Biztonságos és privát digitális kommunikáció, decentralizált hálózatokon keresztül

Rostás Raul

Rostás Raul
Felkészítő tanárok: Hegedűs Tamás Frigyes
TSzC Széchenyi István Gazdasági és Informatikai Technikum,
2800 Tatabánya, Dózsa György út 58.

1. Bevezetés

A digitális kommunikáció jelentősen fejlődött az elmúlt években. Ma már bárkivel azonnal kommunikálhatunk a világ bármely pontján. Sajnos egyre több, ilyen szolgáltatásokat nyújtó platformon folyik digitális megfigyelés. Elsődleges célunk egy biztonságos és privát decentralizált hálózat kifejlesztése, amely azonnali üzenetküldésre használható, és amelyet nem cenzúrázhat vagy felügyelhet semmilyen szervezet vagy kormányzat.

2. A digitális kommunikáció jelenlegi helyzete

A digitális kommunikáció jelenlegi állapota pozitív és negatív is egyben. Egyrészt azonnal kommunikálhatunk bárkivel a világ bármely pontján. Másrészt egyre több platform digitális megfigyelésnek van kitéve.

Ennek a megfigyelésnek számos formája van. Néha egy kormányzati szerv követel hozzáférést a felhasználói adatokhoz. Máskor rosszindulatú kiberbűnözők törnek be a rendszerekbe, hogy információkat lopjanak. Néha pedig maga a platform adja el a felhasználók adatait hirdetőknek, vállalati haszonszerzés céljából.

Bármilyen formában is jelentkezik, a digitális megfigyelés komoly problémát jelent. Megsérti a magánéletünket, megakadályozza, hogy szabadon kifejezzük magunkat, és akár manipulálásra is felhasználható.

3. Centralizált vs. decentralizált számítógépes hálózatok

A kommunikációs platformok nagy része teljesen központosított. Ez azt jelenti, hogy az üzeneteket vagy más média formákat a platform szerverére küldik feldolgozásra.

Nyilvánvaló, hogy Ön nem fér hozzá ehhez a szerverhez, és nem tudja, hogy mit csinálnak az adatokkal a színfalak mögött. Bár a központosított hálózatokat sokkal egyszerűbb használni, a bizalmas adatok cseréje tekintetében mindenképpen hatalmas biztonsági kockázatot jelentenek.

A decentralizált hálózatok (más néven Peer-to-Peer hálózatok) olyan számítógépes hálózatok, ahol minden felhasználó kapcsolatban áll a másik felhasználóval, akiben teljes mértékben megbízik. A decentralizált hálózatokban nincs egyetlen hibapont sem, és minden felhasználó ugyanolyan

jogosultságokkal rendelkezik, mint a többiek. A felhasználók ellenőrzését általában kriptográfiai funkciók sorozata végzi.

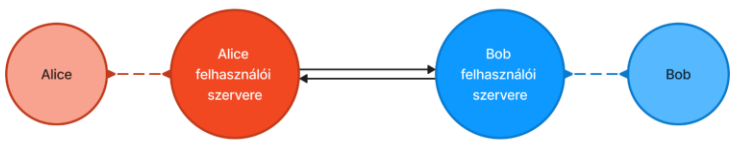
4. Probléma megoldásának menete

Mi egy decentralizált hálózat kiépítését választottuk, mivel a központosított hálózatokkal ellentétben, egy valóban privát kommunikációs platform létrehozására használható.

4.1. A hálózat felépítése

Hálózatunk 2 típusú csomópontból áll: *felhasználókból* és *felhasználói szerverekből*. (A valódi felhasználó és a *felhasználó* csomópont megkülönböztetése érdekében a csomópontok *dőlt* betűvel vannak leírva)

A *felhasználók* olyan csomópontok a hálózatban, amelyek a saját *felhasználói szerverükkel* kommunikálnak, és soha nem kommunikálnak külső *felhasználói szerverekkel*. A *felhasználói szerverek* feladata, hogy a *felhasználó* helyett külső *felhasználói szerverekre* továbbítsák a kéréseket.



1. ábra: Egy egyszerű kétszemélyes hálózat

4.2. Felhasználói szerverek közötti kommunikáció

A felhasználók hálózaton keresztül történő összekapcsolása érdekében elsődlegesen a *felhasználói szerverek* közötti kommunikáció valósul meg. A hálózatban minden egyes *felhasználói szerver* egyszerre viselkedik kliensként és kiszolgálóként, ugyanúgy, ahogyan egy központosított hálózatban is elvárható lenne a kettő viselkedése.

A fő kommunikáció a *felhasználói szerverek* között az üzenetek küldése/fogadása, a *felhasználó* üzenet történetének szinkronizálása, valamint kulcspárok lekérése egy külső *felhasználói szerver* hitelességének ellenőrzésére.

4.3. Felhasználók és felhasználói szerverek közötti kommunikáció

A felhasználó által indított műveletek esetében a *felhasználó* a saját RESTful *felhasználói szerverének* küld egy felkérést. Ezenkívűl egy valós idejű *RFC 6455* websocket kapcsolat jön létre a *felhasználó* és a *felhasználói szerver* között,

ahol a *felhasználói szerver* minden új eseményről vagy üzenetről publikál értesítést a *felhasználó* számára.

4.4. Az első üzenetek váltása

Példánkban 2 felhasználó létezik: *Alice* és *Bob*. Alice üzenetet szeretne küldeni Bob számára, és tudja Bob felhasználónevét és *felhasználói szerverének* címét: *bob@bob.hu*. Alice a saját felhasználói szerverén létrehoz egy szobát, és meghívja Bobot. Bob *felhasználói szervere* értesül, hogy egy szobába került Bob. Lekéri a meghívó *felhasználói szervertől* a szükséges szoba információt, és hozzáadja Bobot a szobához. Ezután publikálja a meghívóhoz, hogy Bob becsatlakozott a szobába. Ilyenkor Alice és Bob is tud üzeneteket váltani.

4.5. Egy felhasználó hitelességének ellenőrzése

A fenti példát használva, Alice üzenetet küld a szobába: “Helló, Bob!”, továbbá a következő adatokat: a *feladó* (*alice.hu*), a *címzett* (*bob.hu*), a *kezdeményező* (*alice@alice.hu*) , az esemény *ISO-8601 időbélyegzője,* az előző üzenet történet *hash*-je (az időbélyegzőhöz képest), és az *üzenet aláírása*, amelyet Alice magánkulcsával írt alá.

Amikor Bob *felhasználói szervere* megkapja az üzenetet, akkor:

- Meggyőződik róla, hogy a *feladó* egy érvényes *felhasználói szerver*.
- Meggyőződik róla, hogy az említett szoba lokálisan létezik, és hogy *Bob* tagja annak.
- Ellenőrzi az üzenet *aláírását* a *feladó* segítségével.
- Ellenőrzi, hogy az üzenet történetnek *hash*-je helyes-e az *időbélyeg* használatával.
- Ha a fenti feltételek mindegyike teljesül, az üzenet elküldésre kerül Bob részére, és Alice *felhasználói szervere* egy visszaigazolást kap.

4.6. Kiegészítő biztonság titkosítással

Ha Alice vagy Bob úgy dönt, hogy engedélyezik a titkosítást, akkor egy közös kulcs kiszámításra kerül a “Diffie-Hellman” kulcs megállapodásos protokoll segítségével. Ha a protokoll sikertelen, a titkosítás nem lesz engedélyezve. Ellenkező esetben a jövőbeni üzenetek a két fél által kiszámított közös kulcs használatával titkosítva lesznek. A titkosítás kikapcsolása a sikeres engedélyezés után nem lehetséges biztonsági okokból.

5. Elért eredmények

Készítettünk egy működő szerver és kliens applikációt, ahol a felhasználók biztonságos és privát módon kommunikálhatnak.

A szerver alkalmazás képes a felhasználói fiókok és kulcspárok kezelésére, üzenetek és fájlok küldésére, több eszköz között megosztott

üzenetek titkosításra és teljesen személyre szabható felhasználói profilok készítésére.

A kliens alkalmazás egy letisztult és intuitív felhasználói felületet biztosít, amit használni egyszerű és hatékony. Az üzenetek tartalmazhatnak Markdown-t. Az alkalmazás teljesen testreszabható. Saját “témák” létrehozása lehetséges CSS/SCSS használatával. Saját “bővítmények” pedig JavaScript/TypeScript használatával.