

OPENSSL KUTUBXONASIDA KALIT GENERATSIYALASH USULLARI VA ALGORITMLARI

Yusupova S.M., Amonov A.H., Abdullayev I.K., Avazbekov M.A

Toshkent axborot texnologiyalari universiteti

abdujabbor.madina.1989@gmail.com

Anatatsiya: OpenSSL – bu har tomonlama kriptografiya kutubxonasi bo'lib, u TLS (Transport Layer Security – Transport qatlami xavfsizligi) protokolining ochiq manbali ilovasini taklif qiladi. Bu foydalanuvchilarga SSL (Secure Sockets Layer – ularish qatlami xavfsizligi) bilan bog'liq turli vazifalarni, jumladan [CSR \(Certificate Signing Request – Sertifikat imzolash so'rovi\)](#) va shaxsiy kalitlarni yaratish va SSL sertifikatini o'rnatish imkonini beradi.

Kalit so'zlar: OpenSSL kutubxonasi, maxfiy kalit, ochiq kalit, parametr, RSA algoritmi, DSA algoritmi, DH parametri.

Reja:

IX. Kirish

X. Asosiy qism

7. RSA algoritmi yordamida kalit generatsiyalash.
8. DSA algoritmi yordamida kalit generatsiyalash.
9. DH algoritmi yordamida kalit generatsiyalash.

XI. Xulosa

XII. Foydalilanigan adabiyotlar

Kirish

Yigirma birinchi asr axborotlashtirish asri ekaniga tobora ko'pchilik ishonch hosil qilmoqda. Bu albatta ommaviy axborot va hamma bilishi mumkin va zarur bo'lgan axborot haqida gap borganda o'ta ijobjiydir. Lekin konfidensial va o'ta maxfiy axborot oqimlari uchun zamonaviy axborot-kommunikasiya texnologiyalari qulayliklar bilan bir qatorda yangi muammolarni o'rtaga qo'yemoqda. Axborot bazalarida saqlanadigan va telekommunikasiya tizimlarida aylanayotgan axborot xavfsizligiga tahdid keskin oshdi. Keyingi vaqtida, ayniqsa, Internet paydo bo'lidan boshlab, axborot o'g'irlash, axborot mazmunini buzib qo'yish, egasidan iznsiz o'zgartirib qo'yish, tarmoq va serverlardan beruxsat foydalanish, tarmoqqa tajovuz qilish, avval qo'lga kiritilgan uzatmalarni qayta uzatish, xizmatdan

yoki axborotga daxldorlikdan bo'yin tov lash, jo'natmalarni ruxsat etilmagan yo'l orqali jo'natish hollari ko'paydi.

Natijada axborot xavfsizligi muammosi O'zbekiston Respublikasi uchun ham dolzarb muammoga aylandi. Bu o'z navbatida kriptologiya fanini rivojlantirish vazifalarini dolzarb muammolar qatoriga qo'ydi, chunki hozirgi kunda bu yo'l axborot xavfsizligini ta'minlash sohasida asosiy yo'l idir.

Axborotni muhofaza qilish masalalari bilan kriptologiya fani shug'ullanadi. Keyingi oxirigi yillarda kriptologiya yo'nalishini rivojlantirishga davlatimiz tomonidan katta ahamiyat berilmoqda. O'zbekiston Respublikasi Prezidentining 2007 yil 3 aprelda qabul qilgan "O'zbekiston Respublikasida axborotning kriptografik himoyasini tashkil etish chora-tadbirlari to'g'risida" gi PQ-614-son qarorida hamda O'zbekiston Respublikasi Prezidentining 2017 yil 7 fevraldag'i "O'zbekiston Respublikasini yanada rivojlantirish bo'yicha Harakatlar strategiyasi to'g'risida" gi PF-4947-son farmoyishida beshta ustuvor yo'nalishdan biri sifatida axborotni muhofaza qilish tizimini takomillashtirish, axborot sohasidagi tahdidlarga o'z vaqtida va munosib qarshilik ko'rsatish kabilar ko'zda tutilgan.

RSA algoritmi yordamida kalit generatsiyalash

OpenSSL kutubxonadan foydalangan holda maxfiy hamda ochiq kalitlarni shuningdek, raqamli sertifikatni hosil qilish mumkin.

1-qadam. Maxfiy ya'ni shaxsiy kalitni yarash va ko'rish (Windows da).

1) OpenSSL dasturini o'rnatib, unga kiramiz va << genrsa -aes128 –out Surayyo_private.pem 1024>> deb yozamiz (1-rasm). Bunda "genrsa –aes128" – RSA algoritmi orqali AES128 simmetrik blokli shifrlash algoritni yordamida kalit generatsiya qilinadi, kalit uzunligi 1024bit bo'lib, uni [512;4096] oraliqda tanlash mumkin. Hosil qilingan shaxsiy kalit Surayyo_private.pem fayliga joylashtiriladi (bunda fayl nomi ixtiyorli tanlanadi)

```
OpenSSL> genrsa -aes128 -out Surayyo_private.pem 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
.+++++
.....+++++
e is 65537 (0x010001)
```

1-rasm

2) Yaratilgan shaxsiy kalitga kirish uchun kirish paroli o'rnatiladi va tasdiqlash uchun takrorlanadi (2-rasm).

```
e is 65537 (0x010001)
Enter pass phrase for Surayyo_private.pem:
Verifying - Enter pass phrase for Surayyo_private.pem:
OpenSSL>
```

2-rasm

3) Yaratilgan shaxsiy kalit OpenSSL dasturi joylashgan papkaga joylashadi. Uni ko'rish uchun OpenSSL joylashgan papkadan Surayyo_private nomli faylni ochib ko'rish mumkin (3-rasm).

```
Surayyo_private - Блокнот
Файл Правка Формат Вид Справка
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,7F0880AC2E1715E5B38F6ADA51AFD095

JbwFOBZMB8DlURmI/d1luigYEY6YSSn48tIIok2a6R7EcZBeW3E0Lvw2XD87dk8n
+Twd5ah0MdVCw3BfCLKcAoP4Kps6RUYS6XB0fmRnq00mGldBLQ7v1kpgLCy13Gsj
vKK1cW2DyVRV0tGJLjn2uKxzTmY/mDZigOcK+hDCviai8fWa0TV+RYzMC1lQ+++f
s1HZhwoxZeX0e2xjaFmwaopafpn999fbdbBeCbFQe/2UFGI9G6o5jjuIx33LZey+N
SPxPGyy53URZs8/5h0VwJ7VImrkdLv+zB/I7fcXmplSOsIgjfAKJVAGYSVGxhSeI
UQhEJOCFdldqnKLqY5jszyuPolV0HYJ961cBndsrLZuhWUFwX8J3o/Sr8b40hNM
XqPUif1CM9vLgCHkkJA094hfd9nCVJMPntHk+hvu1jANKh3ZLKecGBwoV4Tb096H
33mEKBxLRpct+Xy5HBbuosBrEKEwS+gqy1AeL7zgCF14KfgB0k2TzJu22IDp8BZ5
z37wu5Byk/iD4EATmfXoq8lUSWs/05/wTz18pvKqJvUPCn/i2qb3izym5knUDDnA
JyGd7fJi3zXvItgGxZ0SO/Z+hVKZ4-8Tj5w4Boyd8+Xi30DPCUk/kNMVCpdIOvhab
8ESw7YArEC5xb1A6T2C+pNlBLBx/re4y/2TjmqruiAqh7ZGkxd3UTHc1QVCNC1Qn
NvNAAsddD4d2LDao+DBa1ZzTpAatFZG4RL2uu6ooB5W0CXbwpgaxI/O/YJmhE/Lyc
qDeCeJmTCY0McMclOjknZvhbJh3QUWZLsW8F580xstYHUza9qDCK9gE9Fu3yADKE
-----END RSA PRIVATE KEY-----
```

3-rasm

2-qadam. Ochiq kalitni yaratish va ko'rish (Windows da).

1) <<rsa -in Surayyo_private.pem -pubout -out Surayyo_public.pem>>

Yuqoridagi buyruq orqali ochiq kalit yaratiladi. Bunda RSA algoritmi yordamida Surayyo_private.pem (yaratilgan shaxsiy kalit) dan foydalangan holda ochiq kalit yaratiladi va Surayyo_public.pem (ixtiyoriy nomli) faylga joylashtiriladi (4-rasm).

Yuqoridagi buyruqdan so'ng shaxsiy kalitga kirish paroli so'raladi va parol to'g'ri kiritilsa, ochiq kalit yaratiladi, aks holda yaratilmaydi.

```
Verifying - Enter pass phrase for Surayyo_private.pem:
OpenSSL> rsa -in Surayyo_private.pem -pubout -out Surayyo_public.pem
Enter pass phrase for Surayyo_private.pem:
writing RSA key
OpenSSL>
```

4-rasm

2) Ochiq kalitni ko'rish uchun OpenSSL joylashgan papkadan Surayyo_public nomli fayl ochib ko'rildi (5-rasm).

```
Surayyo_public - Блокнот
Файл Правка Формат Вид Справка
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCaURLgPRbXr21bZzRbwMFKs7jE
atYo2wXydjSF8ukmBQ1rZG92Y2eH8N0HU+Zkj+lT6jw6mMTjb6gAQD95iAybcknC
meDAcS+nhomppzh9XZmDdq1rtMrllEH1A4vhGwOyIuk7G8Ak5Qgk6XACw22D81vXa
rYIdD2ryb8yl8uspjQIDAQAB
-----END PUBLIC KEY-----
```

5-rasm

DSA algoritmi yordamida kalit generatsiyalash

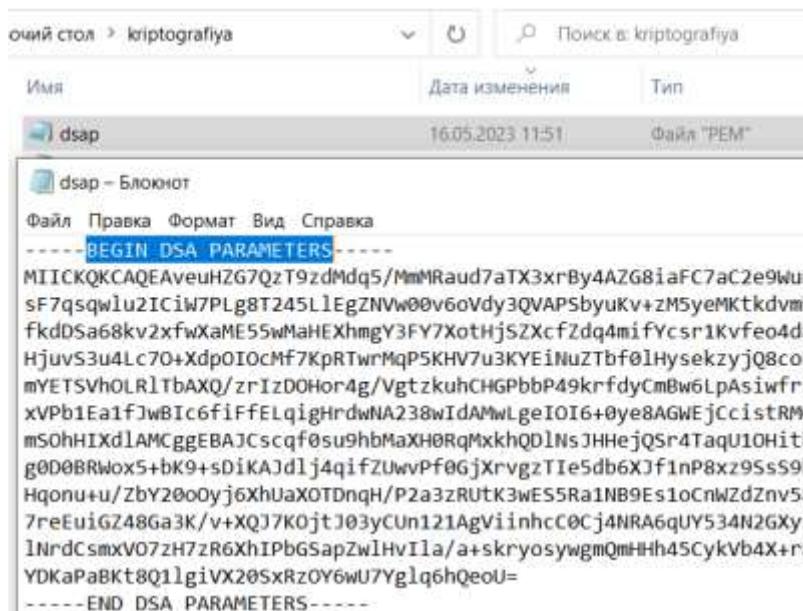
1) DSA parametrini hosil qilish(6-rasm).

```
<openssl genpkey -genparam -algorithm DSA -out dsap.pem> - 2048
```

bitli dastlabki, DSA parametrini hosil qilish, bunda dsap.pem DSA parameter saqlagan fayl nomi.

6-rasm. DSA parametrini hosil qilish.

2) yaratilgan faylni kriptografiya nomli papkaga kirib ko'rish mumkin(7-rqsm).



7-rasm. DSA pametrining ko'rinishi.

7-rasmdan ko'rish mumkinki, ochilgan faylning yuqori qatorida BEGIN DSA PARAMETERS – DSA parametr hosil qilingan ekanligini ko'rsatadi.

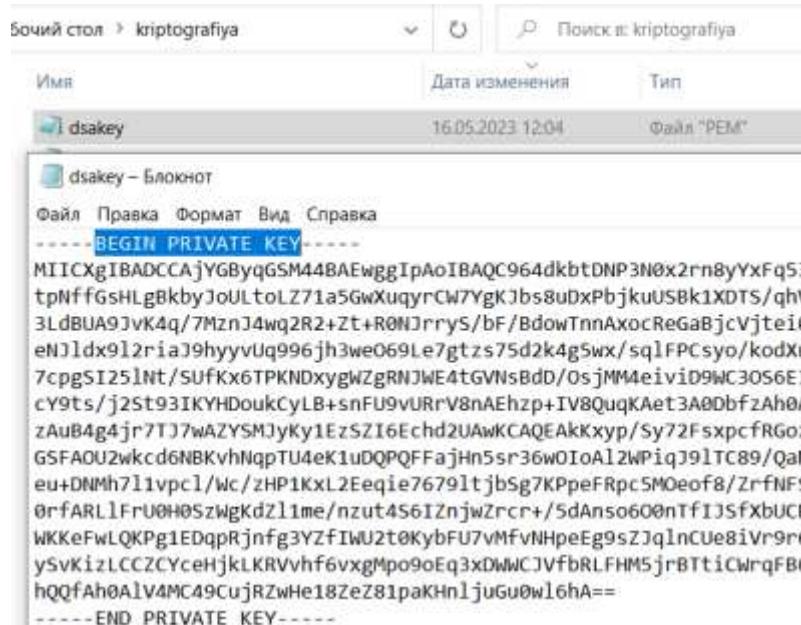
3) yaratilgan DSA parametrdan foydalanib, kalit hosil qilish(8-rasm).

<openssl genpkey -paramfile dsap.pem -out dsakey.pem> - dsap.pem nomli parameter fayldan foydalanib, kalit generatsiya qilish hamda hosil qilingan kalitni dsakey.pem faylga joylashtish.

```
C:\Users\HP\Desktop\kriptografiya>openssl genpkey -paramfile dsap.pem -out dsakey.pem  
C:\Users\HP\Desktop\kriptografiya>
```

8-rasm DSA parametrlarıň faydalanyb, kalit hasil ailiş

4) hosil qilingan kalitni ham kriptografiya nomli faylga kirib, topish mumkin(9-rasm).



9-rasm. DSA parametrdan foydalanib, maxfiy kalit hosil qilish.

DH algoritmi yordamida kalit generatsiyalash

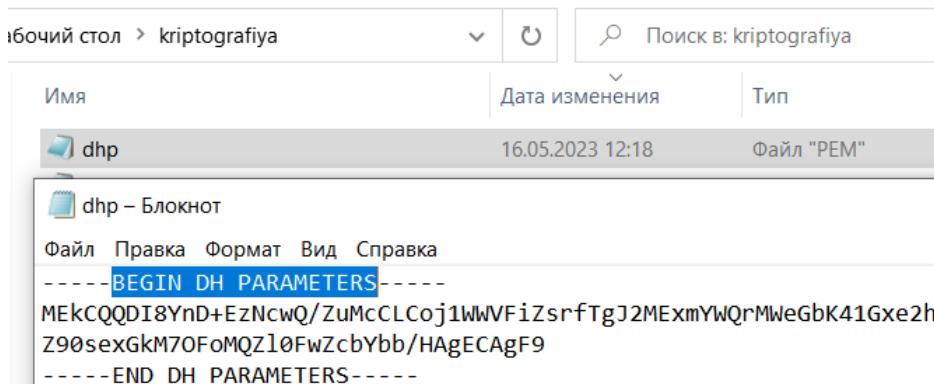
1) DH parametrini generatsiya qilish(10-rasm).

```
<openssl genpkey -genparam -algorithm DH -out dhp.pem -pkeyopt
```

dh_paramgen_prime_len:512> - DH algoritmi asosida 512 bitli dhp.pem DH parametrnini hosil qilish.

10-rasm. DH parametri

2) yaratilgan parametrni ochib ko'rish(11-rasm).



11-rasm. DH parametrining ko'rinishi.

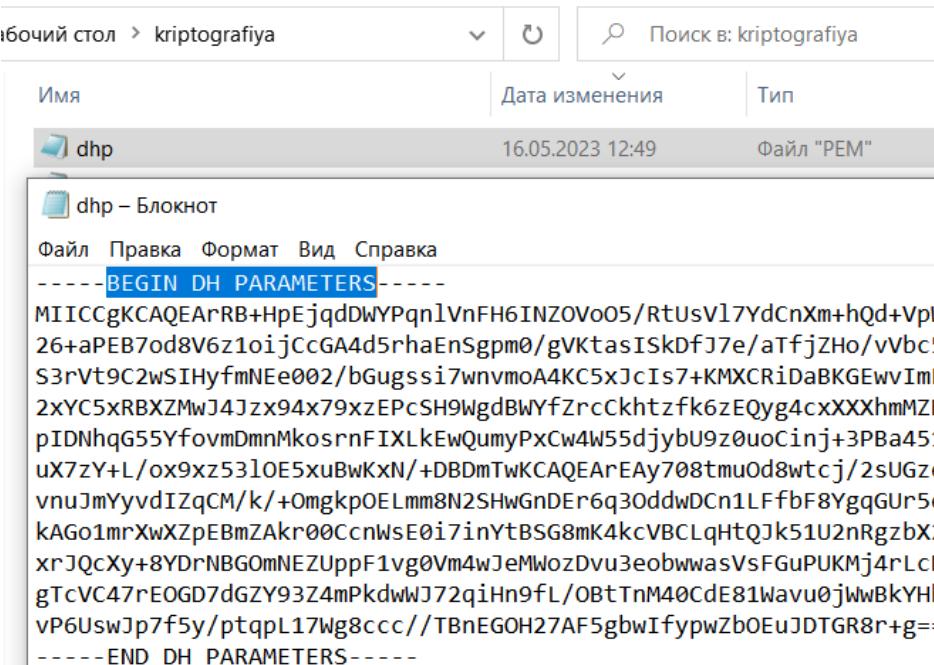
3) DH algoritmining RFC5114 standartidan foydalananib ham DH paramertini hosil qilish(12-rasm).

<openssl genpkey -genparam -algorithm DH -out dhp.pem -pkeyopt dh_rfc5114:2> - DH algoritmnining RFC5114 standartidan foydalangan holda 2048 bitli DH paramatrini hosil qilish.

```
C:\Users\HP\Desktop\kriptografiya>openssl genpkey -genparam -algorithm DH -out dhp.pem -pkeyopt dh_rfc5114:2
C:\Users\HP\Desktop\kriptografiya>
```

12-rasm. DH algoritmi yordamida RFC5114 standarti orqali DH parametr hosil qilish.

4) 17-rasmda hosil qilingan DH parametrni OpenSSL joylashgan parkaga kirib ko'rish mumkin(13-rasm).



13-rasm. DH parametrning ko'rinishi.

5) yaratilgan ixtiyoriy DH parametr dan foydalananib, maxfiy kalit hosil qilish(14-rasm).

<openssl genpkey -paramfile dhp.pem -out dhkey.pem> - dhp.pem DH parametni joylashgan fayldan foydalananib, dhkey.pem mavfiy kalit hosil qilish.

```
C:\Users\HP\Desktop\kriptografiya>openssl genpkey -paramfile dhp.pem -out dhkey.pem
C:\Users\HP\Desktop\kriptografiya>
```

14-rasm. DH parametrдан foydalanib, maxfiy kalit hosil qilish.

6) hosil qilingan ochiq kalitni ko'rish(15-rasm).

```
dhkey – Блокнот
Файл Правка Формат Вид Справка
-----BEGIN PRIVATE KEY-----
MIICQAIBADCCAhkGCSqGSIb3DQEATCCAgcOggEBAK0Qfh6RI6nQ1mD6p5VzxR+i
DWtlaDuf0bVLFZe2HQp15voUHflaVtuvmjxAe6HFFes9aIownBg0Hea4WhJ0oKZt
P4FSrWrCEpA3ye3v2k342R6P7IW3OUt61bfqtseib8n5jRhtNNv2xrolliu8J75q
AOGucSXCL0/iJFwky2gShhMLyJhdsWAucUQV2TMceCc8feMe/ccxD3Eh/VoHQV
mH2a3ApIbc350sxEMoOHMV114ZjGQaSAzYahueWH6L5g5pzJKLK5xSFy5BMELpsj
8QsOfueXY8m1Pc9LqAop4/tzwWu0db1+82P1/6Mfcc+d5Th0cbgcCsTf/gwQ5k8C
ggEBAKxAMu9PLZrjnfMLXI/9rFBs3r57i2mMr3SGagjpSP/jpoJKThC5pvDdkh8B
pwXK+qtznXcAwp9SX2xfGIkh1K+XpABqNZq18F2aRAZmQJk9NAJ1rBN1u4p2LQ
UhvJiuJHFQ0i6h7UCZOdVNp0YMF219sayuHF8vvGA6zQRjpjRGVKArdB4NFZuMCXj
FqMw77t3qG8MGrBRRj1CjI+Ky3CoE3FQu06xDhg+3RmWPd2eJj5HcFie9qoh5/
Xy/zgbu5zONAnRPNVmr7tI1sAZGB4bz+lLMCae3+cv6baqS9e1oPHHHP/0wZxBjh
9uwBeYG8CH8qcGwzhLiq@xkfk/oEHgIcZzqjq08u2mGFF2DSFv6zHDL7YVQLqkT0
TabFFg==
-----END PRIVATE KEY-----
```

15-rasm. Maxfiy kalit ko'rinishi.

Elliptik Egri chiziqqa asoslanib kalit generatsiyalash

1) elliptik egri chiziqqa asosida kalit generatsiyalash(16-rasm).

<openssl genpkey -algorithm EC -out eckey.pem -pkeyopt

ec_paramgen_curve:P-384 -pkeyopt ec_param_enc:named_curve> - Elliptik egri chiziqqa asoslangan P 384 maydonda kalit hosil qilish.

```
C:\Users\HP\Desktop\kriptografiya>openssl genpkey -algorithm EC -out eckey.pem -pkeyopt ec_param_enc:named_curve
C:\Users\HP\Desktop\kriptografiya>
```

16-rasm. Elliptik egri chiziqqa asosida kalit hosil qilish.

Yaratilgan faylni ochib ko'rish(17-rasm).

```
eckey – Блокнот
Файл Правка Формат Вид Справка
-----BEGIN PRIVATE KEY-----
MIG2AgEAMBAQByqGSM49AgEGBSuBBAAiBIGeMIGbAgEBBDCAEqt3fGLZDiLaa5Hz
KvyCoIQPeIEED3d0zLjpNmKvJVTx7jz2+8qvvpVebPCymc40hZANiAASMKLyu5vZR
AwKcnO+WDY4s72SVjF4BRiWYlsCLqn9iafy6vu6c6rUZZRzX0WaOPnERHCaUqFZp
+X6t80dqQSuL3BgtkwFsVoVVgSLtdT2ftmj0KnJ3Z4JXYZZinwJ3XGI=
-----END PRIVATE KEY-----
```

17-rasm. Elliptik egri chiziqqa asosan OpenSSL da kalit ko'rinishi.

Xulosa

OpenSSL kutubxonasi bir qancha kriptografik algoritmlar asosida maxfiy kalit hamda maxfiy kalit oraliq ochiq (hammagaga ma'lum oshkor kalit)

generatsiyasi uchun foydalaniladigan dasturiy vosita bo'lib, undagi buyruqlar orqali har xil kriptografik algoritmlardan foydalanish mumkin. Masalan ushbu maqolada kalit generatsiyasi uchun foydalanilgan algoritmlar RSA, DSA algoritmi va DH parametri orqali kalit generatsiyasi, shuningdek, elliptik egri chiziqqa asoslangan kalit generatsiyalash jarayoni kirishi mumkin.

FOYDALANILGAN ADABIYOTLAR:

22. KRIPTOGRAFIYANING MATEMATIK ASOSLARI O'quv qo'llanma D.Y.Akbarov, P.F.Xasanov, X.P.Xasanov, O.P.Axmedova, I.U.Xolimtayeva Toshkent 2018.
23. Detection of Intrusions and Malware, and Vulnerability Assessment 16th International Conference, DIMVA 2019, Gothenburg, Sweden, June 19–20, 2019, Proceedings
24. Blowfish Cipher Tutorials - Herong's Tutorial Examples, [Dr. Herong Yang](#) · 2020
25. Aldaya, A.C, Pereida Garcia, C., Alvarez Tapia, L.M., Brumley, B.B.: Cache-timing attacks on RSA key generation. IACR Cryptology ePrint Archive 2018(367)(2018). <https://eprint.iacr.org/2018/367>
26. Koblitz, Neal. Elliptic curve cryptosystems. Mathematics of Computation 48 (1987), 203-209. [One of the original articles that proposed the use of elliptic curves for cryptography. The other is by Victor Miller.]
27. <https://lib.fbtuit.uz/assets/files/4.-AkbarovD.YXasanovP.FXasanovX.PAxmedovaO.PXolimtayeval.U.Kriptografiyaningmatematikasoslari.pdf>
28. <https://www.ssldragon.com/blog/what-is-openssl/#:~:text=OpenSSL%20is%20an%20all%2Daround,generation%2C%20and%20SSL%20certificate%20installation>
29. Internet sayti
<https://people.math.rochester.edu/faculty/doug/otherpapers/Husemoller.pdf>
30. Internet sayti:
<https://www.openssl.org/docs/man1.0.2/man1/openssl-genpkey.html>