HTTP PROTOKOLI SARLAVHASIDA AXBOROT BERKITISH ALGORITMI

Mavlonov Obid Nizomovich
Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti

Annotatsiya. Ushbu maqolada HTTP (HTTPS) protokolining yashirin kanallari va ularda axborot berkitish masalalari ko‘rib chiqilgan. Axborotlarni yashirish maqsadida taklif etilgan algoritmda maxfiy ma’lumotlar HTTP so‘rov-paketining sarlavha maydonlariga joylashtiriladi hamda asimmetrik shifrlash algoritmlaridan foydalangan holda shifrlanadi. Shuningdek, uning xavfsizligi ushbu shifrlash algoritmlarining xavfsizligiga bog‘liqligi keltirilgan.

Tayanch iboralar. HTTP, HTTPS, qatlam, yashirin soha, shifrlash, axborotni berkitish.

Аннотация. В данной статье рассмотрены скрытые каналы протокола HTTP (HTTPS) и вопросы сокрытия информации в них. В предлагаемом алгоритме сокрытия информации конфиденциальные данные помещаются в поля заголовка пакета HTTP-запроса и шифруются с использованием алгоритмов асимметричного шифрования. Также указано, что его безопасность зависит от безопасности этих алгоритмов шифрования.

Ключевые слова. HTTP, HTTPS, слой, скрытая область, шифрование, сокрытие данных.

Abstract. In this article are discussed the hidden channels of the HTTP (HTTPS) protocol and the issues of hiding information in them. In the proposed algorithm to hide information, confidential data is placed in the header fields of the HTTP request packet and encrypted using asymmetric encryption algorithms. Also it is given that its security depends on the security of these encryption algorithms. In the proposed algorithm, the data is sent to the hidden fields and is done using asymmetric encryption algorithms (certificates). In this case, the receiver generates the keys and stores the private key. The sender encrypts the traffic with a public key.

Keywords. HTTP, HTTPS, layer, hidden area, encryption, data hiding.

KIRISH

HTTP gipermatnli internet protokoli mijoz va server o‘rtasidagi aloqani standartlashtirishga qaratilgan bo‘lib, TCP/IP protokol stekiga asoslangan OSI modeli amaliy sathidagi tarmoq protokoli sanaladi. U kontent tuzilishi, so‘rovi va internet orqali uzatilishini belgilaydi. Odatda TCP/IP protokollari 80-port va boshqa shu kabi portlardan ham foydalanish mumkin. Ammo, HTTPS va xavfsiz HTTPS protokollari 443-portdan foydalanadi.

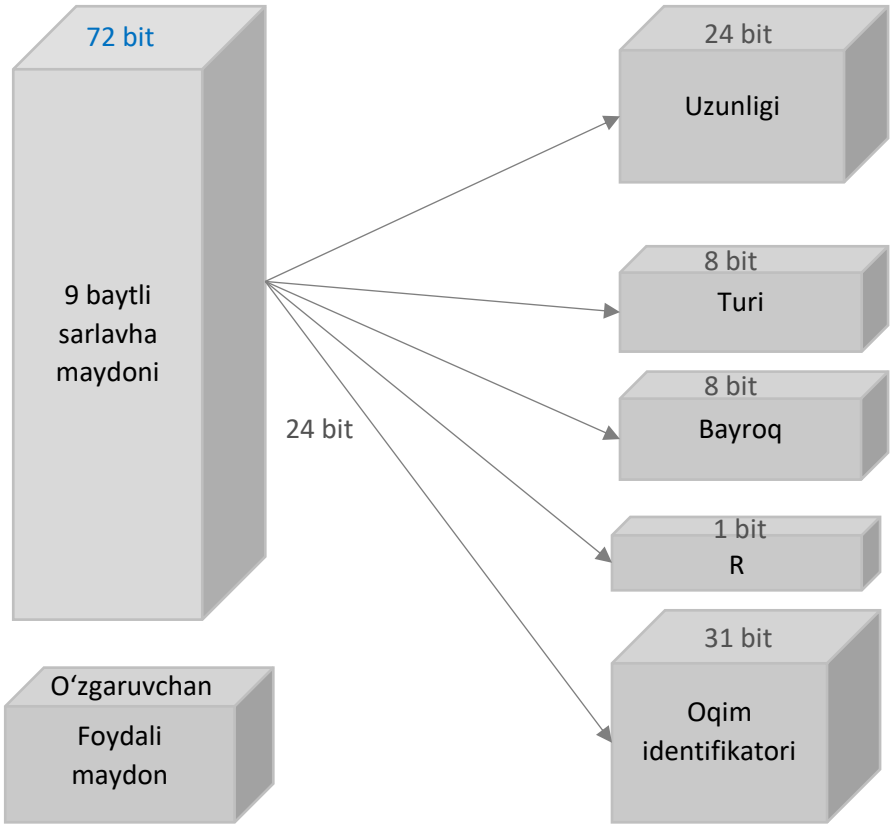
HTTP ning birinchi hujjatlashtirilgan versiyasi 1991-yilda ishlab chiqilgan va uni HTTP/0.9 kabi belgilash kirtilgan. Bu protokol GET deb nomlangan yagona so‘rov usuli asosida ishlagan. Ya’ni, server so‘rovni oladi va HTML so‘rovlari asosida javob beradi va kontent uzatilishi bilanoq ulanish yopiladi [1].

1996-yilda HTTP ning navbatdagi versiyasi, ya’ni HTTP/1.0 paydo bo‘ldi, u oldingi versiyaga nisbatan ancha yaxshilangan. HTTP/0.9 versiyasidan farqli ravishda boshqa javob formatlarini, tasvirlar, video fayllar, oddiy matn yoki boshqa kontent turi bilan amalga oshirgan.

ASOSIY QISM

HTTP ulanishi mustaqil ikki tomonlama oqimlarni o‘tkazishi mumkin, bu yerda ko‘plab oqimlar parallel ravishda xabar almashishi mumkin. Har bir xabar oxirgi nuqtaga yuboriladigan kichikroq freymlarga bo‘linadi. HTTP orqali uzatiladigan har bir kadr oqim bilan bog‘lanadi hamda barcha oqimlar yagona va boshqa oqimlar tomonidan foydalana olmaydigan oqim identifikatorlariga biriktiriladi. Ruxsat etilgan 9-baytli sarlavha va freym turiga bog‘liq bo‘lgan o‘zgaruvchan uzunlikdagi foydali yukdan iborat 10 xil turdagi freymlar mavjud. Har bir freym sarlavha va foydali axborotdan iborat. Har bir freym sarlavhasi qo‘shimcha ravishda quyidagi maydonlarni o‘z ichiga oladi [2]:

24-bit freymning foydali qismi;
8-bit freym turi;
8-bitli bayroqlar;
1-bitli zaxira maydon (R);
31-bit oqim identifikatori.
Quyidagi rasmda sarlavha va uning tarkibiy qismlari keltirilgan.



10-rasm. HTTP protokolining sarlavhasi va uning tarkibiy qismlari.

Uning boshqa protokollardan farqli jihatlaridan biri sifatida, URL satri va HTTP sarlavhasi yoki HTTP xabar tanasining o‘lchamlari bo‘yicha cheklovlar yo‘qligini keltirish mumkin.

Odatda turli xil ilovalarda turli cheklovlar kiritiladi. Masalan, Apache serverlari versiyaga qarab hajmi HTTP sarlavhalarini 8 KB yoki 16 KB gacha qabul qiladi.

HTTP protokolini axborotlarni berkitish nuqtai-nazaridan qaraladigan bo‘lsa, unda ko‘plab imkoniyatlar mavjud. Quyida undagi mavjud yashirin kanallar hosil qilish yo‘llari va imkoniyatlari keltirilgan [3].

Padding yordamida yashirin kanal hosil qilish. DATA, HEADERS va PUSH PROMISE freymlarida xabarlar hajmini yashirish maqsadida foydalaniladi.

Oqim identifikatorlaridan foydalangan holda yashirin kanal hosil qilish. Oqim identifikatori belgisiz 31 bitli butun son bilan taqdim etiladi. 0×0 qiymati ulanishni boshqarish xabarlari uchun, 0×1 qiymati esa HTTP so‘rovi uchun ajratilgan.

PING freymidan foydalangan holda yashirin kanal hosil qilish. PING freymlari xar ikki tomon, mijoz va serverdan yuborilishi mumkin hamda ular faqat oqim identifikatori 0×0 bilan bog‘lanadi. Ular bo‘sh ulanish ishlayotganligi va aylanishni o‘lchash uchun ishlatiladi.

Oqimning ustuvorligi va bog‘liqliklaridan foydalanadigan yashirin kanallar hosil qilish. HTTP boshqa oqimlardan ustuvorlikka ega va uni foydalanuvchining o‘zi tanlashi mumkin. Yuborish imkoniyati cheklangan bo‘lsa, jo‘natuvchi ustuvorliklar asosida kadrni uzatish uchun oqimni tanlaydi.

Turli sonli o‘ziga xos turdagi freymlardan foydalanadigan yashirin kanallar hosil qilish. Bitta sarlavha freym, undan keyin nol yoki keyingi freymlar blokini o‘z ichiga oladi. Agar, freym yagona bo‘lsa, keyingisi sifatida axborot yashiringan freymni jo‘natish yoki ko‘p freymli qatorlarda ularning ichiga kiritish orqali erishish mumkin.

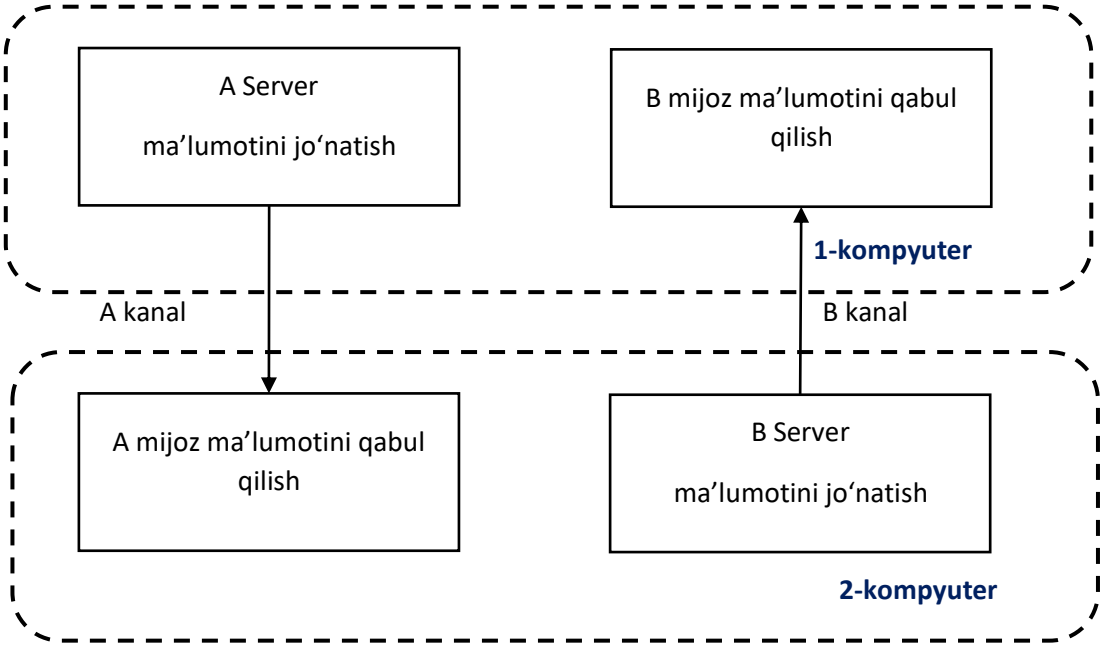
Cookie sarlavhasi maydonidan foydalangan holda yashirin kanal hosil qilish. HTTP cookie juftlarni bir cookie sarlavhasi maydonidan har birida bir yoki bir nechta cookie-juftlarga ega bo‘lgan bir nechta cookie sarlavhalari maydoniga ajratishga ruxsat beradi. Ajratilgan cookielarni siqish orqali, bo‘sh qolgan joyda bir yo‘nalishli yashirin kanal yaratish erishiladi.

SETTINGS freymlaridan foydalangan holda yashirin kanal hosil qilish. SETTINGS freymlari ulanishning so‘zboshi bosqichida har ikki tomondan turli ulanishning o‘ziga xos parametrlarini sozlash uchun ishlatiladi, ammo ular HTTP ulanishi paytida istalgan vaqtda yuborilishi mumkin.

Oqim nazorati yordamida yashirin kanal hosil qilish. HTTP da oqimni boshqarish har bir alohida oqimda yoki butun ulanishda amalga oshiriladi. Ushbu holatda oqimning o‘rtasida qo‘shimcha sifatida kiritishga erishiladi.

HPACK (HTTP sarlavhasini kichiklashtiradi) yordamida yashirin kanallar hosil qilish [4]. Sarlavhani kichiklashtirish *String literal ()* orqali amalga oshirilib, bir bit bayroqda Haffman kodlash usuli qo‘llanilgan yoki qo‘llanilmaganligini hamda kodlash uchun ishlatiladigan baytlar soni va qator ma’lumotlar maydonining kodlangan ma’lumotlarini ko‘rsatadi.

Keltirilgan yashirin kanal hosil qilish usullari va uning algoritmlarini o‘rganish natijasida, mijoz-server ilovalarida qo‘llaniladigan ma’lumotlar tashuvchisi sifatida qo‘llaniladigan HTTP protokoli sarlavhalarida ma’lumot berkitish algoritmi tanlandi. Unda aloqa o‘rnatish uchun ilova ikkita qismiga ega bo‘lishi kerak: ma’lumotlar yuboradigan server va xabarlar qabul qiladigan mijozning qismi [5].

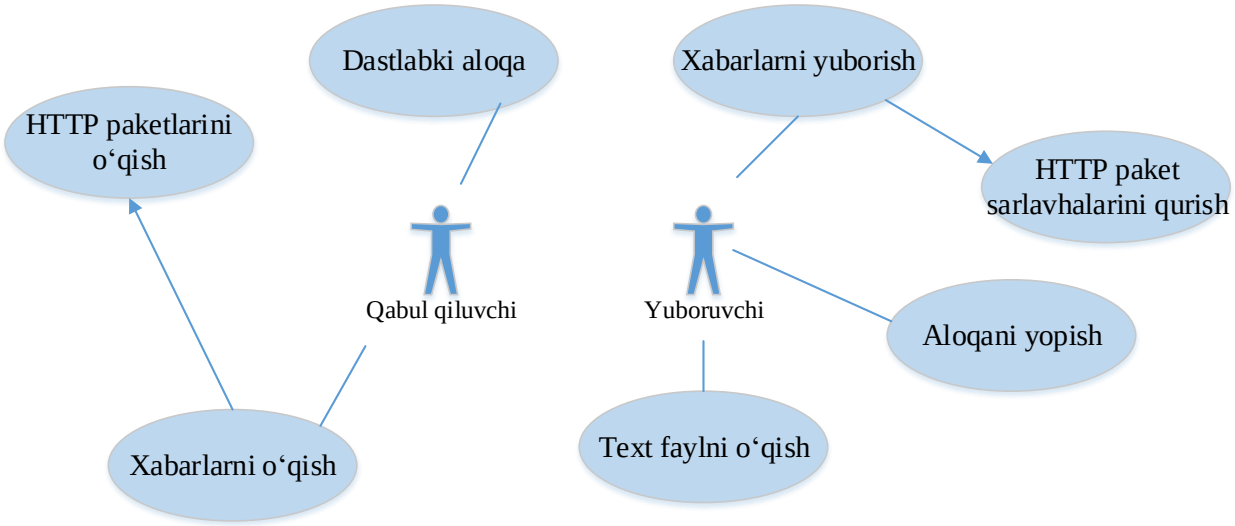


2-rasm. Ilovada qo‘llaniladigan ikki tomonlama aloqa

Protokolda muloqot qilish bir yo‘l bilan amalga oshiriladi: bitta jo‘natuvchi va qabul qiluvchi yoki har bir foydalanuvchiga qaratilgan dasturning ikki qismi. Ikkinchi holda, ma’lumotlarni yuboradigan ikkita turli xil server mavjud bo‘ladi, shuning uchun ikkita turli xil aloqa kanallaridan foydalaniladi. Bunday yechim ilova xavfsizligini yetarlicha oshiradi.

Ikki kanalli aloqa 2-rasmda ko‘rsatilgan. Unga ko‘ra ikkita foydalanuvchi mavjud: qabul qiluvchi va jo‘natuvchi, umuman olganda, ma’lumotni kutayotgan bir nechta qabul qiluvchi bo‘lishi mumkin. Qabul qiluvchi aloqani boshlaydi, agar qabul qiluvchi ishga tushirilmagan bo‘lsa, jo‘natuvchiga qabul qiluvchi aloqada emasligini bildiradi va muloqot boshlanmaydi. Bu HTTP so‘rovi va javob g‘oyasiga asoslangan. Server HTTP so‘rovini yuboradi va qabul qiluvchi tomondan HTTP javobi qaytganida aloqaga kirishadi. Xabarlarni yuborish HTTP sarlavhalarini tahrirlashda boshlanadi. Xabarlarni yaratish va jo‘natish qurilishi HTTP sarlavhalari, xususan, yangi maydonni shakllantirish va uning ma’lumotlarini tahrirlashni o‘z ichiga oladi.

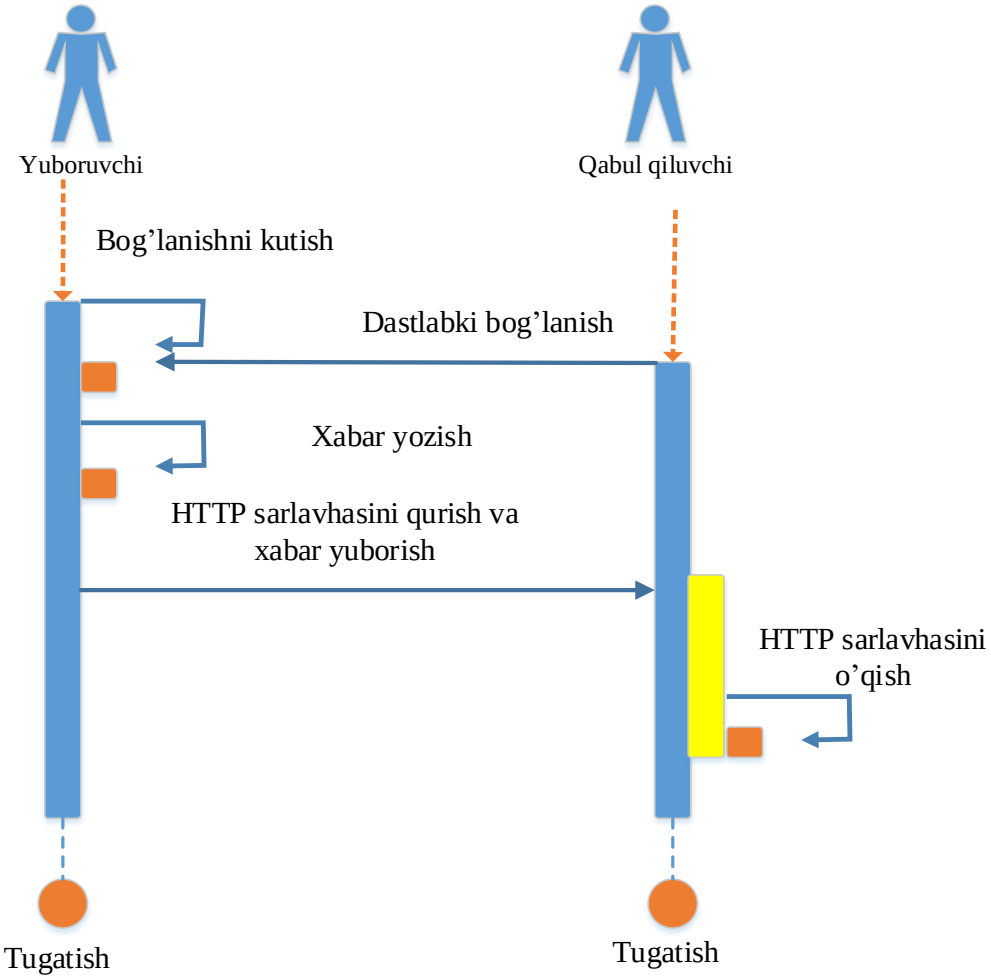
Qabul qiluvchi huddi shu dastur yordamida serverning ikkinchi nusxasi orqali javob beradi va parallel ravishda ochiladi [6]. 3-rasmda aloqa paytidagi ketma-ketliklar tartibi ko‘rsatilgan. Dastlab server ishlay boshlaydi va shu yerda jo‘natuvchi qabul qiluvchiga uzatiladigan xabarlar sonini belgilaydi.



3-rasm. Ilovadan foydalanish diagrammasi

Keyin server tinglashni boshlaydi va u ma’lumotlarni yuborish imkonini beradi. Oxirgi xabarda qabul qiluvchining ilovasini yopadigan maxsus bayroq mavjud. Vaholanki qabul qiluvchi o‘z ilovasini qayta ishga tushirishi mumkin va keyin xabarlarning qolgan qismi yetkaziladi, lekin oldingi ma’lumotlar yo‘qoladi. Muloqot boshlanishidan oldin mijoz va server jo‘natuvchining IP manzili va aloqa o‘tkaziladigan port raqami haqida ma’lumot almashishi kerak[9]. Bunda maxfiy ma’lumotlar va Staganografik kalit ham boshqa himoyalangan kanal orqali yuborilishi lozim.

Mijoz va server tomonidagi dasturda bunday xususiyat o‘rnatilgan, shuning uchun aloqa qilishdan oldin bunday ma’lumotlarni almashishning hojati yo‘q. Yana bir imkoniyat - qabul qilishda protokol snifieridan foydalanish mumkin. Ushbu dastur tahlili tarmoq trafigida Sniferlar har bir paketni ushlaydi va uni o‘qish imkoniyatini beradi. Biror kishi barcha paketlarni qo‘lga kiritishi va HTTP javoblarini qidirishi hamda ular orqali maxfiy xabarni topishi mumkin [7].

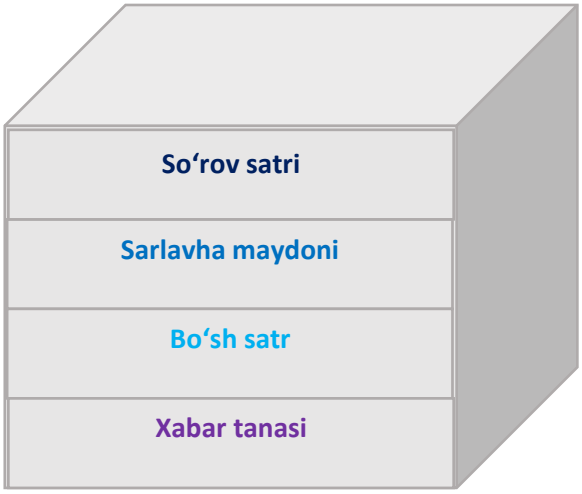


4-rasm. Ilovaning bajarilish ketma-ketlik diagrammasi

HTTP protokoli tuzilishiga asoslanib, TCP/IP paketiga ma'lumotlar massasini joylashtirish muammosini hal qilish uchun real vaqt rejimida ma'lumotlarni yashirish taklif qilinadi. Maxfiy ma'lumotlar HTTP protokoli sarlavhalari maydonlariga kiritiladi. Algoritmning sig'imi tajribalar orqali isbotlangan va 1000 baytdan ortiq ma'lumotlarni HTTP so'rov paketiga (Get-packet) joylashtirish mumkin [10], shaffofligi esa juda yaxshi. Shu bilan birga, yashirin ma'lumotlar paketlarni uzatishga ta'sir qilmaydi.

HTTP protokoli strukturasini yashirin ma'lumotlarning tashuvchisi sifatida qabul qilgan holda, xabar sarlavhalari sifatida yashirilgan ma'lumotlar HTTP so'rov paketi sarlavhalari maydonlariga joylashtiriladi. Algoritmning ko'rinmasligi va yashirish qobiliyati juda yaxshi.

Quyidagi 5-rasmdan ko'rinib turibdiki, HTTP so'rov-paket strukturasi to'rt qismga bo'lingan [8]: Birinchi qism - so'rovlar qatori, bu so'rovlar-paketining birinchi qatori. Ikkinchi qism sarlavha maydonlari va sarlavha maydonlari orqasidagi bo'sh qator.



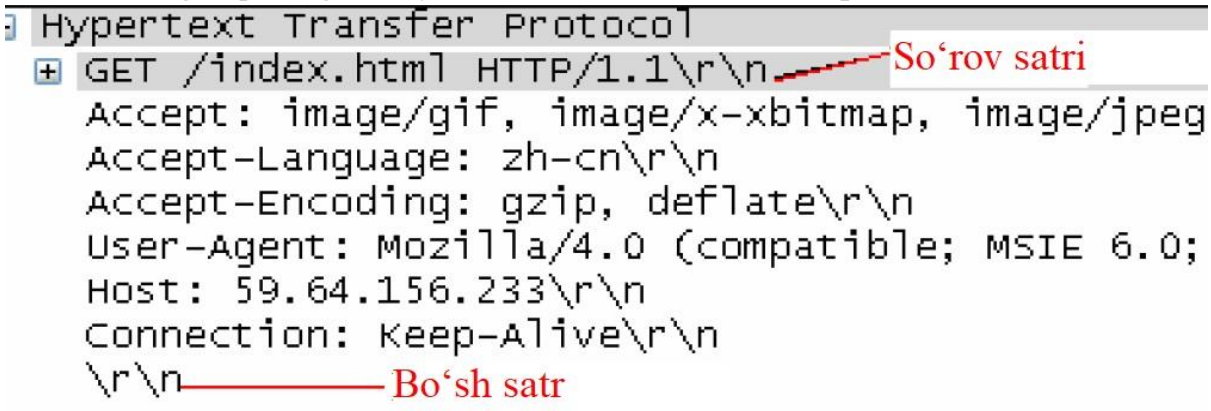
5-rasm. So‘rov xabarining tuzilishi

1. *So‘rov satri*: So‘rov qatori so‘rov usuli bilan boshlanadi, so‘ngra so‘rov-URL va protokol versiyasi bilan davom etadi. Bunda *HTTP/1.1* protokol versiyasidagi *GET / HTTP/1.1/r/n* orqali amalga oshiriladi.

2. *So‘rov sarlavhasi maydonlari*: So‘rov sarlavhasi maydonlari mijozga so‘rov va mijozning o‘zi haqida qo‘shimcha ma’lumotlarni serverga uzatish imkonini beradi. Har bir sarlavha maydoni nomidan keyin ikki nuqta “:” qo‘yilib, maydon qiymati kiritiladi. Maydon nomlari katta-kichik harflarni farqlamaydi.

3. Bo‘sh satr: Sarlavha maydonlarini xabarning asosiy qismidan ajratib turganda amaliy ahamiyatga ega emas[11].

4) Xabar tanasi: GET-paketning xabar tanasi (agar mavjud bo‘lsa) so‘rov yoki javob bilan bog‘liq bo‘lgan obyekt tanasini tashish uchun qo‘llaniladi.



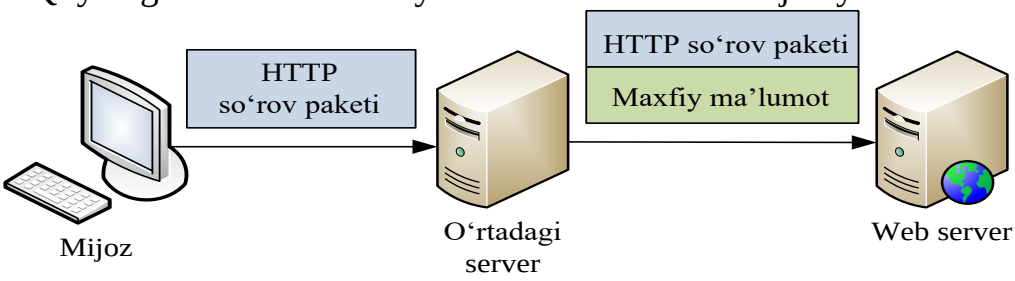
6-rasm. GET-so‘rovi xabari

Yuqoridagi rasmdan ko‘rinib turibdiki, mijoz serverga ulanganida olingan HTTP so‘rov paketi. HTTP so‘rov paketi sakkizta xabar sarlavhasini o‘z ichiga oladi, ammo xabarning asosiy qismi ko‘rinmaydi[12].

Algoritmda maxfiy ma’lumotlarni yashirish uchun HTTP so‘rov paketining tuzilishidan foydalanadi. HTTP/1.1 gipermatnni uzatish protokoliga muvofiq ba’zi xabar sarlavhalarini mustaqil belgilash mumkin. Shuning uchun maxfiy ma’lumotlar

HTTP so‘rov-paketining sarlavha maydonlariga joylashtiriladi va HTTP so‘rov-paketi qabul qiluvchi tomonga yuboradi.

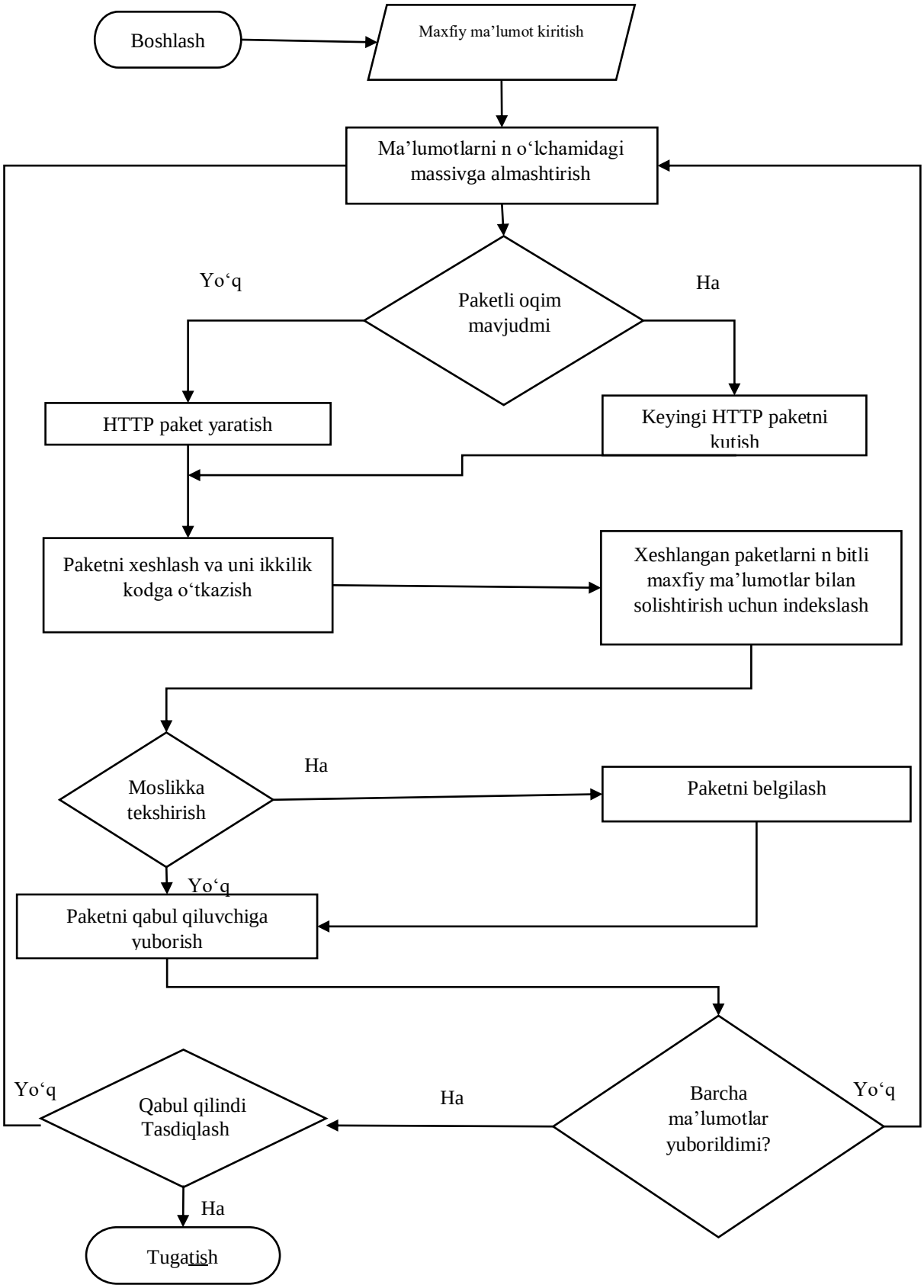
Agar mijoz veb-serverning resursiga kirishni istasa, mijoz va veb-server birinchi navbatda uch marta qo‘l siqish paketlarini uzatadilar, keyin mijoz HTTP so‘rov paketini veb-serverga yuboradi. HTTP protokoli asosida ma’lumotlarni yashirishni amalga oshirish uchun o‘rta server mijoz va veb server o‘rtasida joylashtiriladi. Mijoz va veb-server o‘rtasidagi barcha ma’lumotlar paketlari o‘rta server tomonidan uzatiladi. O‘rta server mijozdan so‘rov paketini qabul qilgandan so‘ng, so‘rov paketining bosh maydonlariga yashirin ma’lumotlarni joylashtiradi va so‘rov paketini taklif qilingan ma’lumotlarni yashirish algoritmini amalga oshiradigan veb-serverga uzatadi. Quyidagi 7-rasmda maxfiy ma’lumotlarni uzatish jarayoni tasvirlangan.



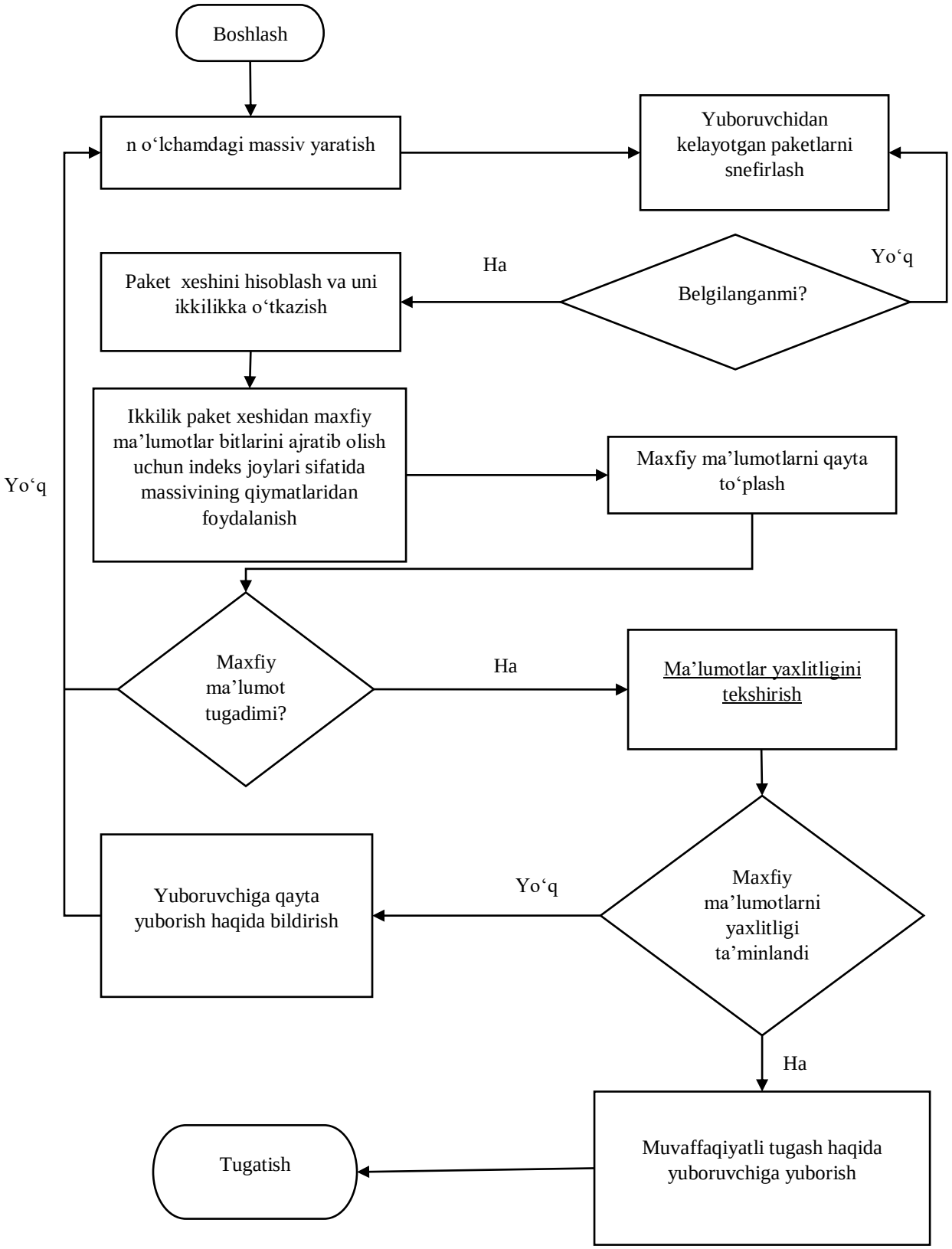
7-rasm. Axborotni yashirish jarayoni

O‘rtadagi server HTTP so‘rov paketiga yashirin ma’lumotni kiritganligi sababli, HTTP so‘rov xabari va unga rioya qilish uchun paket uzunligi o‘sishiga olib keladi. IP nazorat summasini qayta hisoblash va TCP tasdiqlash raqami va TCP tartib raqami o‘rtadagi server tomonidan o‘zgartirilishi kerak[13].

Algoritm jo‘natuvchidan keladigan va qabul qiluvchiga yashirin kanal sifatida o‘tadigan paketlar oqimidan foydalanish uchun mo‘ljallangan. Agar paket oqimi mavjud bo‘lmasa, algoritm yashirin kanal sifatida ishlaydiganlaridan birini yaratadi. Oqimdagi har bir paket, uning xeshining maqsadli joylaridan ma’lum miqdordagi bitlar bir xil miqdordagi maxfiy ma’lumotlar bitlariga mos kelishini tekshirish uchun baholanadi. Agar ular mos keladigan bo‘lsa, paket belgilanadi va yuboriladi. Qabul qiluvchi jo‘natuvchining mashinasidan kelayotgan trafikni ushlaydi va belgilangan paketlarni qidiradi. Belgilangan paketni ko‘rgandan so‘ng, qabul qiluvchi belgilangan paketning bir xil xeshini yaratadi va ushbu paketdan maxfiy ma’lumotlar bitlarini chiqaradi hamda yashirin ma’lumotlarni qayta to‘playdi. Har bir belgilangan paket o‘zgarishi uchun solishtiriladigan paket xesh bitlarida baholanadigan bitlarning joylashuvi[14]. Baholangan bitlarning joylashishini ko‘chirish orqali tajovuzkor to‘g‘ri joylashuvni taxmin qila olmaydi natijda, yashirin ma’lumotlarni chiqara olmaydi. Ular paket xeshidan baholanishi kerak bo‘lgan bitlarning tartibini va qo‘pol kuch hujumini foydasiz qiladigan o‘ziga xos joylashuvini bilishlari kerak. Bundan tashqari, paket yaxlit obyekt sifatida tahlil qilinganligi sababli, statistik tahlil hech qanday natija bermaydi, chunki paket foydasiz va qabul qiluvchidan tashqari hech qanday maxsus ma’noni bildirmaydi.



8 -rasm. Ma’lumot jo‘natuvchi tomoni algoritmi



9-rasm. Ma’lumotni qabul qiluvchi tomoni algoritmi

8-rasmda jo‘natuvchining ma’lumotlarni yashirish va umumiy algoritm yordamida qabul qiluvchiga jo‘natish qadamlari ko‘rsatilgan.

9-rasmda qabul qiluvchi tomonning umumiy algoritmdagi qadamlar ko‘rsatilgan. Dastlab, jo‘natuvchi tomonidan qo‘llaniladigan bir xil almashtirish massivini yaratishdan boshlanib, keyingi bosqichlarda yashirin ma’lumotlarni qidirish bilan davom etadi. Yuboruvchi ham, qabul qiluvchi ham sxemani bilishi va bir xil maxfiy kalitga ega bo‘lganligi sababli, qabul qiluvchi tasodifiy almashtirish generatorini yaratish uchun undan foydalanadi va baholash hajmi bo‘lgan almashtirish massivini yaratadi[15].

Ushbu algoritm HTTP uchun mo‘ljallangan bo‘lib, uni HTTPS orqali amalga oshirilsa, MITM (o‘rtadagi odam) hujumidan himoyalash mumkin bo‘ladi. Ushbu hujum muvafaqqiyatli amalga oshishi uchun esa, SSL sertifikatlari kaliti hujumchida bo‘lishi talab etiladi. Axborot xavfsizligi nuqtai-nazaridan yopiq kalit faqat foydalanuvchining o‘zida saqlanadi.

XULOSA

HTTPS so‘rovlarining mazmunini ko‘rish uchun hujumchi mijoz yoki serverning nomidan ish bajarishi kerak. Agar, mijoz tomonidan qalbaki serifikat orqali so‘rovlarni tinglashga urinish aniqlansa, aloqa tezda to‘xtatiladi.

Dastlabki ulanishda server tomonidan HTTP protokolidan foydalanilsa, MITMni amalga oshiruvchi shaxs SSLSTRIP hujumi orqali oson trafikni tinglay oladi. Chunki, HTTP so‘rovlari shifrlanmagan ochiq matn ko‘rinishida amalga oshiriladi.

Algoritmda ma’lumot yashirin sohalarga biriktirilgan holda jo‘natiladi va asimmetrik shifrlash algoritmlaridan (sertifikat) foydalangan holda amalga oshiriladi. Ushbu holatda qabul qiluvchi kalitlarni hosil qiladi va yopiq kalitni o‘zida saqlaydi. Jo‘natuvchi trafikni ochiq kalit bilan shifrlab jo‘natadi. Uning xavfsizligi shifrlashda qo‘llaniladigan algoritmning xavfsizligiga bog‘liq sanaladi.

FOYDALANILGAN ADABIYOTLAR RO‘YXATI

1. Tran C. M. et al. Cross-protocol unfairness between adaptive streaming clients over http/3 and http/2: A root-cause analysis //Electronics. – 2021. – T. 10. – №. 15. – C. 1755.

2. Brylinski A. S., Bhattacharjya A. Overview of HTTP/2 //Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing. – 2017. – C. 1-6.

3. Ganivev, Abduhalil, Obid Mavlonov, and Baxtiyor Turdibekov. "Improving Data Hiding Methods in Network Steganography Based on Packet Header

Manipulation." 2021 International Conference on Information Science and Communications Technologies (ICISCT). IEEE, 2021.

4. Ghaznavi M. et al. Content Delivery Network Security: A Survey //IEEE Communications Surveys & Tutorials. – 2021. – Т. 23. – №. 4. – С. 2166-2190.

5. Aliwarga H. K., Satriatama A. H., Pratama B. F. A. Performance comparison of fleet management system using IoT node device based on MQTT and HTTP protocol //AIP Conference Proceedings. – AIP Publishing LLC, 2020. – Т. 2217. – №. 1. – С. 020009.

6. Wolsing K. et al. A performance perspective on web optimized protocol stacks: TCP+ TLS+ HTTP/2 vs. QUIC //Proceedings of the Applied Networking Research Workshop. – 2019. – С. 1-7.

7. Akasiadis C., Pitsilis V., Spyropoulos C. D. A multi-protocol IoT platform based on open-source frameworks //Sensors. – 2019. – Т. 19. – №. 19. – С. 4217.

8. Anderson B. et al. Limitless HTTP in an HTTPS World: Inferring the Semantics of the HTTPS Protocol without Decryption //Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy. – 2019. – С. 267-278.

9. Dong, P., Qian, H., Lu, Z., Lan, S., A Network Covert Channel Based on Packet Classification, International Journal of Network Security, 2012, 14(2), 109–116

10. Gianvecchio, S., Wang, H., Wijesekera, D., Jajodia, S., Model-based covert timing channels: Automated modeling and evasion, in: Lippmann, R., Kirda, E., Trachtenberg, A. (eds.) RAID 2008, LNCS, vol. 5230, Springer, 2008, 211–230

11. Gianvecchio, S., Wang, H., An Entropy-Based Approach to Detecting Covert Timing Channels, IEEE Transactions on Dependable and Secure Computing, vol.8, issue 6, 2011, 785–797

12. Jankowski, B., Mazurczyk, W., Szczypiorski, K., PadSteg: Introducing Inter-Protocol Steganography, Telecommunication Systems, 2013, 52(2), 1101–1111

13. Luo, X., Chan, E., Chang, R., TCP Covert Timing Channels: Design and Detection, In proceedings of:

14. IEEE International Conference on Dependable Systems and Networks With FTCS and DCC, Anchorage, Alaska, June 24-27, 2008, 420 – 429

15. Mazurczyk, W., Smolarczyk, M., Szczypiorski, K., On Information Hiding in Retransmissions, Telecommunication Systems: Modelling, Analysis, Design and Management, vol. 52, issue 2, 1113-1121, 2013