**Messec**

Messec umožňuje real-time komunikáciu medzi 2 klientmi cez bezpečný kanál (šifrovaná komunikácia). Pre inicializáciu spojenia medzi klientmi sú obaja pripojený na server.

**Server**

-protokol spojenia - server ustanoví session-key, pošle (pomocou PKI) symetrický kľúč klientovi

-ostatná komunikácia je šifrovaná symetricky (AES-256)

-databáza [Login, hash hesla]

-kontajner aktuálne prihlásených užívateľov [Login overený heslom, IP adresa, session key]

-funkcionalita:

-registrácia nového užívateľa(Login, heslo) => uloží do databázy (Login, hash hesla)

-prihlásenie zaregistrovaného užívatela(login, heslo, IP) => pridá do kontajnera online klientov

-na vyžiadanie pošle klientovi zoznam Loginov všetkých online užívateľov

-odpovie na klientov dotaz, či je „sample-login“ online

-klient1-klient2 komunikácia:

-k1 pošle dotaz na komunikáciu s k2, server pošle k2 požiadavku (ak je online)

-k2 prijme -server vygeneruje symetrický kľúč pre ich komunikáciu

-pošle kľúč a IP adresu (druhého klienta) obom

-k1 sa stane hostom, k2 sa naň pripojí

-komunikácia prebieha mimo server (AES-256)

-k2 odmietne (nechce komunikovať / je obsadený) – server pošle odpoveď k1

-zašifruj správu / súbor

-dešifruj správu / súbor

PKI - server na začiatku vytvára RSA-4096 pár kľúčov (posiela každému klientovi verejný kľúč)

-dodatočne identifikácia užívateľov cez asymetrickú kryptografiu

**Klient**

-uchováva session keys pre server a klientov, s ktorými komunikuje

-funkcionalita:

- pripojenie na server [IP]

-registrácia[Login, heslo]

-prihlásenie[Login, heslo]

-pošle na server dotaz na všetkých online klientov

-pošle na server dotaz či je „sample-login“ online

-nadviaž komunikáciu s klientom (pošle na server)

-pripoj sa na klienta (iniciátor komunikácie je host, prijímatel sa pripojí)

-komunikácia – pošli /prijmi správu

-zašifruj správu / súbor

-dešifruj správu / súbor

**Protokoly**

Na konci každej správy je pripojený jej HMAC, ktorý sa vždy skontroluje pri prijatí správy ak nesedí

odosiela sa znova posledná odoslaná správa ako výzva pre opätovné zaslanie správy.

**Zahájenie komunikácie klient – server:**

0. Pri spustení (server by mal bežať večne) servera si server vygeneruje RSA-4096 kľúč, ktorý overí CA,

ktorú zatiaľ neimplementujeme.

1. Klient sa napojí na server, ktorý mu zašle verejný kľúč.

2. Klient vygeneruje 256 bitový kľúč a inicializačný vektor, zašifruje vereným kľúčom a odošle serveru.

3. Server dešifruje a pošle klientovi výzvu k ďalšiemu kroku.

-- -- -ďalej je všetko už šifrované symetricky-- -- -

**Registrácia klienta**

1. Klient odošle nejakú sekvenciu bitov signalizujúcu, že sa chce registrovať a svoj login a heslo.

2. Server si spočíta hash (heslo + sol) a do databázy uloží login-hash-sol.

3. Server odošle správu s oznámením či to prebehlo úspešne(login mohol byť už zabraný). Ak nie

opakuje sa krok 1.

4. Po registrácií klient nie je prihlásený

**Login klienta**

1. Klient odošle svoj login a heslo.

2. Server si spočíta hash overí či prislúcha danému loginu.

3. Server odošle správu s oznámením či to prebehlo úspešne. Ak nie opakuje sa krok 1.

**Nadviazanie komunikácie s druhým klientom**

**1. Pôvodná verzia**

*1. A vytvorí server, pošle serveru login B a údaje potrebné pre pripojenie na A*

*2. Server odošle potrebné údaje B-čku a spýta sa či je ochotný komunikovať s A*

*3. Ak B príjme, server vygeneruje kľúč pre ich komunikáciu a obom ho pošle. Ak nepríjme server prepošle odmietnutie A-čku.*

*4. B sa, po tom čo dostane kľúč, pokúša napojiť na A cez TLS*

*5. Ak sa pripojí B prebehne Diffie-Hellman pomocou klúča od serveru (tie čísla u sa ešte zašifrujú tým kľúčom od servera)*

*6. Ak je ustanovená komunikácia cez bezpečný kanál A a B sa musia autentizovať tým, že ukážu že sú schopné použiť klúč, ktorý bol vytvorený za pomoci zdielaného tajomstva (klúča od serveru). A pošle číslo n Bčku. B pošle Ačku n + 1 a A pošle Bčku n + 2. Takto vedia, že sú schopný použiť klúče a komunikácia pokračuje. Ak nie, spojenie sa preruší a pokúšajú sa spojiť znova*

**2. Návrh na novú verziu**

1. Každý klient má stále vytvorený server a zverejnené číslo portu na ktorom očákáva nové spojenie.

2. A pošle serveru požiadavku na komunikáciu s B. Server vytvorí správu **Eka**(údaje potrebné na pripojenie k B **||** číslo **||** Ekb (číslo, login áčka, možno ip áčka(to asi netreba))).

3. A sa za pomoci údajov zo server napojí na B a prebhne klasický Diffie-Hellman

4. A pošle Ekb (číslo, login áčka, ip áčka) Bčku, to si to odšifruje a tým vie s kým komunikuje. Pretože do tej správy mohol ten login vložiť jedine server, ktorý pozná klúč kb.

5. B pošle Ačku naspeť číslo a teraz A vie, že je to naozaj B, pretože dokázalo použiť klúč kb.

**Logout klienta**

Klient je odhlásený automaticky ak nepošle v priebehu minúty žiadne dáta serveru. Server vždy pri prihlásení/odhlásení klienta aktualizuje zoznam on-line klientov u každého klienta a tak komunikujúci

klienti vedia, ak sa klient, s ktorým kounikujú odhlásil/padol