**Kako biste pomoću simetričnog, a kako korištenjem asimetričnih ključeva očuvali cjelovitost (ali ne i nužno povjerljivost) poruke? Jeste li u postupku sačuvali izvornost kod simetričnog? A kod asimetričnog? Objasnite**

simetricni: napraviti sazetak pa ga kriptirati kljucem, s druge strane dekriptirati taj sazetak istim kljucem, napraviti ponovo sazetak poruke i vidjeti dali su isti

asimetricni: napraviti sazetak kritpirati ga mojim privatinim kljucem, primatelj mojim javnim kljucem dektriptira sezetak pa radi svoj sazetak i gleda dali su isti

kod asimtricnog je sacuvana izvornost, a kod simetricnog moze biti bilo ko ko zna dijeljeni kljuc pa ne

**Koja je razlika izmedu key exchange protokola i key agreement protokola, koji je bolji?**

Kod key excangea se salje kljuc a kod key agreementa se kljuc nikad ne salje nego ga oba deriviraju iz dijeljenih podataka

(x^a)^b = (x^b)^a kroz kanal se salju x^a i x^b ali ne i cijeli kljuc

agreement je bolji

**Nabrojati i objasniti 4 razine zrelosti API-ja**

Razina 0 (Plain Old XML)
- Upucivanje svih zahtjeva na jedan URI/prozivoljni URIji
- URI ne identificira resurs
- Samo GET I POST http metode
- Poruke zapakirane u proizvoljnom formatu

Razina 1 (Resursi)
- URI identificira resurs – koristenje imenica
- Vise URI-ja na koje se salju zahtjevi

Razina 2 (Glagoli/metode HTTPa)
- Metoda oznacava radnju koju zelimo obaviti (GET, POST, DELETE, PUT..)
- Pravilna uporaba HTTP kodova, nisu vise svi 200 OK

Razina 3 (Hipermedijske kontrole)
- HATEOAS - poveznice koje se nalaze u dobivenom odgovoru pružaju upute o sljedećim koracima

**Koji su mehanizmi neizravne komunikacije?**

Nije potrebna sprega entiteta pružatelja usluge i (jednog ili više) entiteta korisnika
Ne mora postojati ni vremenska ni prostorna sprega
- Grupna komunikacija
- Objavi-pretplati
- Redovi poruka
- Raspodijeljena dijeljena memorija

- Prostori podataka

**Objasniti razliku između nullpotentnih i idempotetnih metoda. Razvrstati GET, HEAD, PUT, DELETE, POST u neku od tih skupina.**

Nullpotentne metode
– "sigurne" metode, ne mijenjaju stanje resursa
– GET, HEAD
Idempotentne – mijenjaju stanje resursa, ali vise istih poziva ne, takodjer sigurne
– PUT, DELETE (jer kad jednom osvjezis, novi poziv nece mijenjat resurs, isto kao samo jednom mozemo obrisati neki resurs nakon toga ce svi biti 404)
POST ne spada pod nijednu od ovih jer vise poziva stvara vise resursa.

**Objasniti svojstva semantike poziva udaljene procedure „at-most-once".**

at-most-once: nije prihvatljivo moguće višestruko izvođenje operacije (npr. novčana transakcija, doziranje lijeka …), radije prijava greške kao rezultat poziva, „ručni" postupak detekcije uzroka greške, određivanje trenutnog stanja sustava i pokušaja oporavka

**Od čega se sastoji utičnica? Objasni na primjeru.**

utičnica se sastoji od ip adrese, porta te protokola koji koristi
primjer: 192.111.0.12:8080 TCP

**Za što služe metapodaci u vanjskim reprezentacijama podataka?**

Metapodaci služe za opisivanje skupova podataka radi povećavanja vidljivosti i razumljivosti

1. **Trojka za socket**

IP adresa, port, protokol

2. **Uloga funkcije push() na strani procesa poslužitelja**

za IP adresu, port i protokol pronalazi odgovarajući socket

3. **6 tipova u JSON**

number, string, boolean, array, object, null

10. **Algoritmi sažetka (hash)** – održavaju cjelovitost (moguće i izvornost)
**Algoritmi šifriranja ključem** – održavaju povjerljivost (moguće i cjelovitost i izvornost)
**Stenografija** – osigurava povjerljivost

12. **Autentikacija** – dokazivavnje identiteta korištenjem raznih metoda (lozinke, tokeni, certifikati)
**Autorizacija** – provjera da li korisnik ima odgovarajuća prava

13.**Digitalni potpis** – pošiljetelj generira sažetak poruke i šifrira ga svojim Pk i dodaje na poruku, primatelj dešifrira sažetak Jk, generira sažetak primljene poruke i uspoređuje dvoje

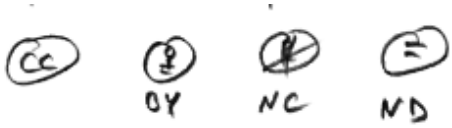
16**. Mehanizmi neizravne komunikacije**

1. Grupna komunikacija – pošiljatelj šalje grupi primatelja; vremenska sprega postoji, prostorna ne
2. Objavi pretplati – veći broj pošiljatelja i primatelja (komunikacija n:m); vremenska sprega postoji, prostorna ne
3. Redovi poruka – pošiljatelj stavlja poruku u red, primatelj čita kad želi; vremenska sprega ne postoji, prostorna da
4. Raspodijeljena dijeljena memorija – temelji se na mehanizmu međuprocesne komunikacije; vremenska sprega postoji, prostorna ne
5. Prostori podataka – posrednički entitet pruža prostor podataka, proizvođač sprema podatke u prostor, potrošač čita; vremenska sprega ne postoji, prostorna ne postoji

19**. Povlačenje podataka (pull)** - zahtjev za podacima entitetu koji ih posjeduje, priroda veze: većinom povremena (trajanje dohvata)
**Guranje podataka (push)** – od strane klijenta – veza privremena, od strane poslužitelja – veza trajnija
**Prozivanje (polling)** - klijent periodički uspostavlja vezu s poslužiteljem i provjerava dostupnost podataka, podaci se dohvaćaju povlačenjem

21. **3 zarade i poslovne prilike otvorenog koda** - Kod programa otvorenog koda ideja je naplatiti samo jednom; dodatno se naplaćuje prilagodba i dodavanje novih mogućnosti, konzultacije, edukacija, usluga kopiranja

22. **Što je**




BY(covjek) – Attribution - autoru se moraju pripisati zasluge za djelo
SA(onaj sign kao za update/refresh) – ShareAlike – ako dijelis mora biti pod istom licencom
NC(prekrizeni dollar) – NonCommercial – ne smije se koristiti u komercijalne svrhe
ND(=) – NoDerivatives – ne smije se objavljivati izmjenjeno
pd (ima I onaj CC0) – public domain, no rights reserved

23. **Slobodni programi – 4 slobode (Navedite 4 slobode otvorenog programskog koda):**

1. Sloboda pokretanja programa za sve namjere
2. Sloboda uvida u rad programa i prilagodbe potrebama
3. Sloboda širenja preslika programa
4. Sloboda poboljšanja i dijeljenja poboljšane inačice drugima

24. **Alokacija portova**- HTTP :80, HTTPS: 443, SMTP :25, POP-3: 110, IMAP: 143, FTP: 20,21, SSH: 22

28. **4 razine zrelosti API-ja**

1. Razina 0 – jedan URI preko kojeg se vrši dohvat svih podataka
2. Razina 1 – više URI-ja za odgovarajuće resurse
3. Razina 2 – odgovarajuće metode za svaki resurs
4. Razina 3 – poveznice između podataka

**Zasto se u oauth2 prvo provjerava autorizacijki kod a ne stvaraju odmah refresh i pristupni token?**
stvara se autorizacijski kod prvo zbog sigurnosti jer komunikacija izmedu vlasnika resursa i autorizacijskog poslužitelja nije sigurna pa zato prvo autorizacijski kod pa se onda s njim dohvaća pristupni token

**Sto je i za sto se koristi base64?**
Base64 je način enkodiranja podataka koji se koristi za pretvaranje binarnih podataka, poput slika, zvuka ili bilo koje druge vrste binarnih podataka, u tekstualni format. Ovaj tekstualni format se sastoji od 64 različita znaka, što uključuje slova (velika i mala), brojeve te nekoliko posebnih znakova, obično "+" i "/". Enkodiranje pomoću Base64-a omogućuje prijenos binarnih podataka kroz tekstualne kanale (HTTP zahtjevi, e-mailovi itd.) bez gubitka podataka ili problema s interpretacijom posebnih znakova.

**Objasnite verzioniranje projekta.**
Verzioniranje se radi na principu major.minor.patch, npr 1.0.2
Različite major verzije – nekompatibilna sučelja
Različite minor verzije – viša verzija samo proširuje funkcionalnost niže verzije sučelja
Različite patch verzije – nema razlika u funkcionalnosti i sučeljima, promjene svedene na ispravke grešaka

**Navedite dvije vrste vanjskog otvorenog zapisa koji je normiran i koristi se za tekst.**
JSON i XML

**Navedite barem 4 akcije koje se rade prilikom ciscenja sakupljenih otvorenih podataka prije njihove objave.**
Ujednačavanje formata datuma, brisanje duplikata i suvišnih podataka, ujednačavanje numeričkih vrijednosti i opsega podataka, provjera pravopisa…

**Sto je zastupnik a sto posrednik u raspodijeljenim sustavima?**
Zastupnik – Korisnik želi komunicirati s udaljenim entitetom koji se nalazi u nekoj udaljenoj mreži, tada zastupnik koji se nalazi u lokalnoj mreži sa korisnikom preuzima na sebe stvarnu komunikacije s udaljenim entitetom, a korisnik svu komunikaciju obavlja samo sa zastupnikom
Posrednik – nalazi se između korisnika i pružatelja usluge, sadrži popis usluga i njihovih svojstava koje oglašavaju pružatelji usluga, a potencijalni korisnici traže usluge odgovarajućih svojstava kod posrednika

**Navedite 3 poboljšanja koja openId donosi u odnosu na oauth2.**
Profil openId, id token sa osnovnim informacijama o korisniku i Userinfo krajnjoj točki.

**Neke druge bitne stvari (nisu iz pitanja):**

EXACTLY-ONCE
Pri pozivu procedure s klijenta ona se na posluzitelju izvrsi samo jednom, nemoguce
MAYBE
Pri pozivu procedure na strani klijenta nema retransmisije zahtjeva, I na posluzitelju nema filtriranja dupliciranih zahtjeva (jer znamo da nece biti vise od jednog), nema ponovnog izvodjenja/retransmisije Ako se dogode greske, izgubi se poruka u dolasku ili izgubi povratna vrijednost whoops nis od toga.
AT-LEAST-ONCE
Postoji retransmisija zahtjeva, ali nema filtriranja dupliciranih zahtjeva na posluzitelju, zato se koristi samo za idempotentne zahtjeve, ima ponovno izvodjenje.
AT-MOST-ONCE
Postoji retransmisija na klijentu I filtriranje dupliciranih zahtjeva I retransmisija poruka na posluzitelju. Posluzitelj pamti rezultat I ne izvodi ponovno operaciju ako ne treba.

**Debeli klijent (fat client**): sadrži slojeve prezentacije i aplikacijske logike zahtijeva veću snagu obrade računala domaćina i veću količinu podataka prenošenih mrežom
**Tanki klijent (thin client):** sadrži samo sloj prezentacije manja snaga obrade, manja količina prenošenih podataka