# Kriptografija

III (11) gimnazijos klasė

## Užduotys

Mokytojas gali pasirinkti, kurias užduotis mokiniai atliks individualiai, o kurias – dirbdami grupėse, taip pat numatyti, kada bus atliekamos pasirinktos užduotys – pamokoje, mokantis savarankiškai, projektinėse veiklose ir kt.

### 1 užduotis (9 skaidrė):

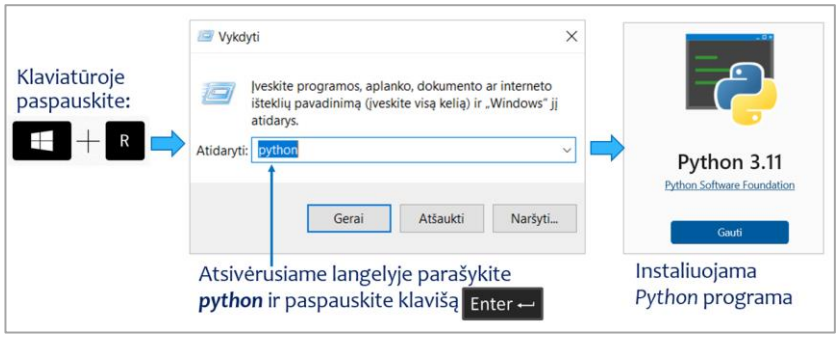
- ✓ Simetrinio šifravimo atveju sauga ypatingai priklauso nuo šifravimo rakto (metodo), šio rakto saugojimo ir jo saugaus perdavimo komunikacijos dalyviui. Pakomentuokite simetrinio šifravimo galimas saugos spragas ir veiksmus, didinančius šifravimo saugą.

### 2 užduotis (16–20 skaidrės):

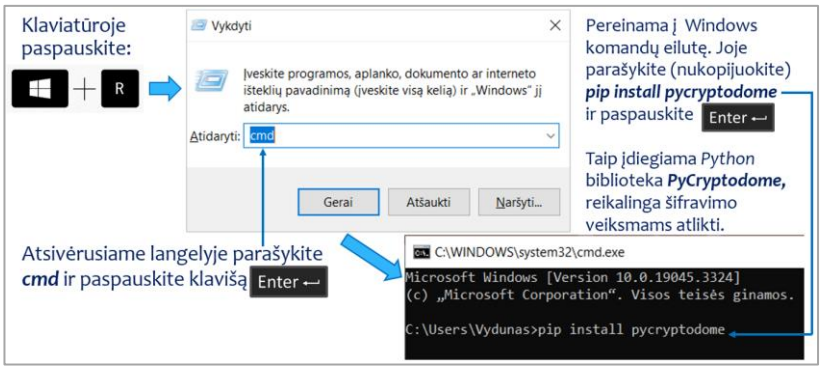
Programavimo kalba Python sudarytos programos, naudojančios simetrinio *Advanced Encryption Standard (AES)* šifravimo biblioteką, taikymas **tekstui užšifruoti**.

**1 žingsnis – PASIRUOŠIMAS** (16–17 skaidrės):

- ✓ Instaliuokite kompiuterio programą *Python*:




- ✓ Komandų eilutėje nurodydami **pip install pycryptodome**, įdiekite ***Python*** biblioteką ***PyCryptodome***:




**2 žingsnis – SIMETRINIO UŽŠIFRAVIMO ĮRANKIO IŠBANDYMAS** (18–20 skaidrės):

- ✓ Nusikopijuokite čia pateiktą programos kodą į programą „Užrašinę" ir įrašykite vardu **SIFR_AESR.PY** į **C:** disko aplanką **py**.

```
# PROGRAMOS SIFR_AESR.PY KODAS
from Crypto.Cipher import AES
def main():
    # Naudokime pastovų 32 baitų ilgio raktą
    key = b"ThisIsASecretKey_32bytesLongAbCd"

    # Inicijuoti AES šifro objektą su EAX režimu
    cipher = AES.new(key, AES.MODE_EAX)

    # Teksto užšifravimas
    plaintext = b"Slapta: Kompiuteris yra tavo ir mano geriausias draugas :-)."
    ciphertext, tag = cipher.encrypt_and_digest(plaintext)

    # Atspausdiname raktą, užšifruotą tekstą, tag ir nonce reikšmes
    print(f"Raktas: {key.decode('utf-8')}")
    print(f"Užšifruotas tekstas: {ciphertext.hex()}")
    print(f"Tag: {tag.hex()}")
    print(f"Nonce: {cipher.nonce.hex()}")  # Atspausdiname nonce reikšmę
    # Teksto iššifravimas
    cipher_dec = AES.new(key, AES.MODE_EAX, nonce=cipher.nonce)
    decrypted = cipher_dec.decrypt(ciphertext)

    # Patikriname, ar iššifruotas tekstas sutampa su pradiniais duomenimis
    try:
        cipher_dec.verify(tag)
        print(f"Pradinis: {decrypted.decode('utf-8')}")
    except ValueError:
        print("Iššifruotasis tekstas yra neteisingas!")
if __name__ == '__main__':
    main()
```

- ✓ Įvykdykite programą, komandų eilutėje nurodydami[1] **python c:\py\SIFR_AESR.PY**
- ✓ Panagrinėkite programos darbo rezultatą – informaciją, pateiktą įvykdžius programą, su bendramoksliais ir mokytoju aptarkite pateiktų dalių **Raktas, Užšifruotas tekstas, Tag, Nonce** paskirtį:

```
Raktas: ThisIsASecretKey_32bytesLongAbCd
Užšifruotas tekstas: e7883d8abd0243e2636be5c52899b1d3c749ba080807a22c959394750e24d72d35547cd15e570627517c700e1148f590208f79f7fafee7fbe7076073
Tag: 733d3cfd1421fdc76095bc8f65175d73
Nonce: b05e6383dee9e5e6406f112795d7276c
Pradinis: Slapta: Kompiuteris yra tavo ir mano geriausias draugas :-).
```

- ✓ Užšifruokite kitą tekstą – įrašykite norimą užšifruoti tekstą[2] atitinkamoje pateiktos programos vietoje ir įvykdykite programą:

**Pateiktame pavyzdyje užšifruojamas patarlės tekstas „Kas nedirba, mielas vaike, tam ir duonos duot nereikia.“**

```
# PROGRAMOS SIFR_AESR.PY KODAS
from Crypto.Cipher import AES
def main():
    # Naudokime pastovų 32 baitų ilgio raktą
    key = b"ThisIsASecretKey_32bytesLongAbCd"

    # Inicijuoti AES šifro objektą su EAX režimu
    cipher = AES.new(key, AES.MODE_EAX)

    # Teksto užšifravimas
    plaintext = b"Kas nedirba, mielas vaike, tam ir duonos duot nereikia."
    ciphertext, tag = cipher.encrypt_and_digest(plaintext)

    # Atspausdiname raktą, užšifruotą tekstą, tag ir nonce reikšmes
    print(f"Raktas: {key.decode('utf-8')}")
    print(f"Užšifruotas tekstas: {ciphertext.hex()}")
    print(f"Tag: {tag.hex()}")
    print(f"Nonce: {cipher.nonce.hex()}")  # Atspausdiname nonce reikšmę
```

*Čia pateiktas programos teksto fragmentas*

- ✓ Nukopijuokite programos darbo rezultato informaciją į, pavyzdžiui, programą „Užrašinė“ ir įrašykite failą – šią informaciją panaudosite kitoje užduotyje iššifruodami užšifruotą pranešimą:

**Programos darbo rezultato informacija panaudosite kitoje užduotyje**

```
Raktas: ThisIsASecretKey_32bytesLongAbCd
Užšifruotas tekstas: dbb6fd975fafc6282d8a27dd9841cc26bf67929926e7648fa
0b50fdc88bff2641ab5e68b7c447f310d527b6ff2df1f5c4498c4be4ede13
Tag: 265df1dfa03b68cd368605e2660c763d
Nonce: 78db0191dcadb80bb318579ad3a535d3
Pradinis: Kas nedirba, mielas vaike, tam ir duonos duot nereikia.
```

[1] *Jei įrašysite programą kitų vardu ir (arba) kitame aplanke, komandų eilutėje rašykite savo informaciją – nurodykite savo aplanką ir savo failo vardą.*

[2] *Šioje užšifravimo kompiuterio programoje turi būti naudojamos tik ASCII kodo raidės (raidės be diakritinių ženklų).*

## 3 užduotis (21–23 skaidrės):

Programavimo kalba **Python** sudarytos programos, naudojančios simetrinio *Advanced Encryption Standard (AES)* šifravimo biblioteką, taikymas **tekstui iššifruoti**.

**1 žingsnis – UŽŠIFRAVIMO INFORMACIJOS ĮKĖLIMAS Į PROGRAMĄ** (21 skaidrė):

✓ Nusikopijuokite čia pateiktą programos kodą į programą „Užrašinė“:

```
# PROGRAMOS SIFR_AESS.PY KODAS
from Crypto.Cipher import AES
from binascii import unhexlify
def decrypt_aes_eax(ciphertext_hex, key, tag_hex, nonce_hex):
    # Konvertuojame šiuos duomenis iš šešioliktainės į baitų sekas
    ciphertext = unhexlify(ciphertext_hex)
    tag = unhexlify(tag_hex)
    nonce = unhexlify(nonce_hex)
    # Sukuriamas šifravimo objektas su pateiktu raktu ir nonce
    cipher = AES.new(key, AES.MODE_EAX, nonce=nonce)

    # Bandoma iššifruoti tekstą ir patikrinti MAC su pateiktu tag'u
    try:
        decrypted_data = cipher.decrypt_and_verify(ciphertext, tag)
        return decrypted_data.decode('utf-8')
    except ValueError:
        return "Užšifruotas tekstas buvo pažeistas. Iššifravimas nutrauktas."
def main():
    # Pateikti duomenys
    key = b"ThisIsASecretKey_32bytesLongAbCd"
    ciphertext_hex =
"dbb6fd975fafc6282d8a27dd9841cc26bf67929926e7648fa0b50fdc88bff2641ab5
e68b7c447f310d527b6ff2df1f5c4498c4be4ede13"
    tag_hex = "265df1dfa03b68cd368605e2660c763d"
    nonce_hex = "78db0191dcadb80bb318579ad3a535d3"
    decrypted_message = decrypt_aes_eax(ciphertext_hex, key, tag_hex,
nonce_hex)
    print(f"Iššifruotas tekstas: {decrypted_message}")
if __name__ == '__main__':
    main()
```

✓ Į programos teksto atitinkamas vietas įkelkite anksčiau išsaugotą užšifravimo informaciją*:
- Raktas → **key**
- Užšifruotas tekstas → **ciphertext_hex**
- Tag → **tag_hex**
- Nonce → **nonce_hex**

✓ Įrašykite programą vardu **SIFR_AESS.PY** į **C:** disko aplanką **py**.

*Pavyzdyje panaudota informacija, gauta užšifravus tekstą „***Kas nedirba, mielas vaike, tam ir duonos duot nereikia“***

**2 žingsnis – UŽŠIFRUOTO PRANEŠIMO IŠŠIFRAVIMAS, TYRINĖJIMAS** (22–23 skaidrės):

- ✓ Įvykdykite programą, komandų eilutėje nurodę[3]: **python c:\py\SIFR_AESS.PY**
- ✓ Programos darbo rezultatas – iššifruotas pranešimas:

```
Iššifruotas tekstas: Kas nedirba, mielas vaike, tam ir duonos duot nereikia.
```

- ✓ Ištirkite, koks bus iššifravimo programos darbo rezultatas, jei joje pateiksime šiek tiek pakeistą užšifruotą tekstą, pavyzdžiui, tekste bus pakeistas vienas simbolis: dbb6 → dbb8

```
# PROGRAMOS SIFR_AESS.PY KODAS
<...>
def main():
   # Pateikti duomenys
key = b"ThisIsASecretKey_32bytesLongAbCd"
   ciphertext_hex =
"dbb6fd975fafc6282d8a27dd9841cc26bf67929926e7648fa0b50fdc88bff2641ab5
e68b7c447f310d527b6ff2df1f5c4498c4be4ede13"
   tag_hex = "265df1dfa03b68cd368605e2660c763d"
   nonce_hex = "78db0191dcadb80bb318579ad3a535d3"
```

*Čia pateiktas programos teksto fragmentas*

- ✓ Išsiaiškinkite, kas vyks iššifruojant, jei pakeisite programos kitų dalių užšifravimo informaciją:
  - key
  - tag_hex
  - nonce_hex

## 4 užduotis (24 skaidrė):

- ✓ Su pateiktu simetrinio iššifravimo įrankiu – programa SIFR_AESS.PY, kurią naudojote 3-oje užduotyje – iššifruokite čia pateiktus užšifruotus pranešimus:

**Pirmas užšifruotas pranešimas:**

Raktas: ThisIsASecretKey_32bytesLongAbCd

Užšifruotas tekstas:
47b3ec03eacbe4aaddb22192c685f69da5c0fab86d5b7c0fb56e9c00f1035ef59eb1146cbd44138d9e9fa69e7b3b8957a65ff9a3464a3eebb5762b9172e64844cfe380807c2baf7c4b11a23af2e25969a52e7275e2e1591dfd3cd0392c18c7db1c6397b69800

Tag: 692a8142327e1fd3de00bd76a1833931

Nonce: 3150360edd204dbcb610883b168b5958

[3] *Jei įrašysite programą kitų vardu ir (arba) kitame aplanke, komandų eilutėje rašykite savo informaciją – nurodykite savo aplanką ir savo failo vardą.*

**Antras užšifruotas pranešimas:**

Raktas: ThisIsASecretKey_32bytesLongAbCd
Užšifruotas tekstas:
a8cfed33a6ca1fd8bc16f5f712544a9c6350f20b5597495d46b4222b81c0cd7c4ead46e9811bcd7040d814348123cbc4718df4e03403106d7dc846e009755cd2943ed6a044761c410e6e88c5f2
Tag: 1d7fce75962fa02764f15b57769d097a
Nonce: 6900c5899fa76800ba7bff77e0fe6cb6

## 5 užduotis (25–28 skaidrės):

Simetrinis šifravimas **naudojant šifravimo raktą**.

**1 žingsnis – SIMETRINIO ŠIFRAVIMO RAKTO SUKŪRIMAS** (25 skaidrė):

- ✓ Pakartotinai įdiekite/atnaujinkite kriptografijos biblioteką komandų (CMD) eilutėje nurodydami **pip install --upgrade cryptography**.
- ✓ Naudodami **Python SIM.PY** programą, aplanke **c:\py** sukurkime simetrinį šifravimo raktą:
  - nusikopijuokite čia pateiktą programos kodą į užrašinę ir įrašykite vardu **SIM.PY** į **C:** disko aplanką **py**;
  - įvykdykite programą, komandų eilutėje nurodydami komandas:
    **cd c:\py**
    **python SIM.PY**

```
# PROGRAMOS SIM.PY KODAS
#  sim.py generuoja simetrinį raktą key.bin
from cryptography.fernet import Fernet
# Sugeneruojame simetrinį raktą
key = Fernet.generate_key()
# Išsaugome raktą į key.bin failą
with open("key.bin", "wb") as key_file:
   key_file.write(key)
print(f"Raktas sugeneruotas ir išsaugotas į key.bin: {key}")
```

  - aplanke **py** bus sukurtas **simetrinio šifravimo raktas** – failas **key.bin**.

**2 žingsnis – INFORMACIJOS UŽŠIFRAVIMAS NAUDOJANT SIMETRINIO ŠIFRAVIMO RAKTĄ** (26 skaidrė):

- ✓ Naudodami **Python** programą **Simetrinis_e_d.py** sukurtu simetrinio šifravimo raktu **key.bin** užšifruokite posakį: **If at first you don't succeed, call it version 1.0.**

*(Jeigu iš pirmo karto nepasisekė, pavadink tai versija 1.0. – Programuotojų išmintis)*

  - Šį posakį įkelkite į programą „Užrašinė“ ir įrašykite į aplanką **c:\py** vardu, pavyzdžiui, **posakis.txt**.
  - Į tą patį aplanką įrašykite ir simetrinio šifravimo/iššifravimo programą **Simetrinis_e_d.py**:

```
# python Simetrinis_e_d.py  e|d  key.bin Tekstas.txt Tekstas.bin
from cryptography.fernet import Fernet
import sys
def encrypt(key, plaintext):
    cipher_suite = Fernet(key)
    return cipher_suite.encrypt(plaintext)
def decrypt(key, ciphertext):
    cipher_suite = Fernet(key)
    return cipher_suite.decrypt(ciphertext)
def main():
    action = sys.argv[1]
    key_file = sys.argv[2]
    text_file = sys.argv[3]
    output_file = sys.argv[4]

    with open(key_file, 'rb') as f:
        key = f.read()

    with open(text_file, 'rb') as f:
        text = f.read()
    if action == 'e':
        encrypted_text = encrypt(key, text)
        with open(output_file, 'wb') as f:
            f.write(encrypted_text)
    elif action == 'd':
        decrypted_text = decrypt(key, text)
        with open(output_file, 'wb') as f:
            f.write(decrypted_text)
    else:
        print("Invalid command. Use 'e' for encryption or 'd' for
decryption.")

if __name__ == '__main__':
    main()
```

- Komandų eilutėje nurodykite[4]:

  **cd c:\py**
  **python simetrinis_e_d.py  e  key.bin  posakis.txt  posakis.bin**

## 3 žingsnis – INFORMACIJOS IŠŠIFRAVIMAS NAUDOJANT SIMETRINIO ŠIFRAVIMO RAKTĄ (27 skaidrė):

✓ Iššifruokite užšifruotą tekstą:

- Naudodami programą „Užrašinė“ atverkite failą **posakis.bin**. Kaip galvojate, ar įmanoma be rakto jį iššifruoti?
- Ištrinkite arba pervardykite aplanke **c:\py** pradinio teksto failą **posakis.txt**. Pabandykite jį atstatyti iš užšifruoto failo vykdydami komandas komandų (CMD) eilutėje:

  **python simetrinis_e_d.py  d  key.bin  posakis.bin  posakis.txt**

- Perskaitykite iššifruotą tekstą, kurį programa įrašė į failą **posakis.txt**. Ar pastebėjote kokius nors neatikimus ar iškraipymus palyginus su pradiniu tekstu?

[4] *Jei įrašysite programą ir šifravimo raktą kitų vardu ir (arba) kitame aplanke, komandų eilutėje rašykite savo informaciją – nurodykite savo aplanką ir savo failų vardus.*

## 6 užduotis (28 skaidrė):

Pranešimo iššifravimas **naudojant simetrinį šifravimo raktą**.

- ✓ Naudodami programą **Simetrinis_e_d.py**, pabandykite iššifruoti čia pateiktą „labai slaptą“ užšifruotą tekstą:

```
gAAAAABlFHw0MeVcPQwYOn2CIn9Uq41Kr2Gr_ND5e9_YKm2lVFqkL8MktROpvZdvJV9LoesjVWG6BTrUdN1EpS
SBAIXQRgzOat4TMI5RcIk8PjCoxXREly0siOq8rGcvRLJqHJjH_WDYsFuTfOcWpSOY8zMqmlLZK7MzHXZwH8v_YQk
EVstb5zuQ83OuhEBl9ob8yqNfHnP0dgEyYsjiHyxg_o3R116LLg==
```

  - Pateiktą užšifruotą tekstą įkelkite į programą „Užrašinė“ ir įrašykite į **c:\py** vardu **slaptas_tekstas.bin**.
  - Į programą „Užrašinė“ įkelkite šį simetrinį raktą: **ucE_arr6D5lyPcjNMU_CuCUwl1PsfwBW7uEDBgVx2yU=** ir įrašykite failą į **c:\py** vardu **key.bin**.
  - Iššifruokite šį tekstą naudodami programą **Simetrinis_e_d.py**, ir simetrinį raktą **key.bin**, komandų (CMD) eilutėje nurodydami:

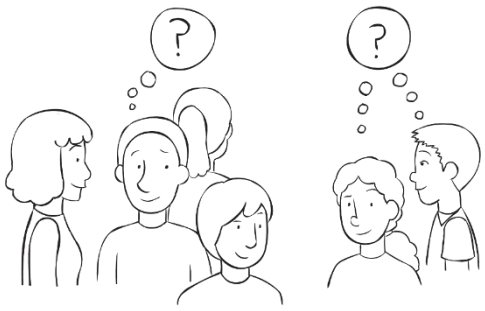
    **python simetrinis_e_d.py d key.bin slaptas_tekstas.bin slaptas_tekstas.txt**

  - Perskaitykite iššifruotą tekstą. Ar tikrai jį buvo galima laikyti labai slaptu? ☺

## 7 užduotis (29–30 skaidrės):

Pranešimo užšifravimas ir iššifravimas **naudojant savo sukurtą šifravimo raktą** (užduotis darbui grupėse po 2–3 mokinius).

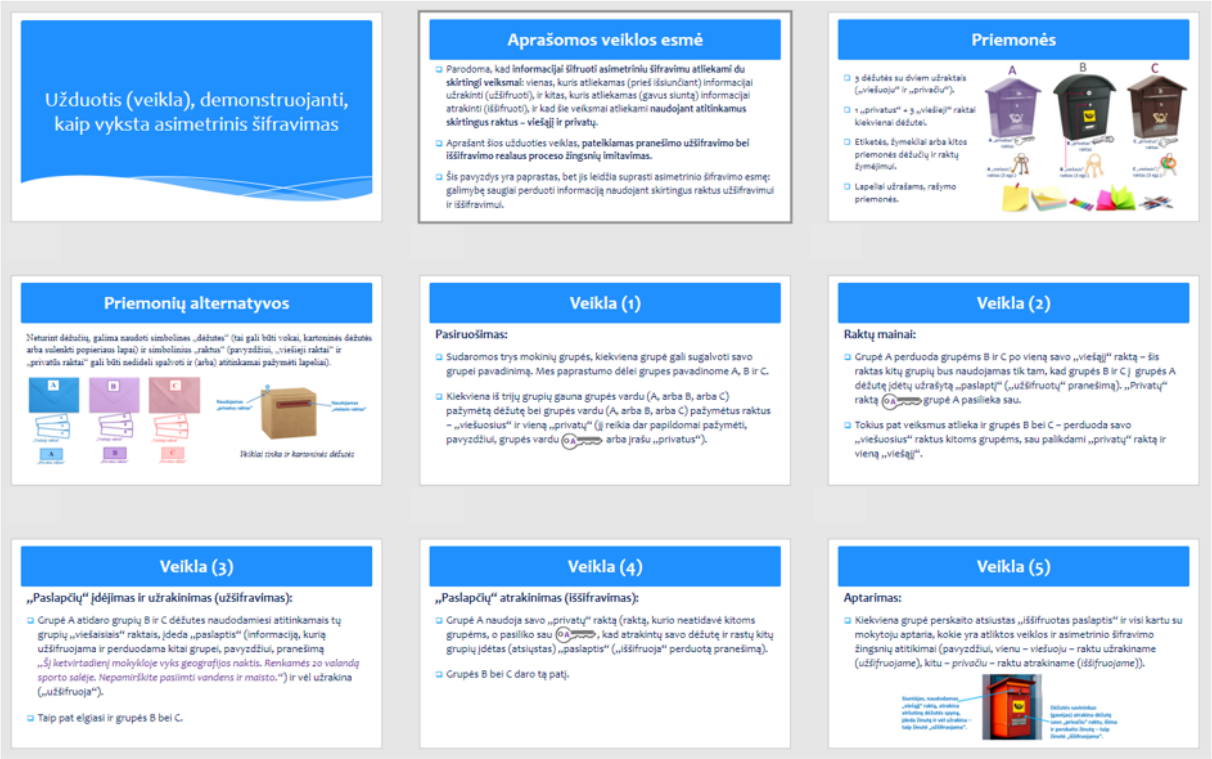
- ✓ Mokytojui padedant pasidalinkite į mažas grupes po 2–3 mokinius.
- ✓ Kiekvienas grupės narys:
  - naudodamas **Python** programą **SIM.PY**, sukurkite savo šifravimo raktą **key.bin**;
  - programa **Simetrinis_e_d.py** su savo sukurtu šifravimo raktu **key.bin** užšifruokite norimą informaciją ir perduokite kitam grupės nariui (būtina perduoti ir šifravimo raktą – pagalvokite, kaip tai atlikti laikantis saugumo reikalavimų);
  - iššifruokite iš grupės draugo gautą informaciją (naudokite programą **Simetrinis_e_d.py** ir grupės draugo perduotą raktą **key.bin**).
- ✓ Grupelėse, po to ir visi kartu su mokytoju aptarkite:
  - kaip pavyko atlikti užduotį;
  - ar pakankamai saugios buvo jūsų pasirinktos šifravimo rakto perdavimo strategijos;
  - ką galima būtų patobulinti siekiant didesnio užšifruotos informacijos saugumo.



https://www.pngkey.com/detail/u2t4t4y3q8y3u2o0_appointing-a-secret-person-line-art/

## 8 užduotis (38 skaidrė):

- ✓ Prisiminkite (arba atlikite) ir aptarkite užduotį, demonstruojančią, kaip vyksta keitimasis informacija naudojant asimetrinio šifravimo metodą (užduotis aprašyta 9 – 10 klasės metodinėje medžiagoje – pateiktyje ir užduočių faile):




*Užduotis su išsamiu aprašymu pateikta metodinės medžiagos 9–10 klasių pateikties 29–37 skaidrėse.*

## 9 užduotis (52–53 skaidrės):

Simetrinio ir asimetrinio **šifravimo derinimo pavyzdys**.

- ✓ Agnė ir Benas dažnai susirašinėja rengdami bendrą projektą.
- ✓ Išnagrinėkite Agnės ir Beno, kurie keičiasi informacija, susitarimus ir veiksmus:
  - Agnė reikalauja, kad Benas el. paštu siųstų jai tik užšifruotas žinutes.
  - Ji sukuria privataus/viešojo rakto porą, saugo savo privatų raktą paslaptyje ir paskelbia savo viešąjį raktą.
  - Benas turi žinutę, kurią nori nusiųsti Agnei.
  - Jis sukuria naują simetrinį raktą ir šį raktą panaudoja, kad užšifruotų savo pranešimą Agnei.
  - Po to Benas naudoja Agnės viešąjį raktą, kad užšifruotų savo simetrinį raktą.
  - Benas siunčia užšifruotą pranešimą ir užšifruotą simetrinį raktą Agnei (toks raktas vadinamas ***suvyniotu*** arba ***apgaubtu raktu***).
  - Agnė naudoja savo privatų raktą (iš privačios/viešos poros), kad iššifruotų simetrinį Beno raktą.
  - Po to ji naudoja simetrinį Beno raktą, kad iššifruotų pranešimą.

- ✓ Kaip manote, ar pakankamai saugiai buvo perduotas Agnei Beno sukurtas simetrinis šifravimo raktas?
- ✓ Aptarkite tai bendroje grupės diskusijoje.

*(adaptuota pagal [https://learn.microsoft.com/](https://learn.microsoft.com/) )*

---

**Medžiagą parengė**

**Tatjana Balvočienė,** informatikos mokytoja ekspertė, Šilutės Vydūno gimnazija

**Antanas Balvočius,** Kompetencijų aprašo, Bendrųjų programų (BP) įvado ir Informatikos BP bei rekomendacijų bendraautorius

2023 m. rugsėjis