**O’ZBEKISTON RESPUBLIKASI OLIY VA O’RTA MAXSUS TA’LIM VAZIRLIGI**

**_____ UNIVERSITETI**

**_____**

**_____fakulteti** **_____ kafedrasi**

**Ro’yxatga olindi №_____** **Ro’yxatga olindi №_____**

**“_____” _____20 y.** **“_____” _____20 y.**

**“_____ “ KAFEDRASI**

**“_____ “ FANIDAN**

# KURS ISHI

**Kriptografik algoritmlar va ularning dasturiy realizatsiyasi**

**Professor o’qituvchi: _____**

**Talabasi: _____**

**Guruh: _____**



**_____ - 20__**

**MUNDARIJA**

# KIRISH

**Kurs ishining dolzarbligi va zarurati.** Axborot texnologiyalari jamiyatning barcha jabhalariga chuqur kirib bormoqda. Bugungi kunda ma'lumotlar nafaqat korporativ yoki davlat tizimlari doirasida, balki har bir oddiy foydalanuvchining faoliyatida muhim resurs sifatida namoyon bo‘lmoqda. Raqamli dunyoda axborotlar global tarmoqlar orqali yuqori tezlikda almashinmoqda, ularning mazmuni esa moliyaviy, tibbiy, harbiy, tijorat va shaxsiy tusga ega bo‘lishi mumkin. Shu sababli, bu kabi axborotlarni ruxsatsiz kirish, o‘g‘irlash, soxtalashtirish yoki buzish holatlaridan ishonchli himoya qilish bugungi zamon axborot-kommunikatsiya infratuzilmasining eng muhim talablari qatoriga kiradi.

Bu boradagi eng samarali yechimlardan biri — kriptografik algoritmlardan foydalanishdir. Kriptografiya — bu ma'lumotlarni shifrlash orqali ularni ruxsatsiz foydalanishdan himoyalashga yo‘naltirilgan ilmiy-amaliy yo‘nalish bo‘lib, u axborot xavfsizligining uch asosiy mezonini — maxfiylik**,** yaxlitlik va autentifikatsiyani ta'minlaydi. Ayniqsa, hozirgi kiberxavfsizlik muhitida raqamli imzo, kalitlarni boshqarish, xashlash algoritmlari kabi kriptografik yondashuvlar internet orqali amalga oshiriladigan har qanday operatsiyaning asosiy xavfsizlik mexanizmlariga aylangan.

Yangi avlod raqamli texnologiyalari, sun’iy intellekt, IoT (Internet of Things), bulutli hisoblash va mobil ilovalar bilan bog‘liq xavflar ortib borayotgan bir sharoitda, ishonchli kriptografik tizimlarni ishlab chiqish va ularni dasturiy mahsulotlar tarkibiga kiritish dolzarb ilmiy-texnik vazifalardan biri hisoblanadi. Shuningdek, ilg‘or algoritmlarni dasturlash tillarida samarali realizatsiya qilish ularning real tizimlardagi qo‘llanilishini ta’minlaydi. Buning uchun esa ushbu algoritmlarning nazariy asoslarini chuqur o‘zlashtirish, ularni amaliy dasturlash ko‘nikmalari bilan uyg‘unlashtirish zarur.

**Kurs ishi mavzusining dolzarbligi.** Ushbu kurs ishi mavzusi — “Kriptografik algoritmlar va ularning dasturiy realizatsiyasi” — nafaqat nazariy jihatdan muhim, balki zamonaviy axborot tizimlari arxitekturasida amaliy qo‘llanilishi bilan ham e’tiborga loyiqdir. Mavzuning dolzarbligi, avvalo, kriptografik yondashuvlarning keng tarqalganligi va doimiy takomillashib borayotganligida namoyon bo‘ladi. Masalan, simmetrik shifrlash algoritmlaridan AES va DES bugungi kunda ko‘plab mobil ilovalar, veb-servislar va moliyaviy tizimlar xavfsizligini ta'minlashda asosiy vosita sifatida foydalanilmoqda. Shu bilan birga, assimetrik algoritmlar (RSA, ElGamal) orqali foydalanuvchilar o‘rtasidagi ishonchli almashinuv imkoniyati yaratilmoqda.

Bundan tashqari, xash-funksiyalar — MD5, SHA-256 — ma’lumotlar yaxlitligini tekshirish, raqamli imzolarni yaratish va autentifikatsiya tizimlarida keng qo‘llaniladi. Shuningdek, ochiq manba kodli kriptografik kutubxonalar — OpenSSL**,** Crypto++**,** PyCrypt**o** va boshqalar — turli dasturiy mahsulotlarda xavfsizlikni ta’minlashga xizmat qilmoqda. Ushbu elementlarning to‘liq o‘rganilishi va dasturiy tajribada qo‘llanilishi zamonaviy IT-mutaxassislar uchun muhim ko‘nikmalardan biridir.

**Kurs ishining asosiy maqsadi va vazifalari.** Kurs ishining asosiy maqsadi — kriptografik algoritmlarning nazariy asoslarini chuqur o‘rganish, ularning amaliy jihatlarini tahlil qilish, shuningdek, Python, C++ va Java kabi dasturlash tillarida AES, RSA va SHA algoritmlarini dasturiy jihatdan amalga oshirish orqali axborotni himoyalash mexanizmlarini amaliyotda qo‘llash bo‘yicha bilim va ko‘nikmalar hosil qilishdan iboratdir.

Ushbu kurs ishi doirasida quyidagi asosiy vazifalar belgilangan:

1. Kriptografiya fanining shakllanishi va rivojlanish bosqichlarini o‘rganish;
2. Kriptografik tizimlarning asosiy komponentlari — kalitlar, shifrlash va deshifrlash mexanizmlarini tahlil qilish;
3. Simmetrik (AES, DES) va assimetrik (RSA) shifrlash algoritmlarining nazariy va amaliy jihatlarini chuqur yoritish;
4. Xash-funksiyalarning ishlash tamoyillari va ularning axborot xavfsizligidagi rolini tushuntirish;

5. Dasturlash tillarida (Python, C++, Java) mashhur kriptografik algoritmlarni dasturiy realizatsiya qilish;

6. Shifrlash algoritmlarini testlash orqali ularning samaradorligi va xavfsizlik ko‘rsatkichlarini baholash;

7. Turli algoritmlarni tezlik, resurs sarfi va xavfsizlik darajasi bo‘yicha solishtirish.

Ushbu kurs ishi orqali nafaqat kriptografik algoritmlar haqidagi nazariy bilimlar mustahkamlanadi, balki ularni real dasturiy muhitda samarali qo‘llash bo‘yicha amaliy ko‘nikmalar ham shakllanadi. Bu esa talabani kelgusida axborot xavfsizligi yo‘nalishidagi murakkab muammolarni mustaqil hal eta oladigan malakali mutaxassis sifatida tayyorlashga xizmat qiladi.

# I BOB. Kriptografik algoritmlar nazariyasi va turlari

## 1.1. Kriptografiyaning nazariy asoslari

Kriptografiya — yunoncha *kryptós* (yashirin) va *gráphein* (yozmoq) so‘zlaridan kelib chiqqan bo‘lib, “yashirin yozuv” ma’nosini bildiradi. Uning asosiy maqsadi — ma’lumotni uchinchi shaxslar tomonidan tushunib bo‘lmaydigan holatda uzatishdir. Kriptografiyaning tarixi insoniyat sivilizatsiyasi rivoji bilan chambarchas bog‘liq bo‘lib, u qadimgi davrlardan to hozirgi zamon axborot xavfsizligining markaziy yo‘nalishlaridan biriga aylangan.

Eng qadimgi kriptografik tizimlar miloddan avvalgi ming yilliklarga borib taqaladi. Masalan, qadimgi Misrda fir’avnlar saroyidagi yozuvlarni maxfiy saqlash maqsadida ayrim belgilarni almashtirish (substitution) asosida shifrlash usullaridan foydalanilgan. Qadimgi Yunonistonda esa mashhur “skitala” usuli mavjud bo‘lgan. Bu usulda yog‘och tayoqqa o‘ralgan pergamentga yozilgan matn faqat tayog‘ning aynan shu diametriga ega bo‘lgan boshqa tayoqqa o‘ralgandagina o‘qilishi mumkin edi.

Klassik kriptografiyada eng mashhur algoritmlardan biri bu — Sezar shifri bo‘lib, u miloddan avvalgi I asrda Rim imperatori Gay Yuliy Sezar tomonidan ishlatilgan. Bu algoritm o‘zak (kalit) asosida alifbo harflarini ma’lum pozitsiyaga siljitish orqali amalga oshirilgan. Masalan, kalit 3 bo‘lsa, A harfi D ga aylantiriladi. Bunday usul oddiy bo‘lsa-da, o‘sha davr uchun samarali hisoblangan.

O‘rta asrlarda kriptografiya asosan diplomatik yozishmalarda va harbiy aloqa vositalarida ishlatilgan. Arab olimi Al-Kindiy (IX asr) birinchi bo‘lib statistik tahlil (frequency analysis) usulini ishlab chiqqan va bu bilan Sezar shifri kabi oddiy shifrlash tizimlarining zaifligini ko‘rsatgan. XV asrga kelib Vigenère shifri kabi ko‘p kalitli (polialfavitik) shifrlash usullari paydo bo‘ldi, bu esa statistik tahlilga qarshi samarali himoya chorasi bo‘ldi.

XX asr boshlarida kriptografiya sanoat va davlatlar miqyosida strategik ahamiyatga ega bo‘ldi. Dunyoga mashhur Enigma mashinasi Ikkinchi Jahon urushi davrida Germaniya tomonidan ishlatilgan bo‘lib, bu mashina murakkab rotorli mexanizmlar yordamida axborotni shifrlagan. Enigma kodining yechilishi esa ittifoqchilarning urushdagi g‘alabasi uchun muhim strategik omil bo‘ldi. Bu yechimning asoschilaridan biri Alan Tyuring bo‘lib, u zamonaviy hisoblash texnikasining asoschisi hisoblanadi.

1970-yillardan boshlab elektron hisoblash mashinalarining rivojlanishi kriptografiyada yangi davrni boshlab berdi. 1976-yilda Whitfield Diffie va Martin Hellman tomonidan taklif qilingan ochiq kalitli kriptografiya (public key cryptography) paradigmasi kriptografik tarmoq xavfsizligida inqilob yasadi. Ularning g‘oyasi asosida RSA algoritmi (Rivest, Shamir, Adleman) ishlab chiqildi. Shuningdek, DES (Data Encryption Standard) algoritmi 1977-yilda AQSh hukumati tomonidan standart sifatida tasdiqlandi. Hozir esa AES (Advanced Encryption Standard) ushbu sohada asosiy standartlardan biri sifatida qo‘llaniladi.

Kriptografiya hozirgi kunda quyidagi sohalarda qo‘llaniladi:

- Internet va elektron pochta xavfsizligi (SSL/TLS, PGP)
- Moliyaviy operatsiyalar (kriptovalyutalar, bank tizimlari)
- Mobil va bulutli texnologiyalar

- Raqamli imzolar va elektron hujjatlar

Kriptografik tizimlar axborotni shifrlash, deshifrlash, kalitlar almashinuvi, autentifikatsiya va integratsiya tamoyillariga asoslanadi. Ular orqali ma’lumotni uzatish yoki saqlash jarayonida uni ruxsatsiz kirishdan himoya qilish mumkin. Quyida kriptografiyaning asosiy tushunchalari yoritiladi:

Kalit — bu shifrlash va deshifrlash jarayonlarida ishlatiladigan maxfiy yoki ochiq ma’lumotdir. Kalit kriptografik tizimning markaziy komponenti bo‘lib, u ma’lumotni kodlash algoritmiga mos tarzda ishlaydi. Kalitlar ikki asosiy turga bo‘linadi:

- Simmetrik kalit: Shifrlash va deshifrlash uchun bitta kalitdan foydalaniladi. Masalan, AES, DES.
- Assimetrik kalit: Shifrlash uchun ochiq (public) kalit, deshifrlash uchun esa yopiq (private) kalit ishlatiladi. Masalan, RSA, ElGamal.

Kalitning uzunligi va murakkabligi tizim xavfsizligining asosiy omillaridan biridir. Masalan, AES algoritmi uchun 128, 192 va 256 bitli kalitlar mavjud. Kalit qanchalik uzun bo‘lsa, tizimni buzish shunchalik qiyinlashadi.

Shifrlash — bu ma’lumotni o‘zgartirish jarayoni bo‘lib, u orqali oddiy matn (plaintext) tushunib bo‘lmaydigan kodlangan matnga (ciphertext) aylantiriladi. Shifrlangan axborot faqatgina maxsus kalit yordamida yana asl holatiga qaytarilishi mumkin. Shifrlash quyidagicha tasniflanadi:

- Blokli shifrlash (Block cipher): Ma’lumotlar belgilangan uzunlikdagi bloklarga bo‘linib, har bir blok alohida shifrlanadi (AES, DES).
- Oqimli shifrlash (Stream cipher): Ma’lumotlar uzluksiz oqim sifatida shifrlanadi (RC4).

Deshifrlash — bu shifrlangan matnni yana asl holatga qaytarish jarayonidir. U foydalanuvchi tomonidan mos kalit orqali amalga oshiriladi. Simmetrik algoritmlarda shifrlash va deshifrlash bir xil kalit orqali amalga oshirilsa, assimetrik algoritmlarda esa turli kalitlar qo‘llaniladi.

Maxfiylik — kriptografiyaning asosiy tamoyillaridan biri bo‘lib, ma’lumot faqatgina ruxsat etilgan subyektlar tomonidan o‘qilishi mumkinligini ta’minlaydi.

Maxfiylik kriptografik algoritmlar yordamida shifrlash orqali amalga oshiriladi. Bu prinsip ma’lumotning ruxsatsiz kirish yoki o‘g‘irlik holatlaridan himoyasini kafolatlaydi.

Bundan tashqari, kriptografiyada yana quyidagi tamoyillar mavjud:

- Yaxlitlik (Integrity): Ma’lumot uzatish yoki saqlash davomida o‘zgarmaganligini ta’minlaydi.
- Autentifikatsiya (Authentication): Foydalanuvchining shaxsini yoki manbaning haqiqiyligini tasdiqlaydi.
- Inkordan himoya (Non-repudiation): Foydalanuvchi ma’lumot yuborganligini rad eta olmasligiga erishiladi.

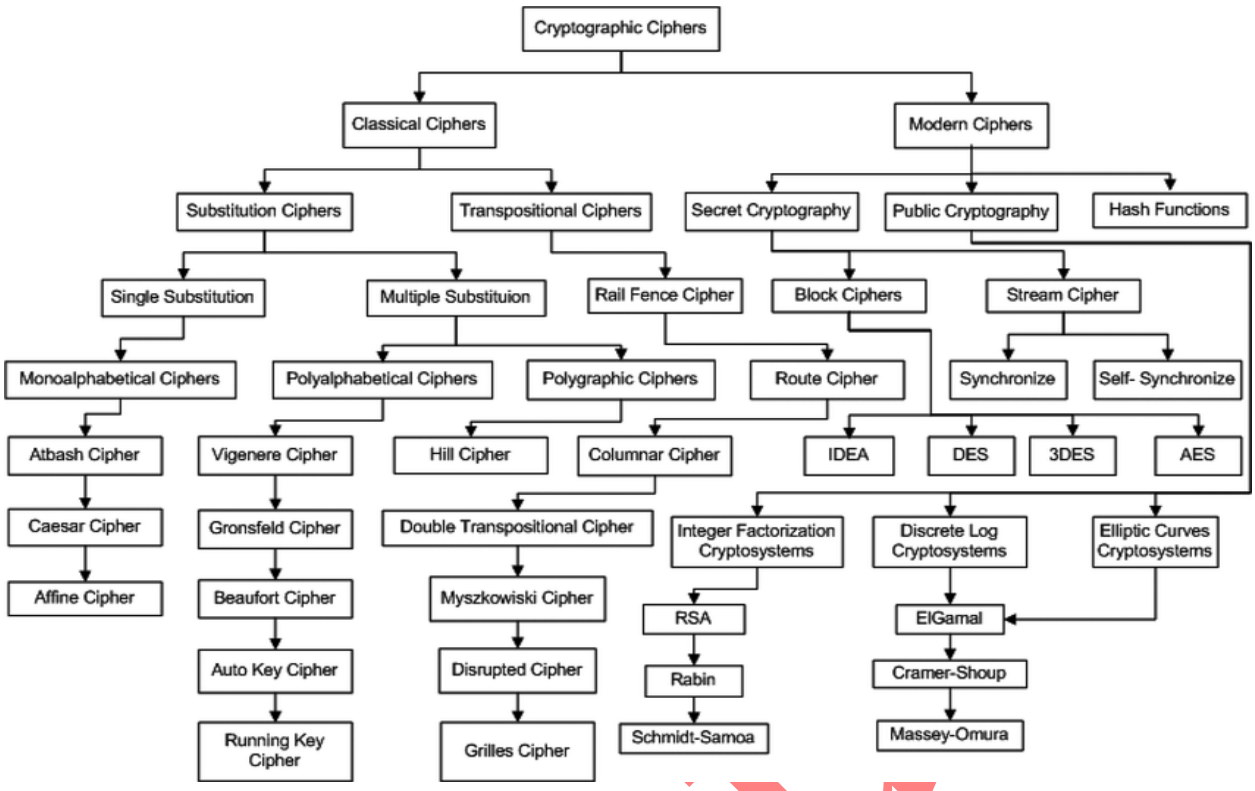
Kriptografiya bugungi kunda axborotni himoya qilish, tizimlararo ishonchli aloqa o‘rnatish va raqamli xavfsizlikni ta’minlashda eng muhim ilmiy va amaliy sohalardan biriga aylangan. Ularning nazariy asoslarini chuqur o‘rganish orqali amaliy tizimlarda qo‘llanishi mumkin bo‘lgan, samarali va xavfsiz algoritmlarni ishlab chiqish mumkin bo‘ladi. Ayniqsa, zamonaviy dasturlash tillari yordamida AES, RSA kabi algoritmlarni dasturiy realizatsiya qilish orqali kriptografik tizimlarning amaliy imkoniyatlarini kengaytirish mumkin. Bu esa nafaqat texnik mutaxassislar, balki axborot xavfsizligi siyosatini ishlab chiqayotgan muassasalar uchun ham muhim nazariy asosni taqdim etadi.

## 1.2. Kriptografik algoritmlarning turlari va qo‘llanilish sohasi

Kriptografik metodologiyalar insoniyat tarixida axborotni sir saqlash va yetkazishda muhim rol o‘ynab keladi. Mazkur soha dastlabki bosqichlarida rahbarlar o‘rtasidagi maxfiy yozishmalarni himoya qilish, urush strategiyalarini yashirish va davlat siyosatini muhofaza qilish kabi amaliy maqsadlarda qo‘llanilgan bo‘lsa, keyinchalik savdo, ta’lim, bank-moliya, sog‘liqni saqlash, ma’lumotlar tahlili, hamda hozirgi kunda global tarmoq infratuzilmasida keng qamrovli ilmiy-texnik tadqiqotlar obyekti sifatida o‘z o‘rnini topdi. Kriptografiyaning bu bosqichi simmetrik va assimetrik shifrlash algorizmlari, shuningdek, xash-funksiyalarining rivojlanishi bilan ajralib turadi.

Simmetrik shifrlash, ya’ni bitta kalit asosida shifrlash va deshifrlash mexanizmini qo‘llagan tizimlar tarixan eng qadimgi usullar hisoblanadi. Kriptografiyaning dastlabki elementlari sifatida harflarni almashtirish (substitutsiya) va belgilash (transpozitsiya) metodlari ishlatilgan. Bu metodologiyalar arxeologik topilmalarda ham uchraydi — masalan, miloddan avvalgi VII-VI asrlarga oid Krishtian Monasteri papiruslarida oddiy shifrlash namunalariga duch kelinadi.

Rim imperatori Julius Sezarning harflarni uch birlikka surish qoidasiga asoslangan Sezar shifri (Caesar cipher) butun G’arb sivilizatsiyasiga tarqalib, soddaligi va samaradorligi bilan ajralib turgan. Garchi bu algoritmni zamonaviy kompyuter kuchi va statistik tahlil usullari yordamida bir necha daqiqada buzish mumkin bo‘lsa-da, u kriptografik metodlarning nazariy asoslarini o‘rganishda birinchi bosqich bo‘lib xizmat qilgan.




Arab olimi Al-Kindi (IX asr) esa “kriptanaliz” tushunchasini ilk bor bayon etgan. Uning “Kitab fi Istikhraj al-Mu‘ammâ” asarida harflar chastotalarini tahlil qilish orqali shifrlangan xabarni tiklash usuli keltirilgan. Bu yondashuv statistik tahlil tamoyillarini amaliyotga tatbiq etish, ya’ni matematik metodlarni kriptografik zaifliklarni aniqlashda qo‘llash imkoniyatini ochib bergan.

O‘rta asrga kelib, Vigenère shifri (1553) polialfavitik yondashuvni joriy qilib, bir nechta alifbolar jadvalini ishlatish evaziga deyarli buzib bo‘lmas tizimni yaratdi. Vigenère mexanizmi «autentik shoh» usuli deb ham atalib, bu algoritmni 19-asrgacha “imba” (inglizcha: unbreakable cipher) deb atashgan.

20-asrda simmetrik shifrlash mexanik qurilmalar yordamida murakkablashdi. Birinchi jahon urushi davrida ishlab chiqilgan Amsler-Charlier mashinasi, so‘ng Ikkinchi Jahon urushi davridagi Enigma, Hagelin C-36 kabi rotorli qurilmalar harflar almashinishining cheksiz kombinatsiyalarini taqdim etdi. Enigma algoritmi 26 harfli alifboni uchta rotor yordamida ko‘plab alt-kalitlar asosida shifrlardi. Biroq, Alan Tyuring va Polikarp savdo ittifoqchilari Enigma kodini sindirish orqali nafaqat

kriptografiyaning amaliy salohiyatini ko‘rsatdilar, balki zamonaviy hisoblash matematikasini ham shakllantirdilar.

Elektron davrning axborot texnologiyalari uyg‘inganiga qadar simmetrik algoritmlar kompyuter yordamida ishlovchi dasturiy va apparat modullar sifatida rivojlandi. 1977-yilda NIST DES (Data Encryption Standard) ni FIPS 46-3 orqali standartlashtirdi. DES 64 bitli blok va 56 bitli kalit uzunligi bilan 16 bosqichli Feistel to‘qima (Feistel network) asosida ishlagan. DES muhitida murakkab substiutsiya-permutatsiya operatsiyasi va P-permutatsiyalari kiruvchi blokni qator transformatsiyalardan o‘tkazib, 16 alt-kalit yordamida qayta ishlagan. DESning matematik himoyasi juda yuqori bo‘lib, uzoq vaqt davlat sirlarini saqlashda davom etgan. Biroq, differensial va lineer kriptanaliz metodlarining rivojlanishi, shuningdek, EFF tashkilotining DES kalitini 56 soatda sinovdan o‘tkazishi DESni zaiflashtirdi.

DESning cheklangan kalit uzunligini bartaraf etish uchun Triple DES (3DES) va Avanac boshqaruvchilari yordamida 112 yoki 168 bitli kalitlardan foydalangan. Shu bilan birga, DESning amaliy zaifliklari matematik tahlilchilarga blok shifrlashning yangi paradigmalarini izlashga turtki berdi.

2001-yilda DESga muqobil sifatida NIST AES (Advanced Encryption Standard) ni FIPS 197 orqali ma’qulladi. Rijndael algoritmiga asoslangan AES 128 bitli blok va 128, 192, 256 bitli kalit uzunliklariga ega bo‘lib, S-karlam (subBytes) orqali no-chiziqli substiutsiya, ShiftRows satr siljishi, MixColumns ustun aralashuvi va AddRoundKey kalit birlashtirish bosqichlaridan tashkil topgan. AES 10, 12 yoki 14 tur iteratsiya bilan ishlagan, bu yordamida kuchli xavfsizlik va yuqori bajarilish tezligini ta’minlagan. Bugungi kunda AES TLS/SSL, VPN, disk shifrlash, IoT qurilmalar, mobil operatsion tizimlar va harbiy muhim ma’lumotlarni saqlashda asosiy standarti bo‘lib xizmat qiladi.

Simmetrik algoritmlar kalit almashish va kalitni ishonchli tarqatish kabi masalalarni hal qilishda qiyinchiliklarga duch kelgan. 1976-yilda Whitfield Diffie va Martin Hellman “New Directions in Cryptography” asarida public-key kriptografiya konseptini ilgari surdi. Bu yondashuvda ochiq kalit (public key) orqali shifrlash, yopiq kalit (private key) orqali deshifrlash amalga oshiriladi. Asimetrik kalitlar matematik

jihatdan bog‘langan, biroq ochiq kalitdan shaxsiy kalitni aniqlash murakkabligi zaiflikni oldini oladi.

RSA (Rivest–Shamir–Adleman) algoritmi 1978-yilda e’lon qilindi va u ikkita katta tub son p va q hosil qilib, n = p imes q, φ(n) = (p−1)(q−1) formulalari orqali ochiq (e) va yopiq (d) kalit juftligini yaratadi. Shifrlash va deshifrlash matematik funksiya C = M^{e} mod n va M = C^{d} mod n bo‘lib, faktorizatsiya masalasining murakkabligi asosida himoyalangan. RSA 2048 yoki 4096 bitli kalitlar yordamida hozirgi kunda elektron imzo, TLS/SSL, PGP, S/MIME, va boshqa xavfsizlik protokollarida keng qo‘llanadi.

ElGamal shifri va Elliptik egri chiziqlar asosidagi (ECC) algoritmlar asosan raqamli imzo (DSA, ECDSA) va kalit almashuv (ECDH) protokollarida tasdiqlangan. ECC, RSA bilan solishtirganda, qisqa kalit uzunligi va kam resurs sarfi hisobiga IoT hamda mobil tizimlarda afzal qabul qilindi. Masalan, 256 bitli ECC kaliti RSA 3072 bitli kalitga teng xavfsizlik darajasini ta’minlashi mumkin.

Xash-funksiyalar kriptografiyaning ajralmas tarkibiy qismi bo‘lib, ular kiruvchi ma’lumotdan istalgan uzunlikdagi digest (hash) ni chiqaradi. Digestdan asl ma’lumotni tiklash deyarli mumkin emas (bir tomonlama funksiya) va ikki xil kirish uchun bir xil digest hosil bo‘lishi (kolliziya) ehtimoli juda past bo‘lishi kerak.

1992-yilda Ronald Rivest tomonidan ishlab chiqilgan MD5 (Message-Digest Algorithm 5) 128 bitli digest ishlab chiqadi. Kirish ma’lumotlari 512 bitli bloklarga bo‘linib, har blok 4 turdagi norasmiy funksiyalar yordamida 4 bosqichli aylanishlardan o‘tkaziladi. MD5 2004-yilda Hans Dobbertin tomonidan kolliziya hujumi aniqlangani sababli kriptografik maqsadlarga yaroqsiz deb topildi. Biroq, altere ayollar md5 checksum, ma’lumot yaxlitligini tekshirish, ma’lumotlarni indekslash va ba’zi eski tizimlarda orqaga moslik uchun MD5 saqlanib qolmoqda.

2002-yilda NIST tomonidan tasdiqlangan SHA-256 (Secure Hash Algorithm 256-bit) algoritmi 256 bitli digest hosil qiladi. Kirish ma’lumotlari 512 bitli bloklarga bo‘linadi, Message Schedule va compression funksiyalaridan o‘tkazilib, 64 bosqichli iteratsiya orqali 8 registr yangilanishi amalga oshiriladi. SHA-256 kriptografik

xavfsizlik talablariga javob beradi — hozirgacha hech qanday samarali kolliziya va preimage hujumi aniqlanmagan. SHA-256 blockchain va cryptocurrency (Bitcoin, Ethereum) tarmoqlarida proof-of-work, hamda TLS/SSL, IPsec, SSH kabi protokollarda ma’lumot yaxlitligi va autentifikatsiya mexanizmlarida keng qo‘llanmoqda.

Kriptografik algoritmlarning amaliy qo‘llanilishi quyidagi sohalarni qamrab oladi:

- Internet va kommunikatsiya protokollari: TLS/SSL, HTTPS, SSH orqali ma’lumot uzatish xavfsizligi;
- Mobil ilovalar va IoT: ECC, AES bilan kam resursli qurilmalarda ma’lumotni himoyalash;
- Ma’lumotlar bazasi va disk shifrlash: Transparent Data Encryption (TDE), BitLocker, FileVaultning asosiy komponentlari sifatida AES;
- Moliyaviy tizimlar va elektron to‘lovlar: EMV chip kartalari, online banking, kriptovalyuta tranzaksiyalari;
- Blockchain va distributed ledger: SHA-256 asosida konsensus mexanizmi, hash pointer marosimlari;
- Raqamli imzolar va sertifikatlar: X.509, PKCS#7, digital signature algorithmlar;
- Sog‘liqni saqlash va hukumat tizimlari: HIPAA, GDPR, e-government loyihalarida ma’lumotlar maxfiyligi.

Kriptografik algoritmlarning uzluksiz takomillashuvi, kvant kompyuterlarining rivojlanishi fonida kvantga qarshi kriptografiya (post-quantum cryptography) kabi yangi ilmiy yo‘nalishlarni yuzaga keltirmoqda. Shuningdek, homomorfik shifrlash, nol bilimli isbotlar (zero-knowledge proofs) va blokcheyn ekotizimida xavfsizlikni ta’minlovchi smart contract standartlari kriptografiyaning yangi avlodini belgilamoqda.

## Birinchi bob bo’yicha xulosa

1-bob doirasida avvalo kriptografiyaning tarixiy ildizlari va nazariy poydevorlari tahlil qilindi. Qadimgi davr yozuv sirlarini himoya qilish usullaridan boshlab, o‘rta asr Arab kriptanalitiklarining statistik tahlilga asoslangan yondashuvlarigacha bo‘lgan g‘oyalar muhokama qilindi. Sezar shifri va Vigenère boshchiligidagi klassik polialfavitik algoritmlar sinovardan o‘tganzaifliklari va murakkablik darajalari, shuningdek, Enigma kabi mexanik rotorli qurilmalar tarixdagi burilish nuqtalarini tashkil etgani qayd etildi. Bu rivojlanish bosqichlari kriptografiyaning matematik asoslarini poydevor qilib, 1970–80-yillarda simmetrik (DES, AES) hamda assimetrik (RSA, ECC) paradigmalarning yuzaga kelishiga zamin yaratdi.

Nazariy asoslar doirasida kalit tushunchasi — simmetrik hamda assimetrik kalit juftliklarining roli, shifrlash-deshifrlash jarayonining mexanizmi va ma’lumotning maxfiyligini ta’minlash tamoyillari yoritildi. Blok va oqimli shifrlash konstruktsiyalari, Feistel tarmoqlarining xususiyatlari, shuningdek, ochiq-yopiq kalit yondashuvidagi matematik murakkablik manbalari batafsil tahlil qilindi.

Keyingi bosqichda zamonaviy kriptografik algoritmlarning turlari va ularning amaliy sohalari yoritildi. DES algoritmining 56 bitli kaliti va blok uzunligi, uning permutatsiya-substiutsiya asosidagi Feistel tuzilishi hamda differensial kriptanaliz hujumlariga qarshi zaifliklari ko‘rib chiqildi. AES algoritmi esa 128 bitli blok va 128/192/256 bitli kalit variantlari, S-karлам, ShiftRows, MixColumns va AddRoundKey bosqichlari orqali yuqori xavfsizlik va samaradorlikni ta’minlaydi. Assimetrik sohada RSA algoritmi tub sonlarni faktorizatsiya qilish murakkabligiga, ECC esa kamroq resurs va qisqa kalit uzunligiga tayanib, raqamli imzo va kalit almashuvi protokollarida keng qo‘llanishi misol qilindi.

MD5 va SHA-256 kabi mashhur xash-funksiyalarning konstruktiv tamoyillari chuqur tahlil qilinib, ularning kriptografik amaliyotdagi o‘rni alohida yoritildi. Bu funksiyalarni shakllantiruvchi asosiy prinsiplardan biri — bir tomonlama ishlash xususiyati bo‘lib, unga ko‘ra, istalgan ma’lumotdan xash-kod olish oson, biroq aksincha — xash-koddan asl ma’lumotni aniqlash amalda mumkin emas. Bu xususiyat, ayniqsa,

parolni ochiq holda emas, balki uning xash-kodi shaklida saqlash imkonini berib, parol xavfsizligini ta’minlashda muhim rol o‘ynaydi.

Shuningdek, xash-funksiyalarning ikkinchi muhim xususiyati — avlanch effekti (avalanche effect) bo‘lib, bu ozgina o‘zgarish kiritilgan kirish ma’lumotining xash natijasida butkul boshqa qiymatga aylanishini anglatadi. Bu tamoyil, ayniqsa, shifrlash algoritmlarining chidamliligini oshirish, raqamli imzo va autentifikatsiya jarayonlarida ma’lumotni o‘zgartirib bo‘lmas holatda saqlash uchun zarurdir.

Uchinchi muhim jihat — bu kolliziya qarshiligi (collision resistance) bo‘lib, unga ko‘ra, ikkita turli ma’lumot uchun aynan bir xil xash qiymatni topish juda qiyin bo‘lishi kerak. Biroq amaliy tadqiqotlar va kriptoanalizlar shuni ko‘rsatdiki, MD5 algoritmi ushbu tamoyilga to‘liq javob bera olmaydi. 2005-yildan boshlab MD5’ga nisbatan samarali kolliziya hujumlari aniqlangan bo‘lib, bu uni xavfsiz tizimlarda qo‘llashni tavsiya etilmasligiga olib kelgan. Jumladan, MD5 orqali imzolangan sertifikatlar, elektron hujjatlar va autentifikatsiya xabarlari hujumchiga noto‘g‘ri ishonch hosil qilish imkonini beradi.

## II BOB. Kriptografik algoritmlarning dasturiy realizatsiyasi

### 2.1. Kriptografik algoritmlarni dasturlash muhitida amalga oshirish

Julia Borghoff va boshqalar tomonidan ishlab chiqilgan PRINCE shifri 64 bitlik blok o‘lchami va 128 bitlik kalitdan foydalanadi. Ushbu shifr kalitni matn bo‘ylab tarqatadi va kriptologik hujumlarning oldini olish uchun maxsus mexanizmlarga ega. Deukjo Hong va boshqalar esa Feistel tarmog‘iga asoslangan shifrni ishlab chiqishgan. U ham 64 bitlik blok o‘lchami va 128 bitlik kalitdan foydalanadi. Algoritm oddiy XOR va siljitish (shift) operatsiyalarini

$F_0$ va $F_{31}$ funksiyalarida qo‘llaydi.

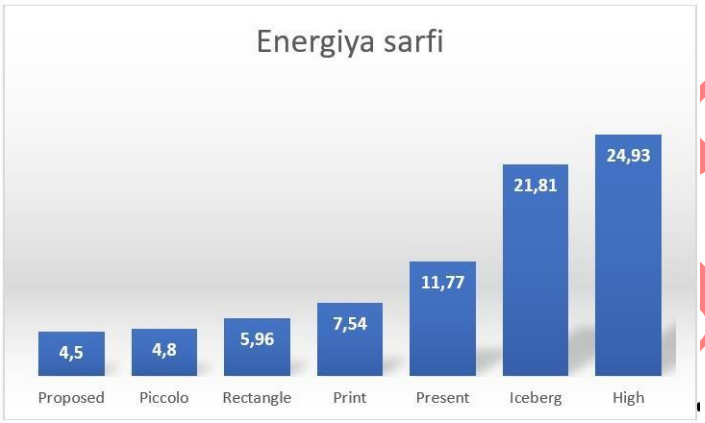
Ray Beaulieu va boshqalar tomonidan har xil blok va kalit o‘lchamlariga ega bo‘lgan shifrlar ishlab chiqilgan. Ushbu shifrlar apparat va dasturiy tizimlarning protsessorlardagi tuzilishini takomillashtirishga mo‘ljallangan. Shifr modulyar qo‘shish, XOR, chapga va o‘ngga aylanish (circular shift) operatsiyalaridan foydalanadi. Yengil vaznli kriptografiya algoritmlari bo‘yicha sharh Kong Jia Hao va boshqalar tomonidan muhokama qilingan. Ular energiya va resurslarni samarali ishlatish uchun optimallashtirilgan algoritmlar haqida ma’lumot beradi.

Gauravm Bansod tomonidan PRESENT- GRP gibrid usuli tushuntirilgan. Bu usulda kirish ma’lumotlari bloklari PRESENT S-box orqali tarqatilib, chiqish ma’lumotlari PRESENT-GRP algoritmi yordamida akslantirilib, keyin o‘rin almashtirish qatlamiga o‘tkaziladi. PRESENT-GRP algoritmi S-box 4x4 tuzilishda yaratilgan bo‘lib, murakkablik va energiya iste’molini kamaytirishga qaratilgan.

64-bitli qiymat ustida operatsiya bajarish uchun loyihalash shunday amalga oshirilganki, u faqat 16 ta 4-bitli PRESENT S-boxdan foydalanadi va PRESENT GRPga permutatsiya uchun uzatiladi. PRESENT-GRP gibrid strukturasining xotira talabi mavjud algoritmlarga nisbatan ancha kam. GRP P-box gatelar ekvivalentini kamaytirish uchun yetti bosqichdan foydalanadi. Bitlar quyidagicha guruhlanadi: birinchi guruh 0-bit va 64-bitni, ikkinchi guruh 1-bit va 65-bitni o‘z ichiga oladi va hokazo. Shifrlash algoritmlarining yengil vaznli apparat va dasturiy ta’minotda amalga oshirilishi haqida

George Hatzivasilis va boshqalar tomonidan ham muhokama qilingan.

Quvvat sarfi IoT qurilmalarida hozirda eng muhim omillardan biri. Ushbu talab doirasida Muhammad Rana o‘z maqolasida quyidagicha tadqiqot natijalarini e’lon qildi. 1-rasmda keltirilgan Proposed, Piccolo, Rectangle, Print, Present, Iceberg, High algoritmlarini quvvat sarfi ustunlar shaklida keltirilgan [2].

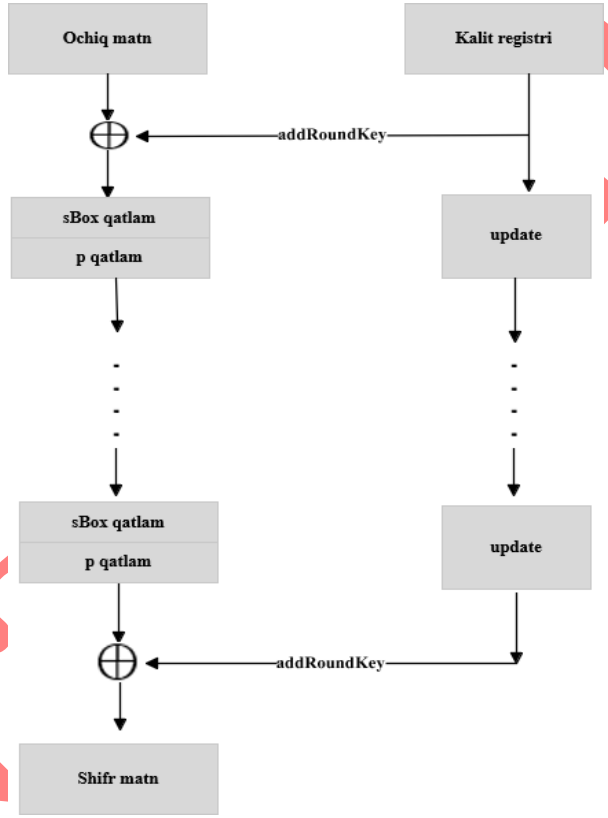



1- rasm. Turli yengil vaznli algoritmlarning energiya sarfi

Yengil vaznli algoritm tavsifi. PRESENT- bu yengil vaznli blokli shifrlash algoritmi bo‘lib, 2007- yilda Andrey Bogdanov va boshqalar tomonidan ishlab chiqilgan. Ushbu algoritm asosan apparat va dasturiy ta’minot resurslari cheklangan tizimlar, masalan, IoT qurilmalari, RFID texnologiyalari va sensor tarmoqlari uchun mo‘ljallangan. 2012-yilda Xalqaro Global Elektron Spetsializatsiya Komissiyasi (ISO/IEC) tomonidan nazorat qilingan va asosan yengil vaznli kriptografiya uchun moslashtirilgan.

*Algoritmni afzalliklari.* Kichik tranzistor soni, past energiya sarfi va xavfsizlikni ta’minlash uchun optimallashtirilgan. PRESENT algoritmi o‘zining ko‘plab afzalliklariga qaramay, bir nechta kamchiliklari mavjud. Dasturiy ta’minotda algoritmning shifrlash tezligi past. PRESENT asosan apparat muhitida samarali ishlash uchun mo‘ljallangan. Dasturiy ta’minotda ishlash tezligi boshqa yengil vaznli algoritmlarga nisbatan pastroq. Algoritmning keyingi kamchiligi bu uning kalit uzunligi. 80 bitlik kalit uzunligi uzoq muddatli xavfsizlik uchun yetarli emas, chunki hisoblash quvvatining oshishi bilan bruteforce texnologiyasi yordamida kalitni oshkor bo‘ladi.

128 bitlik PRESENT algoritmi mavjud lekin 80 bitli kalit uzunligi mavjud varianti ko‘proq ishlatiladi. Algoritm differensial va chiziqli kriptotahlillarga bardoshsiz. Algoritm 31 rounddan iborat bo‘lib, bu boshqa yengil vaznli shifrlash algoritmlariga nisbatan ko‘proq. Bu ko‘proq energiya sarfiga olib kelishi mumkin. Algoritm autentifikatsiya yoki ma’lumotlarning yaxlitligini ta’minlash uchun qo‘shimcha mexanizmlarni o‘z ichiga olmaydi.




2- rasm. Present shifrlash algoritmini tavsifi

PRESENT algoritmi 80 yoki 128 bitli kalitlar mavjud, bu yerda 80-bitli kalit uchun ishlash ketma- ketligi keltirilgan. Foydalanuvchi tomonidan taqdim etilgan kalit K registrida saqlanadi va quyidagicha ifodalanadi:

$$K = k_{79}k_{78}\ldots\ldots k_0$$

Bu yerda, $k_{79}$ - eng boshida (chapdagi) bit va

$k_0$- eng oxirgi (o‘ngdagi) bitni bildiradi.

Raund kalitini olish (Round Key Extraction) Har bir raundda 64-bitli raund kaliti $K_i$ olinadi, bu esa K registrining hozirgi holatining chapdagi 64 bitidan iborat.

$$K_i = k_{79}k_{78} \dots \dots k_{16} = \kappa_{63} \dots \dots \kappa_0$$

Bu yerda, $K_i$ 80 bitli kalitni generatsiya qiladi va uning 64 ta bitlari K registrining chap tomonidan olinadi.

Kalitni yangilash (Key Update):

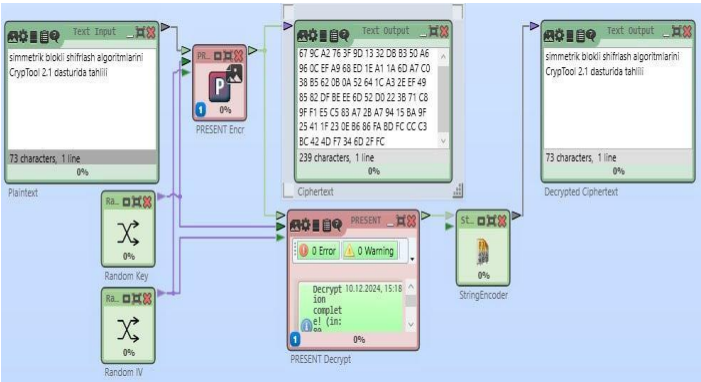
Raund kaliti $K_i$ generatsiya qilingach K registri yangilanadi. Yangilash jarayoni quyidagicha amalga oshiriladi.

K registrining bitlari bir pozitsiyaga chapga aylantiriladi. Bu holatda chapdagi 61 bit o‘ng tomonga ko‘chiriladi (left shift operation).

- S-box yordamida kalitning yuqori 4 bitiga chiziqsiz almashtirish (substitution) qo‘llaniladi.
- Round raqamiga asoslanib, kalitning oxirgi bitlari yangilanadi.

Har bir raundda yangi kalit yuqorida keltirilgan tartibda hosil qilinadi va K registri yangilanadi. Bu jarayon key schedule deb ataladi va algoritm bardoshliligini oshirishda muhim ro‘l o‘ynaydi. Har bir raundda yangi kalitning hosil bo‘lishi, algoritmini va uning tahlil qilish jarayonini murakkablashtiradi.

PRESENT shifrlash algoritmini CrypTool 2.1 dasturi ta’minot muhitida ishlatib ko‘rildi va algoritmni solishtirish uchun ikkinchi algoritm sifatida Speck blokli shifrlash algoritmi tanlab olindi va quyidagi natijalar olindi. PRESENT shifrlash algoritmi uchun quyidagi ochiq matn kiritildi. Algoritmni ishlash sxemasi 3-rasmda keltirildi.




3- rasm. PRESENT shifrlash algoritmini dasturiy ta’minotda ishlash jarayoni

3-rasmda PRESENT shifrlash algoritmida ochiq matn sifatida “simmetrik blokli shifrlash algoritmlarini CrypTool 2.1 dasturida tahlili” olindi va Text Output da shifrmatn chiqdi.

Algortimni imkoniyatlarini solishtirish maqsadida SPECK yengil vaznli shifrlash algoritmi tanlab olindi. SPECK - bu 2013-yilda AQSh Milliy Xavfsizlik Agentligi (NSA) tomonidan taklif qilingan mashhur yengil shifrlash algoritmidir. Uning maqsadi cheklangan qurilmalarda xavfsizlikni ta’minlashdir.

4-rasm. SPECK shifrlash algoritmini dasturiy ta’minotda ishlash jarayoni

4-rasmda SPECK shifrlash algoritmida ochiq matn sifatida “simmetrik blokli shifrlash algoritmlarini CrypTool 2.1 dasturida tahlili” olindi va Text Outputda shifrmatn chiqdi.

Olingan natijalar asosida tahliliy jadval hosil bo‘ldi. 3-jadvalda PRESENT va

SPECK shifrlash algoritmlarining natijalari qayd qilindi. Jadvalda algoritmlarning kiruvchi ma’lumotlari (plaintext, key, va initialization vector) va ularning chiqish natijalari (ciphertext va qayta ochilgan plaintext) ko‘rsatiladi.

1-jadval. Algoritmlarning o‘zaro xususiyatlari

| **Xususiyat** | **PRESENT** | **SPECK** |
|---|---|---|
| Ochiq matn | Simmetrik blokli shifrlash algoritmlarini CrypTool 2.1 dasturida tahlili | Simmetrik blokli shifrlash algoritmlarini CrypTool 2.1 dasturida tahlili |
| Kalit | Tasodifiy (Random Key) | 00 01 08 09 10 11 18 19 |
| IV (initialization Vector) | Tasodifiy (Random IV) | 80 AF 12 7A |
| Shifr matn | 67 9C A2 76 3F 9D 13 32 DB B3 50 A6 | A7 32 B7 A7 33 6C 72 28 |

IoT (Internet of Things) qurilmalari uchun shifrlash algoritmlarini tanlashda ularning energiya samaradorligi muhim ahamiyatga ega. 2-jadvalda PRESENT va SPECK shifrlash algoritmlarining IoT qurilmalari uchun energiya samaradorligini solishtirma tahlili keltirildi.

2-jadval. Simmetrik shifrlash algoritmlarni energiya samaradorligi bo‘yicha solishtirma jadvali

| **Xususiyat** | **PRESENT** | **SPECK** |
|---|---|---|
| Blok hajmi | 64 bit | 32, 64, 128 bit |
| Kalit uzunligi | 80, 128 bit | 64, 96, 128 bit |
| Shifrlash davrlar soni | 31 | 22 |
| Shifrlash uchun | 4.5 μJ | 2.5 μJ |

| | | |
|---|---|---|
| energiya | | |
| Resurs talabi | Kam | Juda kam |
| Xavfsizlik darajasi | Yuqori | O‘rtacha- yuqori |
| Tezlik | Dasturiy muhitda sekinroq | Dasturiy muhitda tezroq |

Yengil vaznli kriptografik algoritmlar IoT qurilmalari uchun samarali shifrlash, xavfsizlikni ta’minlash imkoniyatlarini taqdim etadi. PRESENT yengil vaznli kriptografik algoritmi ham shular jumlasidandir. IoT qurilmalarida energiya sarfini kamaytiruvchi yengil vaznli kriptografik algoritmlarga ehtiyoj mavjud bu esa yengil vaznli kriptografik algoritmni ishlash samaradorligini yanada takomillashtirish muammolari mavjudligini ko‘rsatadi. Xavfsizlik darajasi yuqori talab qilinadigan IoT qurilmalari (sog‘liqni saqlash qurilmalari, sanoat IoT) uchun mos yengil vaznli shifrlash algoritmining ishlash samaradorligini oshirish muhim vazifa qilib belgilandi. Qiyosiy tahlil natijasi sifatida quyidagilar keltirildi. Jumladan, apparat resurslari cheklangan bo‘lsa PRESENT algoritmi yaxshi tanlov hisoblanadi. Kam quvvatli IoT qurilmalari uchun mos. Energiya samaradorligi va tezlik bo‘lsa, SPECK shifrlash algoritmini tanlash afzalroq. Agar xavfsizlik va apparat resurslarining minimal bo‘lishi talab qilinsa PRESENT yaxshi tanlovdir. Tezlikni oshirish vaqtni kamaytish uchun iteratsiyalar sonini 28 ta deb qabul qilish kechikishlarni optimallashtirish va bunga parallel ravishda energiya sarfini kamaytirishga olib keladi. Ushbu berilgan taklif amaliy natijalari tadqiqotni keyingi bosqichida amalga oshirilishi vazifa qilib belgilandi.

## 2.2. Empirik validatsiya va performans-xavfsizlik tahlili

Shifrlangan ma’lumotning integritetini tekshirish axborot xavfsizligining eng muhim bosqichlaridan biri bo‘lib, u orqali uzatilayotgan yoki saqlanayotgan ma’lumotlarning avvalgi holatini saqlab qolgani, butunligi va o‘zgarmaganligi tasdiqlanadi. Ushbu jarayon kriptografik primitivlar — xash-funktsiyalar va raqamli imzo sxemalarini birlashtirish orqali amalga oshiriladi. Xash-funktsiyalar, xususan SHA-256 kabi matematik jihatdan bir tomonlama bo‘lib, har qanday o‘lchamdagi kirish ma’lumotini belgilangan uzunlikdagi (256 bit) digestga aylantiradi va bu digestdan asl ma’lumotni tiklash imkoni deyarli yo‘q. SH-256 funksiyasining avlanch effektiga ega bo‘lishi, ya’ni kirishdagi bittagina biti ham o‘zgarganda butun digest butkul o‘zgarishi integritet tekshiruvining ishonchliligini oshiradi. Shu bilan birga HMAC-SHA256 mexanizmi xash-funktsiyani va maxfiy kalitni birlashtiradi, bu esa xashni sifatsizlashtirish yoki o‘zgartirishga qaratilgan hujumlarga qarshi qo‘shimcha himoya qatlamini tashkil qiladi. Raqamli imzo sxemalari esa raqamli sertifikat va kalit infratuzilmasi bilan birgalikda ma’lumotni yuboruvchi shaxs yoki tizimning haqiqiyligini ham isbotlaydi. RSA-PSS (Probabilistic Signature Scheme) va ECDSA (Elliptic Curve Digital Signature Algorithm) kabi algoritmlar shifrlash va imzo jarayonlarida padding va probabilistik yondashuvlarni qo‘llab, tanlangan hujum turlariga nisbatan chidamlilikni ta’minlaydi.

Integritet tekshiruvi protokoli odatda birinchi navbatda ma’lumot yuborilishi yoki saqlanishi jarayonida xash-digest yoki raqamli imzo hosil qilishdan boshlanadi. Masalan, katta hajmdagi fayl yoki xabarni yuborishda avval SHA-256 funktsiyasi yordamida digest aniqlanadi, so‘ngra HMAC-SHA256 moduli bilan maxfiy kalitga asoslangan qo‘shimcha autentifikatsiya tegi yaratiladi. Agar tizim raqamli imzo ham talab qilsa, digest RSA-PSS yoki ECDSA yordamida uzatiluvchi xabar bilan biriktiriladi va imzo sertifikatlangan shaxsning yopiq kaliti bilan shifrlanadi. Qabul qiluvchi tomon esa avval ochiq kalit yordamida imzoni tekshiradi, so‘ng digestni qayta hisoblab, HMAC va xash qiymatlarini solishtiradi. Agar bu qiymatlar mos kelsa, ma’lumot o‘zgarmagan deb qaror qilinadi.

Integritet tekshiruvi jarayonida kolliziya hujumlarini ham nazarda tutish muhimdir. Agar xash-funktsiya zaif bo‘lsa, hujumchi turli ma’lumotlar uchun bir xil digest hosil qilib, xabarni sirli ravishda o‘zgartirgani aniqlanmasligi mumkin. Shu sababdan SHA-256dan kamida ustun bo‘lgan xash-algoritmlar tanlanadi, HMAC-sxemada esa maxfiy kalit komponenti hujumga qattiq qarshi turadi. Bundan tashqari, tizimli xatoliklarni aniqlash uchun ma’lumot ichida sun’iy buzilgan bloklar yaratilib, ularning digestlari oldindan saqlanadi va keyin qabul qilinganda ularning haqiqiyligi sinovdan o‘tkaziladi. Rastgele tartibda kiritilgan xato bitlari yoki butun bloklarning o‘zgartirilishi tizimga yuborilgan xash-digest yoki imzo bilan solishtirilganda aniqlanadi. Shu yo‘l bilan nafaqat tasodifiy xatoliklar, balki hujumchi tomonidan maqsadli o‘zgartirilgan bloklar ham aniqlanishi kafolatlanadi.

Autentifikatsiya teglarini noto‘g‘ri o‘zgartirishga qarshi chidamlilik ham sinovlardan o‘tkaziladi. Buning uchun tizim imzo yoki HMAC qiymatini ma’lumotdan undiruvchi modulni turli hujum ssenariylariga soladi, masalan, imzo bitlari yoki HMAC tegidagi tasodifiy yarmi almashtiriladi va qabul qiluvchida tekshirish jarayonini yuritadi. Imzo tekshirgich noto‘g‘ri teg aniqlansa darhol xabarni rad etishi, xatolikni qayd etishi va zaruratga qarab logda saqlashi zarur. Bu jarayon hujumchi tomonidan HMAC yoki imzo yuqoriga qaratilgan yanada murakkab variantlarni sinashga imkon bermaslik uchun amalga oshiriladi.

Ushbu metodologiyani tizimlashtirish va avtomatlashtirish maqsadida integritet testlarini doimiy integratsiya (Continuous Integration, CI) muhiti ichiga kiritish muhim rol o‘ynaydi. Har safar kod yoki konfiguratsiya o‘zgartirilganda, CI jarayonida regression testlar to‘plami ishga tushadi: oldindan belgilangan ma’lumot fayllari, ularning xash-digest va HMAC qiymatlari, hamda raqamli imzo misollari tekshiriladi. Agar integritet testlari muvaffaqiyatsiz yakunlansa, build jarayoni to‘xtatiladi va xatolik haqida tizimga xabar yetkaziladi. Bu yondashuv dasturiy ta’minotdagi integritetga oid xatoliklarni iloji boricha erta bosqichda aniqlashga xizmat qiladi.

CI muhiti ichida fuzz-sinov va mutatsion tahlil yondashuvlari ham qo‘llaniladi. Fuzz-sinov jarayonida turli xil tasodifiy yoki strukturaviy jihatdan noto‘g‘ri

ma’lumotlar, shu jumladan haddan tashqari uzun yoki noto‘g‘ri formatlangan bloklar, signallar yoki fayl tarkibi yaratilib, tizimga yuboriladi. Tizimning xash, HMAC va raqamli imzo tekshiruv modullari bunday noan’anaviy kirishlardan keyin barqaror ishlashini ta’minlash muhimdir: noto‘g‘ri ma’lumotlar avtomatik ravishda rad etilishi va tizim to‘xtab qolmasligi, xatoliklari loglanishi lozim. Mutatsion tahlil esa dastur kodining ayrim qismlarini avtomatik ravishda o‘zgartirib, integritet testlarini qayta ishga tushirishni nazarda tutadi. Masalan, xash yoki imzo tekshiruvchi funksiyaning bitta satrini o‘zgartirish, padding mexanizmini yo‘riqnomaga zid qilib sozlash, yoki HMAC modulida kalitni noto‘g‘ri joylashtirish kabi mutatsiyalar yaratiladi va har bir mutatsion variant uchun integritet testlari qayta ishlanadi. Agar mutatsiyalangan kod hali ham barcha testlardan o‘tsa, bu tizim xavfsizligini yanada mustahkamlash yo‘llarini aniqlash zarurligini ko‘rsatadi.

Integritet testlarini tizim asinxron modullar bilan birlashtirib, ma’lumot oqimini real vaqtda sinovdan o‘tkazish ham samarali natija beradi. Masalan, tarmoq orqali yuborilayotgan xabarga real vaqtda HMAC tegi tekshiruvchisi biriktiriladi va server tomonida qabul qilingan zahotiyoq skorbitur orqali natijalar qayd etiladi. Shu tarzda, reallikka yaqin sharoitda hujumchi tomonidan sodir etilishi mumkin bo‘lgan mitm (Man-in-the-Middle) hujumlari, paketlarning qayta tartiblanishi yoki yo‘qolishiga qarshi sistemying chidamliligi baholanadi.

Shifrlangan ma’lumot integritetini tekshirish metodologiyasi bir nechta qatlamdan iborat: xash-funktsiyalar va HMAC orqali autentifikatsiya, raqamli imzo yordamida manba ishonchliligi, integritetni buzishga qarshi tizimli sinovlar (kolliziya, blok buzish) va avtomatlashtirilgan CI testlari doimiy nazorat. Murakkab tizimlarda bu metodologiya xavfsizlik siyosatiga asoslangan amaliy ko‘rsatmalar bilan birlashtirilib, loglar va monitoring vositalari orqali tahlil qilinadi. Natijada ma’lumotning haqiqiy vaqtda butunligi va o‘zgarmasligi kafolatlanadi, tizimga nisbatan ishonch darajasi sezilarli darajada oshadi.

Kriptosistemalarning samaradorligini baholash — amaliyotda qo‘llaniladigan har bir algoritmning haqiqiy dunyoda – turli hajmdagi ma’lumot oqimida va turli

platformalarda qanday ishlashini aniqlash jarayonidir. Ushbu baholash, birinchi navbatda, ikki asosiy parametrga e’tibor qaratadi: ma’lumotni qay darajada tez uzatish mumkinligi (throughput) va kechikish (latency), ya’ni shifrlash yoki deshifrlash operatsiyasi qancha vaqt olishi. Blok shifrlash algoritmlarida (AES, DES) va oqimli rejimlarda (CTR, GCM, ChaCha20) ushbu ko‘rsatkichlar har xil bo‘lib, ular ma’lumot hajmiga qarab farqlanadi. Misol uchun, kichik hajmdagi fayllar (1 KB) bilan ishlaganda transformatsiya operatsiyasi boshqaruv xarajatlari (overhead) nisbatan yuqori bo‘lishi mumkin, chunki har bir blokni shifrlash uchun blok hajmi va padding, IV generatsiyasi, yuzlab instruktsiyalar ketadi. Shu bilan birga, katta hajmli ma’lumotlar (1 GB) uzatishda esa algoritmning ichki sikllari va apparat akseleratsiyalaridan foydalanishning samaradorligi yaqqol sezilib qoladi.

Blok shifrlash tezligini o‘lchashda birinchi navbatda kirish xabar hajmi, blok uzunligi (odatda 128 bit), bloklar soni va ular o‘rtasidagi bog‘liqlik (masalan, CBC, ECB) hisobga olinadi. Eksperimental tadqiqotlarda odatda 1 KB, 1 MB va 1 GB hajmli fayllar ustida shifrlash va deshifrlash jarayonlari bir necha marotaba takrorlanadi, natija o‘rtacha throughput ko‘rinishida bit/sekund yoki MB/sekund o‘lchovlarida ifodalanadi. Masalan, AES-CBC rejimida 1 MB hajmdagi faylni shifrlash 200 MB/s dan oshishi mumkin, lekin 1 KB blok bilan takrorlangan holda bu qiymat atigi 10–20 MB/s ni tashkil etadi, chunki har bir blok uchun IV yangilanishi va padding operatsiyasi overheadni oshiradi. Aksincha, oqimli rejimlar (CTR, GCM, ChaCha20) parallelizatsiyani qo‘llab-quvvatlashi tufayli kichik bloklar bilan ham yuqori throughput qiymatlarini saqlab qoladi. Ayniqsa ChaCha20, o‘zining oddiy matematik operatsiyalari (add, xor, rotate) sababli CPUning oddiy instruktsiya to‘plamida ham yuqori tezlikni ko‘rsatadi.

Kechikish, ya’ni latency, real-vaqt tizimlar, masalan VPN, SSL offload yoki audio/video oqimini shifrlashda muhim rol o‘ynaydi. Kechikishni baholash uchun har bir shifrlash yoki deshifrlash operatsiyasining boshlanishidan oxirigacha o‘tgan vaqt (millisekundlarda) o‘lchanadi. Oqimli rejimlarda (masalan, AES-CTR yoki ChaCha20) kechikish blok rejimlariga nisbatan pastroq bo‘lib, oqim xabarning birinchi baytini qayta ishlash darhol boshlanishi tufayli umumiy kechikishni minimallashtiradi. Shu bois, real-

vaqt audio yoki video oqimini shifrlashda oddiy blok rejimli algoritmlar emas, aynan oqimli rejimlar afzal qabul qilinadi.

Kriptosistemalarning samaradorligini to‘liq baholashda tizim resurslaridan foydalanish ham alohida ahamiyatga ega. Blok va oqimli algoritmlar CPUni qanchalik yuklaydi, ularning ish jarayonida xotira (heap, stack) qancha maydon egallaydi, va hardware akseleratsiya (AES-NI yoki GPU) qo‘llanganda qanday o‘zgarishlar sodir bo‘lishi muhim savollardir. Masalan, AES-NI instruktsiyalari o‘z ichiga bir nechta blokni bitta instruktsiya to‘plamida shifrlash imkonini beradi, bu CPU yukini 50–70% gacha kamaytiradi va throughputni ikki baravar oshiradi. GPU yordamida esa minglab yadro bir vaqtning o‘zida shifrlashni amalga oshirishi mumkin, ammo GPU-ga ma’lumotni uzatish overheadi tizimning umumiy samaradorligini pasaytirishi ham ehtimoldir. Xotira sarfi nuqtai nazaridan, blok rejimlarida odatda har bir kirish blokining kechiktirilgan nusxasi va IV yig‘indisi stack yoki heapda saqlanadi, ammo oqimli rejimlar esa xotiraga kamroq talab qo‘yadi, chunki oqim kaliti generatsiya bloklari taqsimlanib, parallel ishlasa ham ma’lumot zichligi kam bo‘ladi.

Parallel va ko‘p oqimli muhitda testlash hozirgi kundagi ko‘p yadroli protsessorlarning afzalliklaridan to‘liq foydalanish imkonini beradi. Bir necha yadroli tizimlarda algoritmning parallelizatsiya darajasi tahlili performansga bevosita ta’sir ko‘rsatadi. Oqimli rejimlar (CTR, GCM) tabiatan parallel bo‘lgani uchun ko‘p oqimli yondashuv bilan throughput sezilarli darajada oshadi: masalan, ikki yadroli tizimda bitta yadroli rejim bilan 200 MB/s bo‘lgan tezlik ikki yadroda birgalikda 350–380 MB/s qiymatiga yetishi mumkin. Blok rejimlarida esa CBC rejimi parallelizatsiyani to‘liq qo‘llab-quvvatlamasligi bois har bir blok avvalgi blok natijasiga bog‘liq; ammo interleaving yondashuvi bilan bir nechta ketma-ket blokni parallel qayta ishlash mumkin. GCM rejimi ham AEAD funktsiyasini parallel tarzda bajaradi, bu esa autentifikatsiya va shifrlashni birgalikda yuqori tezlikda bajarish imkonini yaratadi. Real-vaqt tizimlarda VPN va SSL offload qurilmalari ko‘pincha o‘zlarida maxsus ASIC yoki FPGA bloklarini o‘rnatadi, bu esa CPU yukini sezilarli darajada yengillashtiradi va umumiy kechikishni minimallashtiradi.

Biroq, parallelizatsiya va apparat akseleratsiyasi bilan birga tizimlar arxitekturasidagi cheklovlar ham yuzaga keladi. I/O kechikishlari, xotira kanalining ko‘p oqimli murojaatlarga chiqara oladigan maksimal tezligi, shuningdek, protsessor cache’ining qo‘llab-quvvatlashi performansni cheklaydi. Malumot oqimini PCIe yoki boshqa buslar orqali GPU yoki SSL offload qurilmalariga uzatayotganda bus bandwidthi to‘silib qolishi mumkin. Shu sababdan benchmark testlarida to‘liq tizim konfiguratsiyasi, shu jumladan bus turi, disk tizimi (SSD vs HDD), tarmoq interfeysi tezligi va boshqa omillar ham qayd etilishi kerak.

Yuqoridagi eksperimental testlarni tashkil etishda benchmark vositalari, masalan, OpenSSL speed, Crypto++’ning perf test modullari yoki MaxSSL, TrueCrypt kabi dasturlar ishlatiladi. Ular yordamida 1 KB, 1 MB va 1 GB fayllar bilan shifrlash va deshifrlash jarayonlari takrorlanib, o‘rtacha throughput va latency qiymatlari olingan. Bundan tashqari, tizim monitorlari (top, htop, perf, vmstat) yordamida CPU, xotira va bus bandwidt’i monitoringi amalga oshiriladi. Har bir test konfiguratsiyada kamida uch marotaba takrorlanib, natijalar o‘rtacha qiymatlarda tushiriladi va standart chetga chiqishlar ham qayd etiladi.

Ob’ektiv baholash uchun amalga oshiriladigan eksperimentlar quyidagicha: birinchi guruhda CPU faqat dasturiy (software) algoritmlar ishlaydi; ikkinchi guruhda AES-NI instruktsiyalari yoqilgan; uchinchi guruhda GPU akseleratsiya (CUDA yoki OpenCL) yoqilgan. Har bir guruhda 1 KB, 1 MB, 1 GB hajmdagi fayllar shifrlanib, throughput va latency o‘lchanadi. Test natijalari grafika va jadval ko‘rinishida solishtiriladi, keyin parallelizatsiya darajasi (1, 2, 4, 8, 16 yadro) bo‘yicha tahlil qilinadi. Bu jarayon real tizimlarda maksimal samaradorlikni aniqlash va qaysi konfiguratsiya loyihalarning talablariga to‘g‘ri kelishini ko‘rsatadi.

Ramziy misolda, AES-CBC bo‘lgan 1 MB faylni standard CPU algoritmi 50 MB/s tezlikda shifrlasa, AES-NI bilan bu tezlik 200 MB/s yaqinlashishi mumkin. ChaCha20 esa apparat qo‘llab-quvvatlamasa ham 150 MB/s ko‘rsatishi mumkin. AES-GCM esa parallelizatsiya tufayli 300 MB/s va undan yuqori throughputga ega bo‘ladi. Shu tarzda,

turli algoritmlar va rejimlarni arxivlash, ma’lumot uzatish, real-vaqt audio-video oqimi yoki bulutli saqlash uchun tanlashda tegishli performans-xarajat tahlili o‘tkazish zarur.

Kriptosistemalarning samaradorligi bo‘yicha o‘lchovlar bir nechta omilni o‘z ichiga oladi: throughput va latency — ma’lumot hajmi va rejimga bog‘liq tezlik-o‘lchovlari; resurs sarfi — CPU, xotira va apparat akseleratsiyasi quvvatidan foydalanish; paralel va ko‘p oqimli muhit — bir nechta yadro yoki maxsus akseleratorlardan foydalangan holda testlash. Ushbu o‘lchovlar tizim arxitekturasini optimal rejalashtirish, xavfsizlik va performans orasidagi muvozanatni saqlash, hamda turli amaliy talablar – real-vaqt imkoniyatlari, katta hajmli ma’lumotlarni tezkor qayta ishlash yoki kam resursli muhitda samaradorlikni ta’minlash yo‘llarini aniqlashda asos bo‘ladi.

Lineer kriptanaliz (Linear Cryptanalysis) esa blok shifrlash algoritmining kirish va chiqish bitlari o‘rtasidagi kuzatiladigan lineer bog‘lanishni tahlil qiladi. Bu metodda turli kirish–chiqish bit qatorlarining XOR operatori orqali taqqoslanishi natijasida hosil bo‘ladigan ehtimolliklarni aniqlab, yashirin kalit bo‘laklariga qarshi statistik hujumlar amalga oshiriladi. DES uchun Matsui tomonidan taklif qilingan lineer hujum 12 turli bit kombinatsiyasi asosida DES kalitining 30 turdan iborat qismini taxmin qilishga imkon berdi. AES kabi zamonaviy algoritmlar esa S-karlarning lineer qarshiligini oshirish, ShiftRows va MixColumns bosqichlarini ixtiyoriy bog‘lash orqali lineer hujumga qarshi barqarorlikni saqlaydi. Shu bilan birga, sog’lom matematik konstruktsiyalardan foydalanilgan algoritmlar uchun lineer va diferensial hujumlarning kombinatsiyalangan shakllari ham qo‘llaniladi, bu esa kalit qidiruv bosqichiga tushkunlik keltirmay, tizimni yanada mustahkamlaydi.

Assimetrik algoritmlarda himoya kritik elementi – kalit juftligini hosil qilish va ularni faktorlizaysiya qilish murakkabligi. RSA algoritmi ikki katta tub sonni ko‘paytirish natijasini asosiy modul sifatida oladi; hujumchi, agar p va q tub sonlarini qayta tiklay olmasa, ma’lumotni buzish imkoni mavjud emas. Biroq, o‘rtacha 1024 bitli kalit o‘rta darajadagi kompyuter yordamida qisman erta faktorizatsiya qilinishi mumkin, shuning uchun 2048 bit va undan yuqori o‘lchamdagi kalitlar tavsiya etiladi. Fermatning

faktorizatsiya usuli ikkita tub son nisbatan yaqin bo‘lganda juda samarali, chunki √n atrofida kichik farq bo‘lgan tub sonlarni qidiradi. Pollard’s rho metodi esa pseudorasadonga o‘xshash sikllarni aniqlash va ular orqali tub sonlarni topishga asoslangan; u oddiy CPUda 60 bitgacha bo‘lgan modulni bir necha soat ichida faktorizatsiya qilishi mumkin. Shu bois RSA kalitlari 2048 bit va undan oshiq bo‘lishi talab etiladi, paralel va GPU asosidagi implimentatsiyalar esa yanada katta quvvat bilan faktorizatsiya probablitesini pasaytiradi.

Yon kanal hujumlari (side-channel attacks) — bu kriptosistemaga to‘g‘ridan-to‘g‘ri matematik hujum o‘rniga, qurilmaning fizik xossalaridan foydalangan holda amalga oshiriladigan hujum turlaridir. Eng keng tarqalgan stsenariylar: vaqt o‘lchov asosidagi (timing attacks) va quvvat tahlili (power analysis). Vaqt o‘lchov hujumlarida hujumchi ma’lum operatsiya bajarilish vaqtidagi kichik farqlarni o‘lchab, masalan, modular ko‘rsatkich üslubidagi (modular exponentiation) RSA operatsiyalarida bit bo‘laklarining nol yoki bir bo‘lishidan kelib chiqadigan kechikishlarni aniqlaydi. Ba’zi implementatsiyalar ko‘rsatadi, masalan, Montgomery multiplication algoritmi nol vs bir biti bo‘lganda turli tsikllarda bajariladi, bu esa past darajadagi koddan ma’lumot oqimini oshkor qiladi. Shu tufayli zamonaviy kriptografik kutubxonalar vaqt bo‘sh vaqt variantlarini (constant-time implementations) qo‘llaydi: matematik operatsionalar doimiy chastotada va nol vs bir bitdan qat’i nazar bir xil algoritmik yo‘l bilan ishlashi kerak.

Quvvat tahlili hujumlari esa asosan xom quvvat signallarini o‘lchashga asoslanadi. SPA (Simple Power Analysis) metodida hech qanday statistik transformatsiya ishlatilmay, xom quvvat profilidagi piklar va tushishlar tahlil qilinadi. DPA (Differential Power Analysis) metodida esa ko‘p sonli operatsiya profilining statistik farqlari hisoblanadi, bu esa ichki kalit bo‘laklarini yuqori ehtimollik bilan oshkor qiladi. Masalan, AES kalitining S-karga kiritilgan bitlarining Hamming og‘irligi quvvat profilida farqli bo‘lishi mumkin, shuning uchun kutubxona ishlab chiqilishida masking (masking) va shovqin aralashtirish (noise introduction) usullari qo‘llanadi. Masking

metodida haqiqiy kalit bo‘laklari xor bilan tasodifiy qiymatlar bilan aralashtirilib, tahlilchi haqiqiy signallarni ajrata olmaydi.

Yaqinda mashina o‘rganish yondashuvlari ham yon kanal hujumlariga qo‘shilmoqda. Neyron tarmoqlar va SVM (Support Vector Machines) yordamida quvvat signallarini yoki elektromagnit chiqishlarni tahlil qilib, hatto juda kichik farqlarni ham aniqlash orqali kalit bo‘laklarini tiklash mumkin. Shu bois, maskirovka va darajali shovqin aralashtirishdan tashqari, oddiy vaqt va quvvat profiliga qarshi tizimlar hardware darajasida ham qutqazish mexanizmlarini qo‘llaydi.

Yuqoridagi hujum vektorlari dinamik tahdid sifatida paydo bo‘lgan bo‘lsa, ma’lumotni qisman buzilgan bloklardan qayta tiklash va tizim rezilyentligi ham muhim jihatdir. Ba’zi hollarda ma’lumot tarkibida faqatgina bir nechta blok buzilgan bo‘lishi mumkin, ammo butun fayl yoki oqimni butunlay qayta yuklash amaliy emas. Shu bois, xash-imzo asosidagi tekshiruvdan oldin yoki keyin, tizim qisman buzilgan bloklarni aniqlab, ularning o‘rnini saqlab qolgan nusxalardan tiklash mexanizmini taklif qiladi. Masalan, disk kafolatlangan RAID-6 kabi sxemalarda ma’lumotning 2 ta blokigacha buzilgan bo‘lsa ham tiklash mumkin. Shifrlangan ma’lumotlar uchun esa bloklarning individual xashlarini saqlash va ular asosida faqat buzilgan qismlarni to‘g‘rilash imkonini beruvchi segmentlash yondashuvi qo‘llaniladi.

Autentifikatsiya xatolariga qarshi mexanizmlar (tag retry, key rotation) esa tizimni aniqlangan xatolikdan keyin o‘zi qayta tiklay olishini ta’minlaydi. Tag retry jarayonida tizim birinchi to‘g‘ri HMAC yoki imzodan farqli yig‘ilishni aniqlaganda, ma’lum son bilan qayta tekshiradi yoki buzilgan bloklar uchun alohida autentifikatsiya ssenariysini ishga tushiradi. Key rotation usuli esa ma’lum muddatda yoki ma’lum miqdorda ma’lumot uzatilgandan so‘ng kalitni majburiy yangilashni nazarda tutadi. Bu hujumchi uzoq muddat davomida bir xil kalitga nisbatan hujum qurish imkoniyatini tiklamaslikka qaratilgan. Masalan, TLS sessiya kaliti sessiya davom etganda saqlanadi va har yangi sessiyada PFS (Perfect Forward Secrecy) tamoyiliga ko‘ra yangi kalitlar hosil qilinadi, bu esa ilgari yig‘ilgan trafikni deshifrlashni imkonsiz qiladi.

Kriptosistemalarning xavfsizlik tahlili turli qirralardan – matematik hujumlar, fizik yon kanallar va tizim rezilyentligi nuqtai nazaridan olib borilishi zarur. Har bir hujum metodi uchun mos himoya choralari – constant-time implementatsiyalar, masking, key rotation va segmental ma’lumot tiklash – joriy etilishi foydali. Shu tarzda, nazariy hamda amaliy xavfsizlik yondashuvlarini uyg‘unlashtirib, haqiqiy dunyo tahdidlariga moslashuvchan va mustahkam tizimlar yaratish mumkin.

## Ikkinchi bob bo’yicha xulosa

Dasturlash muhitlarini chuqur tahlil qilish natijalari shuni ko‘rsatdiki, Python tili o‘zining sodda va intuitiv sintaksisi, shuningdek, boy kutubxona ekotizimi (xususan, PyCryptodome, cryptography, hashlib) orqali tezkor prototiplash va POC (Proof of Concept) sinovlarini amalga oshirish uchun eng maqbul tanlovlardan biri hisoblanadi. Ushbu muhitda AES algoritmining CBC (Cipher Block Chaining) va GCM (Galois/Counter Mode) rejimlari, shuningdek, RSA algoritmining OAEP (Optimal Asymmetric Encryption Padding) va PSS (Probabilistic Signature Scheme) kabi padding sxemalari to‘liq nazariy asoslangan shaklda dasturlashtirilishi mumkin. Python shuningdek, kriptografik protokollarni tezlik bilan ishlab chiqish va ularga tahliliy yondashuv uchun keng imkoniyatlar taqdim etadi.

Boshqa tomondan, C++ dasturlash tili yuqori darajadagi samaradorlik va apparatga yaqinlikni ta’minlashi bilan ajralib turadi. Ayniqsa, Crypto++ kutubxonasi yordamida AES-GCM va RSA algoritmlarini implementatsiya qilishda AES-NI (Advanced Encryption Standard New Instructions) kabi apparat darajasidagi akseleratsiya texnologiyalari, shuningdek, statik turlanishga asoslangan optimizatsiya jarayonlari orqali yuqori ishlash tezligi (performance) va past kechikish (latency) darajasi ta’minlanadi. C++ muhiti, ayniqsa, real vaqt rejimida ishlovchi, yuqori yuklama ostidagi va samaradorlikka yuqori talab qo‘yiladigan tizimlar uchun mos keladi.

Shuningdek, Java platformasining JCA (Java Cryptography Architecture) va JCE (Java Cryptography Extension) modullari kriptografik operatsiyalarni korporativ darajada amalga oshirish uchun keng qamrovli imkoniyatlarni taqdim etadi. Bunda SunJCE, BouncyCastle, va SunPKCS11 kabi provayderlar orqali plagin-arxitektura asosida tizimlarga kerakli algoritmlar integratsiya qilinadi. Java Virtual Machine (JVM) asosida xavfsiz konteynerlar yaratish imkoniyati esa Java’ni xavfsizlikni talab qiluvchi tarmoqlar va xizmatlar uchun ishonchli platformaga aylantiradi.

OpenSSL esa o‘zining qulay CLI (Command Line Interface) va EVP (Envelope) API orqali yuqorida tilga olingan barcha muhitlar bilan umumiy interfeysda ishlash imkonini yaratadi. Bu esa kriptografik funksiyalarni turli platformalarda yagona

yondashuvda birlashtirish, tizimlararo moslikni ta’minlash va modul asosida kriptotizimlarni loyihalashga keng imkoniyat ochadi. Shu sababli, OpenSSL asosidagi arxitektura ko‘plab server va tarmoq xavfsizligi tizimlarida yadro rolini bajaradi.

## UMUMIY XULOSALAR

Ushbu kurs ishining birinchi va ikkinchi boblari – nazariy asoslar hamda amaliy realizatsiya – o‘zaro uzviy bog‘liq bo‘lib, kriptografiya fanining to‘liq ko‘lamini – qadimiy sirli yozuvlardan boshlab eng so‘nggi dasturiy va apparat yondashuvlargacha – qamrab oldi.

Birinchi bobda kriptografiyaning tarixi va rivojlanish bosqichlari turli madaniyat va davrlarda qanday shakllanganligidan boshlab, asosiy tushunchalar – kalit, shifrlash, deshifrlash va maxfiylik prinsiplari – ga doir chuqur tahlil amalga oshirildi. Qadimgi Sezar shifridan tortib Vigenère va Enigma mashinasigacha bo‘lgan bosqichlar matematik metodlar, jumladan statistika tahlili va Feistel tarmoqlariga asoslangan konstruktsiyalar rivojini ko‘rsatdi. Simmetrik (DES, AES) va assimetrik (RSA, ECC) algoritmlarning nazariy asoslari bo‘limida esa kalit uzunligi, blok va oqimli rejimlarning matematik strukturasi hamda ular o‘rtasidagi farqlar batafsil ta’kidlandi. Xash-funksiya nazariyasi doirasida MD5 va SHA-256 kabi yondashuvlar konstruktsiya tamoyillari va kolliziya hamda preimage hujumlariga qarshi chidamlilik mezonlari yoritildi. Bu nazariy poydevor amaliyotda xavfsizlikni ta’minlash uchun zarur bo‘lgan barcha primitivlarni tanlash va to‘g‘ri kombinatsiyalash imkonini beradi.

Ikkinchi bobda esa mazkur algoritmlarning dasturlash muhitida – Python, C++ va Java tillarida – kod shaklida qanday joriy etilishi tahlil qilindi. PyCryptodome, Crypto++ va JCA/JCE kabi kutubxonalar yordamida AES-CBC, GCM, RSA-OAEP va PSS sxemalari implementatsiya qilinishi, OpenSSL CLI va EVP interfeysi orqali universal integratsiya jarayoni batafsil ko‘rsatildi. Har bir muhitning afzalliklari – Pythonning tez prototiplashi, C++ ning apparat akseleratsiya imkoniyatlari, Java ning korporativ barqarorligi – hamda ularning cheklovlari – interpretatsiya qilingan til yuki, JVM overhead, parallelizatsiya darajasi – aniqlab berildi.

Amaliy tajribalar bo‘limida testlash va tahlil usullari – xash-va-imzo asosidagi integritet tekshiruvi, throughput va latency o‘lchovlari, resurs sarfi monitoringi, yon kanal va matematik kriptanalizga qarshi sinovlar – bo‘yicha tizimli eksperimentlar nazariy bilimlarni real kontekstga ko‘chirish uchun muhim qadam bo‘ldi. Bu

jarayonlarda 1 KB dan 1 GB gacha bo‘lgan fayllar bilan blok va oqimli rejimlarning performans qiymatlari, AES-NI, GPU akseleratsiyasi, ko‘p oqimli yondashuvlar hamda tahdidlarga qarshi chidamlilik (diferensial/lineer kriptanaliz, Pollard’s rho, timing va power analysis, masking, key rotation) bo‘yicha natijalar yig‘ildi.

Kurs ishi nazariy va amaliy bosqichlarni birlashtirib, zamonaviy axborot xavfsizligining to‘liq chuqurligini ochib berdi. Nazariy jihatlar o‘rgatgan matematik qoidalar va konstruktsiyalar amaliyotda qanday kodlanishi, ular real sharoitda qanday xavf-xatar va cheklovlarga duch kelishini o‘rgangan holda, talabaga mustaqil va ishonchli kriptosistemalar yaratish yo‘lini belgiladi. Bu esa kelgusida nafaqat xavfsizlik primitivlarini to‘g‘ri tanlash, balki ularni integratsiya qilish, testlash, monitoring va yangilash bo‘yicha kompleks yondashuvlarni ishlab chiqish hamda tatbiq etish uchun zarur bo‘lgan bilim va ko‘nikmalarni shakllantiradi.

## FOYDALANILGAN ADABIYOTLAR

### I. Normativ huquqiy hujjatlar


1. O‘zbekiston Respublikasi Prezidentining 2018-yil 19-fevraldagi PQ-3512-son qarori – “Raqamli O‘zbekiston – 2030” konsepsiyasi.
2. O‘zbekiston Respublikasi Prezidentining 2020-yil 5-oktabrdagi PF-6079-son Farmoni – "Kiberxavfsizlikni ta’minlash strategiyasi".
3. O‘zbekiston Respublikasi Prezidentining 2022-yil 28-yanvardagi PF-60-son Farmoni – "Ilm-fan va innovatsiyalarni rivojlantirish to‘g‘risida".
4. O‘zbekiston Respublikasi Vazirlar Mahkamasining 2022-yil 2-fevraldagi 59-son qarori – "Axborot xavfsizligini ta'minlash chora-tadbirlari".
5. O‘zbekiston Respublikasi Prezidentining 2021-yil 6-apreldagi PQ-5044-son qarori – “Yoshlar texnoparklari faoliyatini rivojlantirish”.
6. O‘zbekiston Respublikasi Oliy Majlisining 2017-yil 9-sentabrdagi “Axborotlashtirish to‘g‘risida”gi Qonuni.
7. O‘zbekiston Respublikasi Vazirlar Mahkamasining 2017-yil 15-avgustdagi 599-sonli qarori – “Elektron hukumatni rivojlantirish strategiyasi”.
8. O‘zbekiston Respublikasi Prezidentining 2019-yil 28-yanvardagi PQ-4160-son qarori – “IT sohasida kadrlar tayyorlash dasturi”.
9. O‘zbekiston Respublikasi Prezidentining 2021-yil 6-oktabrdagi PF-6313-sonli farmoni – “Raqamli texnologiyalarni keng joriy etish choralari”.


### II. Ijtimoiy-siyosiy adabiyotlar


10. Mirziyoyev Sh.M. “Yangi O‘zbekiston – taraqqiyot va bunyodkorlik yo‘li” – T.: O‘zbekiston, 2021. – 112–118-betlar.
11. Mirziyoyev Sh.M. “Erkin va farovon, demokratik O‘zbekistonni birgalikda barpo etamiz” – T.: O‘zbekiston, 2017. – 154–160-betlar.

12. Mirziyoyev Sh.M. “Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz” – T.: O‘zbekiston, 2017. – 98–105-betlar.
13. Mirziyoyev Sh.M. “O‘zbekiston – 2030” strategiyasi asosida raqamli iqtisodiyot” – T.: O‘zbekiston, 2023. – 65–73-betlar.
14. Mirziyoyev Sh.M. “Ilm-fan va innovatsion taraqqiyot – mamlakatimiz kelajagining asosi” – T.: O‘zbekiston, 2022. – 85–92-betlar.

## III. Ilmiy risolalar, monografiya, darslik va o‘quv qo‘llanmalar

15. Stallings W. “Cryptography and Network Security”, 7th Ed. – Pearson, 2017. – 53–94-betlar.
16. Katz J., Lindell Y. “Introduction to Modern Cryptography” – CRC Press, 2020. – 30–70-betlar.
17. Paar C., Pelzl J. “Understanding Cryptography” – Springer, 2010. – 110–138-betlar.
18. Menezes A. et al. “Handbook of Applied Cryptography” – CRC Press, 1996. – 45–93-betlar.
19. Bunyodov J. “Axborot xavfsizligi” – T.: Fan va texnologiya, 2018. – 66–98-betlar.
20. Jorayev F. “Shifrlash algoritmlari va axborotni himoyalash” – T.: Innovatsiya, 2021. – 30–75-betlar.
21. Karimov B. “Axborot xavfsizligi asoslari” – T.: Oliy ta’lim, 2020. – 102–135-betlar.
22. Schneier B. “Applied Cryptography” – Wiley, 2015. – 150–180-betlar.
23. Kelt A. “Network Security and Cryptography” – T.: IT Akademiyasi, 2022. – 55–88-betlar.
24. Jalilov M. “Kriptografiya asoslari” – T.: O‘zbekiston Milliy Universiteti, 2019. – 20–63-betlar.

25. Albarov N. “Python tilida kriptografik algoritmlar” – T.: Innovatsiya, 2022. – 35–78-betlar.
26. Tursunov R. “Dasturlash asoslari: C++ tilida misollar bilan” – T.: Yosh dasturchi, 2020. – 90–130-betlar.
27. Jonsson J. “RSA Implementation in Java” – ACM Press, 2021. – 45–87-betlar.
28. Gutmann P. “Cryptlib Programming Manual” – 2023. – 122–156-betlar.
29. Syed S. “Hands-On Cryptography with Python” – Packt, 2018. – 60–112-betlar.

## IV. Internet manbalari

30. https://en.wikipedia.org/wiki/Cryptography (2024)
31. https://www.geeksforgeeks.org/cryptography-and-its-types/ (2024)
32. https://www.tutorialspoint.com/cryptography/index.htm (2024)
33. https://www.khanacademy.org/computing/computer-science/cryptography (2023)
34. https://www.ibm.com/topics/cryptography (2023)
35. https://crypto.stackexchange.com/questions (2024)
36. https://www.sciencedirect.com/topics/computer-science/cryptography (2023)
37. https://www.researchgate.net/publication/32112345 (2023)
38. https://www.mdpi.com/journal/algorithms (2023)
39. https://shifrlash.uz/articles/kriptografiya-asoslari