МИРЗО УЛУҒБЕК НОМИДАГИ
ЎЗБЕКИСТОН МИЛЛИЙ УНИВЕРСИТЕТИ
МЕХАНИКА МАТЕМАТИКА
ФАКУЛЬТЕТИ

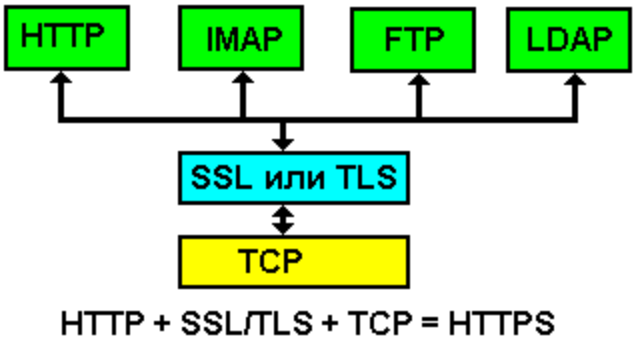
РЕФЕРАТ

Мавзу: **Тармоқ архитектураси.**

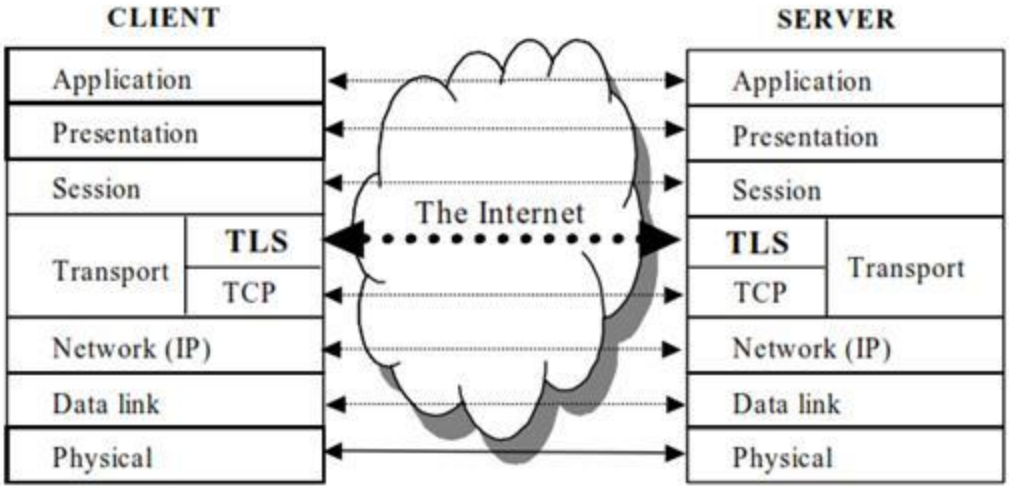
Reja

1. **Simsiz tarmoqlar xavfsizligi protokollari.**
2. **Detallardagi TLS bog`lanishlarni tasdiqlash protseduralari.**
3. **TLS dagi bog`lanishlarni oddiy tasdiqlash.**

Har xil kompyuterlar va turli dasturlar tarmoq aloqasi jarayonida bir-birlarini tushunish uchun maxsus texnik qoidalar qo`llaniladi. Tarmoq sohasida bunday qoidalar to`plami protokol (bayonnoma) deb ataladi.



HTTP + SSL/TLS + TCP = HTTPS



TLS (Transport Layer Security) ning protokol konfiguratsiyasi o‘zaro xavfsiz harakatlanishga mo‘ljallangan. U shifrlash programmalarini, sertifikatlar formatini, raqamli imzo algoritmini va aloqa seansi jurnalidan foydalanishni o‘z ichiga oladi. Konfiguratsiya yana transport darajasida qayta qabul qilish – uzatishdan aloqani himoyalash uchun TLS ni tunnellash uslubini ham aniqlaydi.

TLS spetsifikatsiyasi tarmoq sathida konfidensiallik va yaxlitlik vositalarining to‘liq to‘plamini tavsiflangan holda, mubolag`asiz fundamental ahamiyatga ega. TLS asosida yuqoriroq sath (tatbiqiy sathga qadar) protokollarini himoyalash mexanizmi hamda xavfsizlikning tugallangan vositalari, xususan virtual xususiy tarmoqlar quriladi. Albatta TLS kriptografik mexanizmlariga va kalit infratuzilmalariga tayanadi.

TLS spetsifikatsiyasi turli vazifalarni bajaruvchi ko‘pgina dasturiy maxsulotlarda ishlatiluvchi ommaviy Secure Socket Layer (SSL) protokolini rivojlantiradi va oydinlashtiradi.Tunnelning initsiatori va terminatori orasida uzatiluvchi axborotni shifrlash transport sathi TLS(Transport Layer Security) yordamida amalga oshiriladi. Tarmoqlararo ekran orqali autentifikatsiyalangan o‘tishni standartlash uchun SOCKS deb ataluvchi protokol aniqlangan va hozirda

SOCKS protokolining 5-versiyasi kanal vositachilarini standart amalga

oshirilishida ishlatiladi.

1999 - yili SSL 3.0 versiyasi o‘rniga, SSL protokoliga asoslangan va hozirda Internet standarti hisoblangan TLS protokoli keldi. SSL 3.0 va TLS protokollari orasidagi farq juda ham jiddiy emas.

SSL va TLS protokollarining kamchiligi - o‘zlarining xabarlarini tashishda tarmoq sathidagi faqat bitta — IP-protokolidan foydalanishlari va faqat IP-tarmoqlarda ishlay olishlari. Undan tashqari, SSL/TLSning amalda qo‘llanishi tatbiqiy protokollar uchun to‘la shaffof emas. SSLning yana bir salbiy tomoni shunday iboratki, agar mijoz va server ulanishni uzsalar, ular uni ma’lumotlarning minimal hajmini almashish yo‘li bilan tiklashlari va Session ID ning eski parametrlaridan foydalanishlari mumkin. G`araz niyatli shaxs oldingi sessiyalardan birini obro‘sizlantirib uni tiklash muolajasini server bilan o‘tkazishi mumkin. Natijada bu sessiyada uzatiladigan keyingi barcha ma’lumotlar obro‘sizlantiriladi. Undan tashqari, SSLda autentifikatsiyalashda va shirflashda bir xil kalitdan foydalaniladi. Bu esa ma’lum bir holatlarda zaiflikka olib kelishi mumkin. Bunday yechim turli kalitlar ishlatilganiga nisbatan ko‘p statistik ma’lumotlarni yig`ishga imkon beradi.

Simsiz tarmoqlar xavfsizligi protokollari. Himoyalangan ulanishlar protokoli — Secure Sockets Layer (SSL) Internet brauzerlarining xavfsizligi muammosini yechish uchun yaratilgan. SSL taklif etgan birinchi brauzer — Netscape Navigator tijorat tranzaksiyalari uchun Internet tarmog`ini xavfsiz qildi, natijada ma’lumotlarni uzatish uchun xavfsiz kanal paydo bo‘ldi. SSL protokoli shaffof, ya’ni ma’lumotlar tayinlangan joyga shifrlash va ras-shifrovka qilish jarayonida o‘zgarmasdan keladi. SHu sababli, SSL ko‘pgina ilovalar uchun ishlatilishi mumkin.

SSL o‘zidan keyingi TLS (Transport Layer Security - transport sathi himoyasi protokoli) bilan Internetda keng tarqalgan xavfsizlik protokolidir. Netscape kompaniyasi tomonidan 1994 yili tatbiq etilgan SSL/TLS hozirda har bir brauzerga va elektron pochtaning ko‘pgina dasturlariga o‘rnatiladi. SSL/TLS

xavfsizlikning boshqa protokollari, masalan, Private Communication Technology (PCT — xususiy kommunikatsiya texnologiyasi), Secure Transport Layer Protocol (STLP-xavfsiz sathning transport protokoli) va Wireless Transport Layer Security (WTLS — simsiz muhitda transport sathini himoyalash protokoli) uchun asos vazifasini o‘tadi.

SSL/TLSning asosiy vazifasi tarmoq trafigini yoki gipermatnni uzatish protokoli HTTPni himoyalashdir. SSL/TLS aloqa jarayonining asosida yotadi. Oddiy HTTP-kommunikatsiyalarda TCP-ulanish o‘rnatiladi, xujjat xususida so‘rov yuboriladi, so‘ngra xujjatning o‘zi yuboriladi.

SSL/TLS ulanishlarni autentifikatsiyalash va shifrlash uchun ishlatiladi. Bu jarayonlarda simmetrik va asimmetrik algoritmlarga asoslangan turli texnologiya -

lar kombinatsiyalari ishtirok etadi. SSL/TLSda mijozni va serverni identifikatsiyalash mavjud, ammo aksariyat hollarda server autentifikatsiyalanadi.

SSL/TLS turli tarmoq kommunikatsiyalar xavfsizligini ta’minlashi mumkin. Protokolning juda keng tarqalishi elektron pochta, yangiliklar, Telnet va FTP (File Transfer Protocol — fayllarni uzatish protokoli) kabi mashxur TCP-kommunikatsiyalar bilan bog`liq. Aksariyat hollarda SSL/TLS yordamida kommunikatsiya uchun alohida portlar ishlatiladi.

TLS va uning oldingi versiyasi SSL kriptografik protokollar bo`lib, internet tarmog`ida uzellar aro berilganlarni uzatishda himoyalashni ta’minlaydi. TLS va SSL autentifikatsiyalash uchun asimmetrik kriptografiyadan ishonchlilikni ta’minlash uchun ularni autentifik kodlaridan foydalanadi.

Berilgan protokol internet tarmog`i bilan ishlashda ilovalardan keng foydalanadi. Masalan, veb – brovzerlarda elektron pochtalar bilan ishlashda, IP – telefonlarda va ko`p ma’lumot almashishda foydalaniladi.

TLS - protokoli Netscape Communications kompaniyasi tomonidan ishlab chiqilgan SSL 3.0 versiyasi spesifikatsiyasiga asoslangan. Hozirda IETF bilan

shug`ullanuvchi TLS standarti rivojlangan. So`ngi paytlarda esa protokol RFC 5246 (Август 2008) va RFC 6176 (Март 2011) ga yangilangan.

TLS tarmoqlarda aloqalarni amalga oshiruvchi mijoz – server ilovalarining amalga oshirish imkoniyatlarini beradi. Bunda sanktsiyasiz murojatlar va eshitishlar oldi olinadi.

Ko`pgina protokollarda TLS (yoki SSL) siz bog`liqliklardan foydalanish mumkin, bunday hollarda mijoz TLS ni o`rnatishni hohlasa o`rnatish paytida serverga ko`rsatib qo`yishi kerak. Bu shunga olib kelishi mumkin-ki yoki unifikatsiyalangan portning nomeridan foydalanish yordamida barcha bog`lanishlarni o`rnatish mumkin yoki ixtiyoriy portdan foydalanish bilan va mijoz tomonidan serverga uzatilgan maxsus buyruqlar yordamida, protokolni maxsus mexanizmlardan foydalanish bilan TLS protokolida bog`lanishlarni o`rnatish mumkin. Mijoz va server TLS dan foydalanishni kelishilgan holda amalga oshiriladi va ular himoyalangan bog`lanishlar o`rnatish kerak. Bu tasdiqlangan bog`lanishlarining protseduralari yordamida amalga oshiriladi. Bu jarayon vaqtida mijoz va server xavfsizlik bog`lanishlarni o`rnatish uchun zarur bo`lgan turli xil kelishilgan parametrlarni qabul qiladi.

Bog`lanishlarning himoyalangan seansini yaratish protseduralarining asosiy qadamlari quyidagilar:

- Mijoz TLS ni qo`llab quvvatlovchi serverni yoqadi va himoyalangan bog`lanishlarni so`raydi;
- Mijoz shifrlash algoritmlari va xesh funksiyalarini himoyalovchi ro`yhatni ko`rsatadi;
- Server mijoz ko`rsatgan ro`yhatdan serverni himoya qiluvchi algoritmlarni tanlaydi;

- Server asl autentifikatsiya uchun mijozga raqamli sertfikat jo`natadi. Oddiy raqamli sertfikat server nomi, sertfikatsiyani tasdiqlash markazining nomi va serverning ochiq kalitidan iborat;

- Mijoz sertfikatsiya markazining server bilan bog`lanishi mumkin va ma`lumotlarni uzatish boshlanguncha uzatilgan sertfikatning haqiqiyligini tasdiqlaydi;

- Himoyalangan bog`lanishlar uchun seans kalitini generatsiyalashda mijoz – server ochiq kalitning tasodifiy generatsiyalangan raqamli ketma – ketliklarni shifrlaydi va natijalarni serverga uzatadi. Bog`lanishlarni o`rnatish uchun asimmetrik shifrlash algoritmlarining spesifikatsiyasidan foydalaniladi. Bunda server o`zining yopiq kalitidan foydalanib olingan ketma – ketlikni deshifrlaydi.

Bu bilan aloqalarni tasdiqlash protsedurasi yakunlanadi. Mijoz va server orasida bog`lanish havfsizligi o`rnatiladi, aloqa tugamaguncha jo`natilgan ma`lumotlar shifrlanadi va shifrlash kalitidan foydalanib deshifrlanadi.

Bog`lanishlar o`rnatilamaganda va bog`lanishlarni tasdiqlashning yuqorida ko`rsatilgan qadamlarida xatoliklar payda bo`ladi.

Havfsizlik. Protokolning oldingi versiyalarni pasayishini yoki shifrlash algoritmini ishonchlikini kamayishinidan himoyalash

- Ilovalardagi ketma – ket yozuvlarni nomerlash va ma`lumotlarni autintifikatsiyasi (MAC) kodidagi tartib nomeridan foydalanish.

- Ma`lumotlarning identifikatoridagi kalitdan foydalanish ( faqat egasigina ma`lum bo`lgan kalit orqali ma`lumotlar autintifikatsiyasi kodini tekshirish mumkin) xesh – ma`lumotlarning identifikatsiya kodi bo`lib, RFC 2104 da aniqlangan TLS shifrlar to`plamidagi ko`pgina shifrlarda foydalaniladi.

- Barcha ma`lumotlar xeshlardan iborat bo`lib, bog`lanishlarni tasdiqlash bilan yakunlanadi bunda bog`lanishlarni tasdiqlash davomida tomonlar alamashadi.

- Psivdotasodifiy funksiyalari 2 qismga bo`linadi va har biri turli xesh funksiyalarda qayta ishlanadi, shuningdek ma`lumotlarning autintifikatsiyasilarini yaratishda olingan o`ramlar uchun XOR hisoblanadi.

Bu esa xesh funksiyalardan birida bo`ladigan zaifliklar himoyasini taminlaydi. TLS 1.0 protokolining zaifligi 2011- yil sentabrda Ekoparty konferensiyasida nazariy jihatdan ko`rsatildi. Namoyish foydalanuvchilarning autintifikatsiyasi uchun ishlatiladigan cookieslar deshifrini o`z ichiga oladi.

2009 – yil avgustda topilgan bog`lanishlar fazasidagi zaifliklarni bartaraf qilishda kriptoanalitika https bog`lanishlarni buzishda ma`lumotlarga maxsus so`rovlar qo`shishga mijoz - server bog`lanishlarga imkon beradi. Kriptoanalitikada mijoz va serverning o`zaro habarlarini deshifrlash mumkin emas. Bunday tipdagi hujum g`araz niyatli shaxs tipidagi standart hujumdan farq qiladi.

Shunday holatlar bo`ladiki foydalanuvchi brozerlar indeksiga e`tibor bermasa ham sessiya havfsiz hisoblanadi. Bunda g`araz niyatli shaxs tipidagi hujumlar uchun zaiflikdan foydalanish mumkin. Bunday zaiflikni o`rnatish uchun mijoz tomonidan taklif bo`lishi kerak va server tomon keyingi bog`lanishlar haqidagi axborotni qo`shish kerak va qayta yangilanuvchi bog`lanishlarda tekshiruv amalga oshiriladi. Bu RFC 5746 standartida OpenSSL ning so`ngi versiyasida va boshqa kutubxonalarda ko`rsatilgan. Hujumlarning mavjud varianlari protokollarning programma realizatsiyasiga bevosita asoslangan.

**Detallardagi TLS bog`lanishlarni tasdiqlash protseduralari.** TLS protokoliga muvofiq ilovalar yozuvlarga inkapsulyatsiyalangan ma`lumotlarga almashinadi va ular uzatilishi kerak. Har bir yozuvlar bog`lanishlarning joriy holatiga bog`liq holda ma`lumotlarning autintifikatsiyasi kodining identifikatsiyalangani yoki shifrlangani, qo`shimchalar va siqilgan ma`lumotlar

bo`lishi mumkin. TLS dagi har bir yozuv quyidagi maydonlardan iborat: content type (yozuv tarkibidagi tipni aniqlash) maydon paketning ko`rsatilgan uzunligi va TLS ning ko`rsatilgan versiyasi.

Qachonki bog`lanish o`rnatilganda o`zaro harakatlar content type, handshake, TLS protokoli bo`yicha ketadi.

**TLS dagi bog`lanishlarni oddiy tasdiqlash.** Quyida bog`lanishlarning o`rnatishning oddiy misolini ko`ramiz, bunda server mijozning sertfikati bo`yicha autintifikatsiyalashga o`tadi.

1. So`zlashuvlar fazasi.

    - Mijoz TLS protokolini qo`llab quvvatlovchi so`ngi versiyaga ClientHello ma`lumotini, shifrlash metodlarni himoyalovchi ro`yhatni va tasodifiy sonlarni jo`natadi;
    - Server ServerHello ma`lumotini beradi va uni ma`lumotlar tarkibida server tanlagan protokolning versiyasi tasodifiy sonlar mos shifrlash algoritmlari mijozlarning murojaatlaridan iborat ;
    - Server Certificate ma`lumotini jo`natadi bu ma`lumot tarkibida serverning raqamli sertfikati bo`ladi;
    - Server ServerHelloDone ma`lumotini jo`natadi bu ma`lumot bog`lanishlarning tasdiqlashning identifikatsiyalab tugatilishi hisoblanadi.
    - Mijoz ClientKeyExchage ma`lumoti bilan javob beradi, bunda ma`lumot PreMasterSecret (bu PreMasterSecret server sertfikatni ochiq kalit yordamida shifrlash) ochiq kalitidan iborat bo`ladi yoki hech narsa bo`lmasligi mumkin;

- Mijoz va server PreMasterSecret kalitidan foydalanib sonlarni tasodifiy generatsiyalab umumiy maxfiy kalitni hisoblaydi. Kalitlar haqidagi qolgan barcha ma`lumotlar umumiy maxfiy kalitda bo`ladi.

2. Mijoz ChangeCipherSpec ma`lumotini jo`natadi bunda keyingi barcha ma'lumotlarda umumiy maxfiy kalitdan foydalanib, o`rnatilgan bog`lanishlarni tasdiqlash algoritmlari yordamida shifrlanadi. Yozuvlar darajasidagi bu ma`lumotlar 20 - tipga ega lekin 22- tipda emas.

    - Mijoz Finished ma`lumotini jo`natadi bu ma`lumot bog`lanishlarni tasdiqlash protseduralari ma`lumotlari asosida generatsiyalangan MAC va xesh dan iborat;

    - Server Finished ma`lumotini oladi va MAC va xeshlarni tekshiradi. Agar deshifrlash jarayoni yoki tekshirish yaxshi bo`lmasa bog`lanishlarni tasdiqlash foydasiz hisoblanadi va bog`lanishlar uzilishi kerak.

3. Server ChangeCipherSpec va Finished shifrlangan ma`lumotni jo`natadi va mijoz o`z navbatida tekshirish va deshifrlashni bajaradi.

Bu vaqtda bog`lanishlarni tasdiqlash o`rnatilgan protokolda yakunlangan hisoblanadi. Barcha keyingi paketlar 23 – tipda olib boriladi, barcha berilganlar shifrlangan bo`ladi.

Mijoz autintifikatsiyasi bilan bog`lanishlarni tasdiqlash. Quyidagi misolda mijoz va server o`rtasida sertfikatlarni almashish orqali mijozlarning to`liq autintifikatsiyasini ko`ramiz:

1. So`zlashuvlar fazasi.

- Mijoz TLS protokolini qo`llab quvvatlovchi so`ngi versiyaga ClientHello ma`lumotini, shifrlash metodlarni himoyalovchi ro`yhatni va tasodifiy sonlarni jo`natadi;

    - Server ServerHello ma`lumotini beradi va uni ma`lumotlar tarkibida server tanlagan protokolning versiyasi tasodifiy sonlar mos shifrlash algoritmlari mijozlarning murojaatlaridan iborat ;

    - Server Certificate ma`lumotini jo`natadi bu ma`lumot tarkibida serverning raqamli sertfikati bo`ladi;

    - Server CertificateRequest ma`lumotini jo`natadi bu ma`lumot tekshirish uchun mijoz sertfikatlarining so`rovlaridan iborat;

    - Server ServerHelloDone ma`lumotini jo`natadi bu ma`lumot bog`lanishlarning tasdiqlashning identifikatsiyalab tugatilishi hisoblanadi.

    - Mijoz ClientKeyExchage ma`lumoti bilan javob beradi, bunda ma`lumot PreMasterSecret (bu PreMasterSecret server sertfikatni ochiq kalit yordamida shifrlash) ochiq kalitidan iborat bo`ladi yoki hech narsa bo`lmasligi mumkin;

    - Mijoz va server PreMasterSecret kalitidan foydalanib sonlarni tasodifiy generatsiyalab umumiy maxfiy kalitni hisoblaydi. Kalitlar haqidagi qolgan barcha ma`lumotlar umumiy maxfiy kalitda bo`ladi.

2. Mijoz ChangeCipherSpec ma`lumotini jo`natadi bunda keyingi barcha ma’lumotlarda umumiy maxfiy kalitdan foydalanib, o`rnatilgan bog`lanishlarni tasdiqlash algoritmlari yordamida

shifrlanadi. Yozuvlar darajasidagi bu ma`lumotlar 20 - tipga ega lekin 22- tipda emas.

- Mijoz Finished ma`lumotini jo`natadi bu ma`lumot bog`lanishlarni tasdiqlash protseduralari ma`lumotlari asosida generatsiyalangan MAC va xesh dan iborat;

- Server Finished ma`lumotini oladi va MAC va xeshlarni tekshiradi. Agar deshifrlash jarayoni yoki tekshirish yaxshi bo`lmasa bog`lanishlarni tasdiqlash foydasiz hisoblanadi va bog`lanishlar uzilishi kerak.

3. Server ChangeCipherSpec va Finished shifrlangan ma`lumotni jo`natadi va mijoz o`z navbatida tekshirish va deshifrlashni bajaradi.

**TLS bog`lanishlarni qayta tuzish.** Seans kalitlarni generatsiyalashda qo`llaniladigan asimmetrik shifrlash kalitlari hisoblash nuqtai nazaridan qimmat hisoblanadi. Bundan qutilish uchun bog`lanishlarni qayta tuzishda ular takrorlanadi. TLS bog`lanishlarni qayta tuzishda foydalaniluvchi maxsus yorliq yaratadi. Bog`lanishlarni tasdiqlashda mijoz ClientHello ma`lumotiga oldingi sessiyaning identifikatori session id ni qo`shadi. Mijoz TCP- porti va serverning IP adresi orqali session id identifikatori bilan bog`lanadi bunda server oldingi bog`lanishlarning barcha parametrlaridan foydalanishi mumkin. Server bog`lanish parametrlari orqali oldingi sessiyaning identifikatorini taqqoslaydi, bunda master secret va shifrlash algoritmidan foydalaniladi. Bir tomondan master secret yagona bo`lishi kerak aks holda bog`lanishlar bo`lmaydi. Bu kriptoanalitiklarning sanksiyasiz murojatlarini olishi uchun session id dan foydalanishning oldini oladi. ServerHello va ClientHello ma`lumotlardagi tasodifiy sonli ketma – ketliklar kafolatlashga imkon beradi ya`ni oldingi bog`lanishlardagi seans kalit bilan generatsiyalangan seans kalitning farqini ajratishga. RFC da bog`lanishlarning tasdiqlashning bunday tipi qisqartirish deyiladi.

1. So`zlashuvlar fazasi.

- Mijoz TLS protokolini qo`llab quvvatlovchi so`ngi versiyaga ClientHello ma`lumotini, shifrlash metodlarni himoyalovchi ro`yhatni va tasodifiy sonlarni jo`natadi; Shuningdek ma`lumotga oldingi bog`lanishlarning session id identifikatori qo`shiladi.

- Server ServerHello ma`lumotini beradi va uni ma`lumotlar tarkibida server tanlagan protokolning versiyasi tasodifiy sonlar mos shifrlash algoritmlari mijozlarning murojaatlaridan iborat; Agar server session id identifikatorini bilsa, u holda ServerHello ma’lumotiga session id identifikatorini qo`shadi. Bu mijozlar uchun signallar hisoblanib oldinga sessiyalarni qayta tuzishda foydalanish mumkin. Agar server session id identifikatorini bilmasa boshqa o`rindagi qiymatni qo`shadi. Bunday holda mijoz bog`lanishlarni qayta tuza olmaydi. Mijoz va server yagona master secret va seans kalitlarni generatsiyalovchi tasodifiy sonlarga ega bo`lishi kerak.

2. Mijoz ChangeCipherSpec ma`lumotini jo`natadi bunda keyingi barcha ma’lumotlarda umumiy maxfiy kalitdan foydalanib, o`rnatilgan bog`lanishlarni tasdiqlash algoritmlari yordamida shifrlanadi. Yozuvlar darajasidagi bu ma`lumotlar 20 - tipga ega lekin 22- tipda emas.

    - Mijoz Finished ma`lumotini jo`natadi bu ma`lumot bog`lanishlarni tasdiqlash protseduralari ma`lumotlari asosida generatsiyalangan MAC va xesh dan iborat;

    - Server Finished ma`lumotini oladi va MAC va xeshlarni tekshiradi. Agar deshifrlash jarayoni yoki tekshirish

yaxshi bo`lmasa bog`lanishlarni tasdiqlash foydasiz hisoblanadi va bog`lanishlar uzilishi kerak.

3. Mijoz ChangeCipherSpec ma`lumotini jo`natadi bunda keyingi barcha ma’lumotlar umumiy maxfiy kalitdan foydalanib o`rnatilgan bog`lanishlarni tasdiqlash algoritmlari orqali shifrlanadi.

    - Mijoz o`zining Finished shifrlangan ma`lumotini jo`natadi;

    - Server bu ma`lumotni deshifrlab qabul qiladi va xesh MAC larni tekshiradi.

4. Bu vaqtda o`rnatilgan protokoldagi bog`lanishlarni tasdiqlash yakunlangan hisoblanadi. Keyingi barcha paketlar 23- tipda bo`ladi shuningdek barcha shifrlangan berilganlar ham.

Ishlab chiqaruvchanlik nuqtai nazaridan yagona kirishlarni realizatsiyalashda bog`lanishlarni qayta tuzish algoritmlaridan foydalanish mumkin. Bu FTPS protokolini realizatsiyalash uchun muhim qiymatga ega aks holda g`araz niyatli shaxs tipidagi hujumlarning zaifligi bo`ladi.

Sessiyalar mandati. RFC 5077 bog`lanishlar identifikatori o`rniga sessiyalar mandatidan foydalanish orqali TLS ni kengaytiradi. U oldingi sessiyalarning identifikatorini talab qilmaydi TLS seanslarni qayta tuzish usulini aniqlaydi uning holati TLS serverda saqlanadi. Mandatlar sessiyasidan foydalanishda TLS server mandati seansida holatni saqlaydi va TLS mijozga saqlash uchun jo`natadi. Mijoz serverga mandatlar seansini jo`natishda TLS ni qayta tuzadi, server esa qabul qilingan mandatdagi aniq bir sessiyalarning parametrlariga muvofiq TLS ni qayta tuzadi. Sessiyalar mandati serverdagi autentifikatsiya ko`rinishda shifrlanadi va server mandatdan foydalanish ta`minlanganligini tekshiradi. Bu metodning kamchiliklaridan biri uzatiladigan

mandatlar sessiyasida autentifikatsiyasi va shifrlash uchun faqat AES128-CBC-SHA256 metodidan foydalanish TLS parametrlarga bog`liq bo`lmagan holda. Bunda TLS sessiyalardagi ma`lumotlar yaxshi himoya qilinmaydi. Asosiy muammo kanteks ilovalardagi OPENSSL kalitini saqlashga chaqirishdir. Bu esa AES128-CBC-SHA256 ga takror kirishga imkon bermaydi.