Biztonsági Protokollok
Házi feladat

Alsalti Akram (VKXAFI), Erdélyi Ádám (X61BI9), Kovács Dávid (EE8LMN)

Absztrakt:

Ez a dokumentum egy biztonságos chat applikációt ír le. A kommunikációban részt vesznek a beszélgetni kívánó felek, valamint egy szerver, amin keresztül ők kommunikálnak. A leírásban megtalálható az üzenetek titkosításának menete, a kulcsok cseréje a résztvevő felek között, valamint egy támadó modell is.

Funkcionális követelmény:

A chat applikációnak képesnek kell lennie támogatnia 2-nél nagyobb csoportokat a biztonságos kommunikációban. A felhasználók kedvük szerint le és felcsatlakozhatnak az üzenetváltásra. Az applikáció gondoskodik a kommunikáció biztonsági kihívásairól, de a felhasználó a saját publikus és privát kulcsának generálásáról, és hitelesítéséről maga gondoskodik.

Biztonsági követelmény:

Az üzeneteket a felek között továbbító szerver nem lehet képes az üzenetek dekódolására, és az üzeneteknek a végpontok között titkosítva kell lennie. Az üzeneteket védeni kell az újraküldések, módosítások, és más által küldések ellen is. Nem hivatott személyek nem csatlakozhatnak a beszélgetéshez.

Támadó modell:

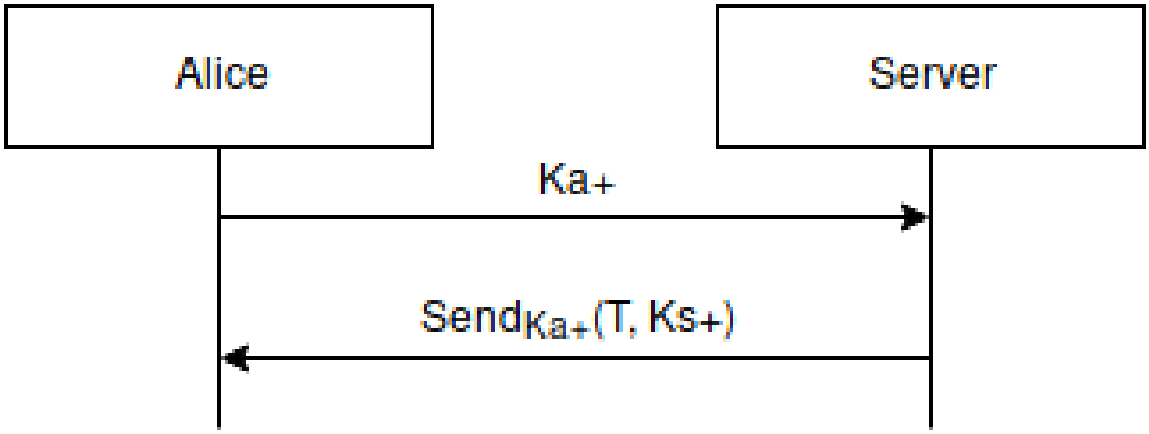
A támadóról azt feltételezzük, hogy a csatornát képes lehallgatni, ott üzeneteket elkapni módosítani és újraküldeni. A szerver címét ismeri, így ő is elkezdheti a bekapcsolódási protokollt.

A protokoll:

A kommunikációban a legfontosabb jelenleg talán a szerver, amin keresztül minden forgalom halad. A szerveren chat szobák találhatóak, a felhasználók ezekhez tudnak csatlakozni. Minden kliens csatlakozásánál a publikus kulcsú tanusítványt leellenőrzi a szerver, hogy az adott beszélgetéshez a kliens csatlakozhat-e (whitelist) és csak kívánt résztvevő esetén kezdődik a protokoll. Minden legalább egy csatlakozással rendelkező chat szobához tartozik egy K kulcs, amit a legrégebben csatlakzott kliens generál és küld el a cél-kliens publikus kulccsával titkosítva a később csatlakozó klienseknek, ezen felül mind a kliensek, mind a szerver renelkezik egy-egy publikus és privát kulcssal, melyeket elsősorban a csatlakozáskor hasznosítanak.

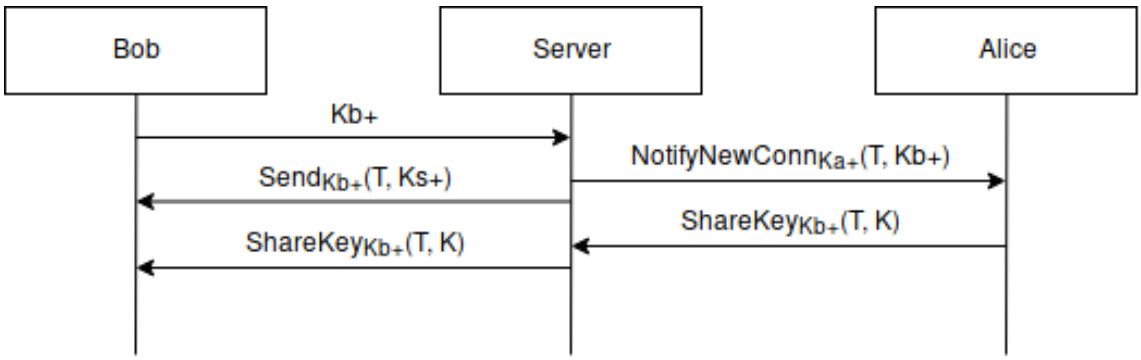
Elsőként csatlakozó kliens esetén a kliens elküldi a publikus kulcsát a szervernek, ami erre elküldi a saját publikus kulcsát. Ezek után a kliens generál egy K kulcsot, amit az egész szoba használni fog, de a szerver mégsem ismeri. Ez a folyamat látható az 1.1-es ábrán. Ezen felül a kliensek és a szerver is minden kiküldött üzenetet digitálisan aláír, hogy a másik fél meggyőződhessen arról, hogy valóban azzal kommunikál, akivel

szeretne. Az üzeneteknek a titkosítandó szöveg mellett része még egy timestamp, hogy az üzenet-visszajátszást elkerüljük, ezt is ugyanúgy titkosított és aláírt formában küldjük el, gyakorlatilag az üzenet részeként.



1.1 ábra – Első csatlakozó egy chatszobához

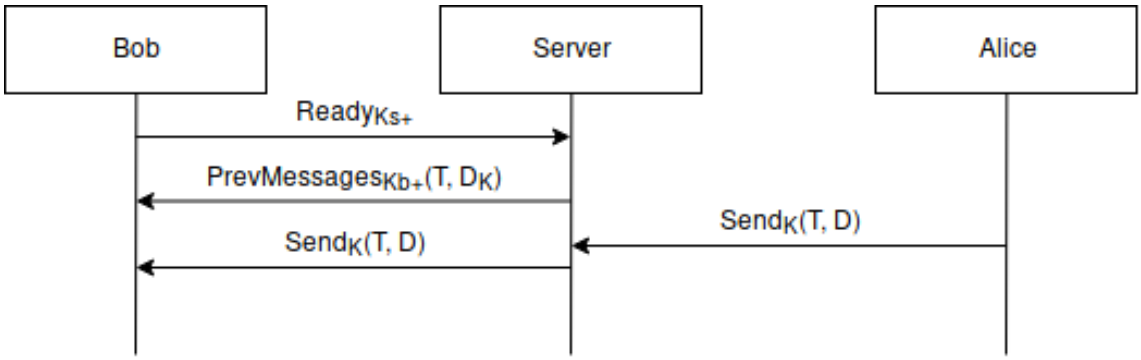
Az 1.1 ábrán egyébként megfigyelhető az is, hogy már az első lehetőségnél titkosított, időbélyegezett és digitálisan aláírt üzenetet küld a szerver, hogy tudjuk, hogy nem csalóval állunk szemben.



1.2 ábra – Csatlakozás egy chatszobához

Ahogy látszik az 1.2-es ábrán Alice már ott van a szobában és legenerálta a chatszoba közös kulcsát, közben amint a szerver megkapja az új csatlakozó Bob nyilvános kulcsát, azt egyből továbbküldi Alice-nak, mivel ő maga nem ismeri a beszélgetés titkosításához használt kulcsot.

Ezután Alice Bob publikus kulcsával titkosítja a chatszobában használt kulcsot és elküldi Bob-nak a szerveren keresztül, hogy ő is részt vehessen kommunikációban. A szerver ezt sem tudja dekódolni, hiszen ahhoz Bob privát kulcsa kéne.



1.3 ábra – Kommunikáció

Az 1.3 ábrán látható kommunikáció egy pár feltevéssel kiegészítendő, ezek a következők:

- Alice már korábban csatlakozott
- Bob most fejezte be az 1.2 ábrán látható csatlakozási folyamatot

Ezekkel egyhangban Bob, mint frissen csatlakozott kliens jelzi a szerver számára, hogy készen áll a részvételre a beszélgetsében, ezt az üzenetet a szerver publikus kulcsával titkosított formában küldi, erre a szerver elküldi Bob számára a chatszoba eddigi tartalmát, tehát a beszélgetési előzményeket, szintén titkosítva. Ezeket a szerver nyugodtan tárolhatja, hiszen ő maga és külső fél sem tudja feloldani dekódolni az eddigi üzeneteket.

Ezután már csak annyi a szerver dolga, hogy szinkronban tartsa a feleket, úgy, ahogy az ábrán is látszik, ahol Alice küld egy üzenetet a chat szobába és a szerver ezt továbbküldi Bob számára. A szerver minden üzenetet tárol chatszobánként, a K kulccsal titkosított formában.

A lecsatlakozás a szerverről szintén nem okoz komoly gondolt (olyan szinten, hogy ábra sem készült róla), ha az egyik kliens kilép a többi zavartalanul tudja folytatni a kommunikációt, hiszen a kulcsot mind ismerik. Az egyetlen különleges eset az, ha minden kliens kilép, ekkor az egész folyamat az elejéről kezdődik egy új kliens csatlakozásakor, az elmentett beszélgetések a szerverről pedig törölhetőek, hiszen legközelebb nem lesz azonos a beszélgetéshez használt kulcs, a szerver pedig nem tudja elolvasni az üzeneteket, tehát csak a helyet foglalják.

A lecsatlakozásnál érdekes lehet még a kulcsot az újonnan csatlakozóknak tovább adó kliens választása, abban az esetben, ha az eddig ezt a feladatot betöltő kliens lett volna az, amelyik kilépett. Ez a kiválasztás egyszerű, a szerver kiválasztja erre a célra a jelenleg online kliensek közül a legrégebben csatlakozottat.

Kulcsok rövid leírása:

Publikus és privát kulcsok

> Ezek az RSA szabályainak megfelelő kulcsok, minden node-nak van egy publikus és egy privát kulcsa is. Ezek a kulcsok használatosak a digitális aláírásokhoz is.

Üzenetekhez használt K kulcs

> Egyelőre ehhez az AES CBC módjára gondoltunk, amihez szükséges egy IV (kezdeti vektor). Azt, hogy az egyforma üzenetek kiszámíthatóak legyenek úgy kerüljük el, hogy minden üzenetnél cseréljük az IV-t. Ezt véletlenszám generátor segítségével oldjuk meg. Ezeken felül fontos még, hogy az üzenetek egyofrma hosszúak legyenek, ha ez mégsem lenne így akkor azzá tesszük őket padding használatával.

Timestamp

> A timestampekhez egy TTP-TSA-t választunk (Trusted Third Party-Time Stamping Authority), minden időbélyeget ez a megbízható fél fog generálni.

Összegzés:

A fentebb leírt protokoll kielégíti a feltételezett támadó modell elleni védelmet, hisz az újrajátszás ellen az időbélyeg, a módosítás ellen, és a küldő hitelesítésére a digitális aláírás véd. Az üzenet a végpontok között végig a K kulcsal titkosítva, a szerver

által sem dekódolhatóan utaznak. A nemkívánt résztvevőket a szerver a whitelist alapján még a kulcsok megszerzése előtt ki tudja szűrni.

Rövidítések feloldása:

- K – Chatszobához tartozó kulcs
- Ka+ - Alice publikus kulcsa
- Ka- - Alice privát kulcsa
- Kb+ - Bob publikus kulcsa
- Kb- - Bob privát kulcsa
- Ks+ - A szerver publikus kulcsa
- Ks- - A szerver privát kulcsa
- T – időbélyeg (timestamp)
- D – Küldött adat (Data)