost** (potvrđuje da potpisnik, a ne neko drugi, potpisao dokument).
- **Neporecivost** (potpisnik ne može kasnije poricati da je potpisao dokument).

**Realizacija pomoću asimetričnih algoritama:**

1. Alice šifrue poruku (ili njen heš) svojim tajnim ključem – time je potpisuje.



2. Bob **dešifruje poruku Alicinim javnim ključem** – time potvrđuje potpis.

**Potpisivanje pomoću Hash funkcija:** Umesto potpisivanja celog dokumenta (što je sporo kod asimetričnih algoritama), potpisuje se **heš dokumenta**. Alice izračunava heš dokumenta, šifrira heš svojim tajnim ključem (potpisuje hash), i šalje dokument i potpisani heš Bobu. Bob izračunava heš primljenog dokumenta, dešifruje primljeni heš, i ako su jednaki, potpis je ispravan.

Two-way authentication (Obostrana autentifikacija)

Two-way authentication (dvostruka autentifikacija) omogućava **obostranu proveru identiteta** (i klijenta i servera). Ovo je jedna od procedura koje definiše X.509 standard.

**Postupak (u kontekstu X.509):** Nadovezuje se na *one-way authentication* gde prover šalje informacije (slučajan broj *rand*, timestamp, potpisane podatke) verifieru. U *two-way authentication* se dodaje dodatna poruka kojom **verifier potvrđuje svoj identitet**, a ona sadrži:

- Primljeni slučajni broj *rand*.
- Novi slučajni broj *rand'*.
- Potpisane podatke koji uključuju *rand*, *rand'* i identitet proveru.

Obostrana autentifikacija (kao i one-way) se koristi na webu u okviru SSL protokola.