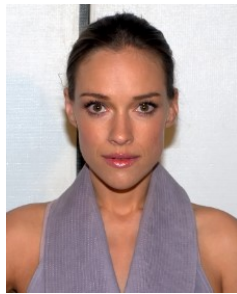


A large green shape on the left side of the slide, resembling a stylized letter 'C' or a bracket. It has a white semi-circular cutout on its right side.

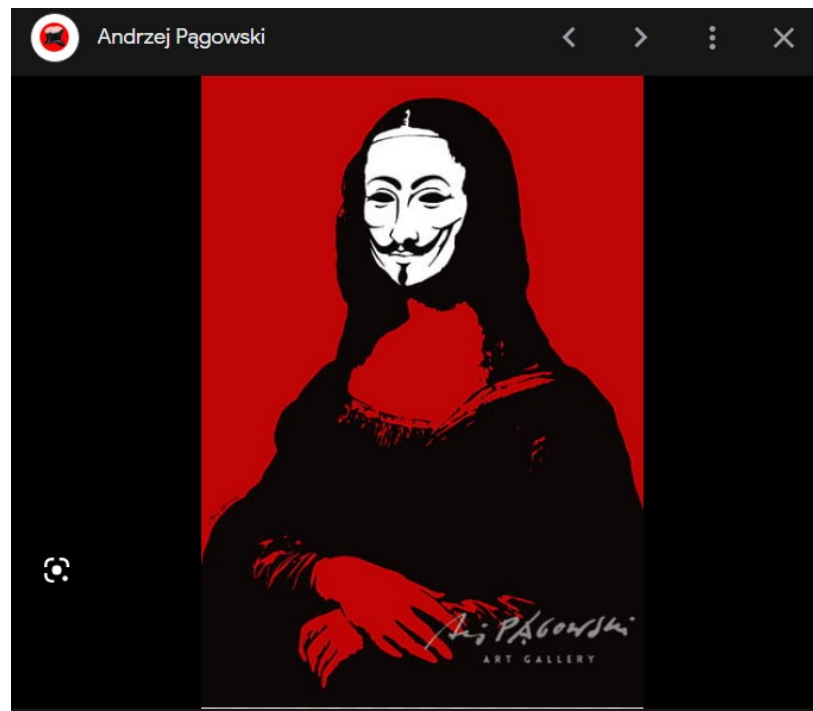
Szyfrowanie



Alicja, Bolek i Ewa

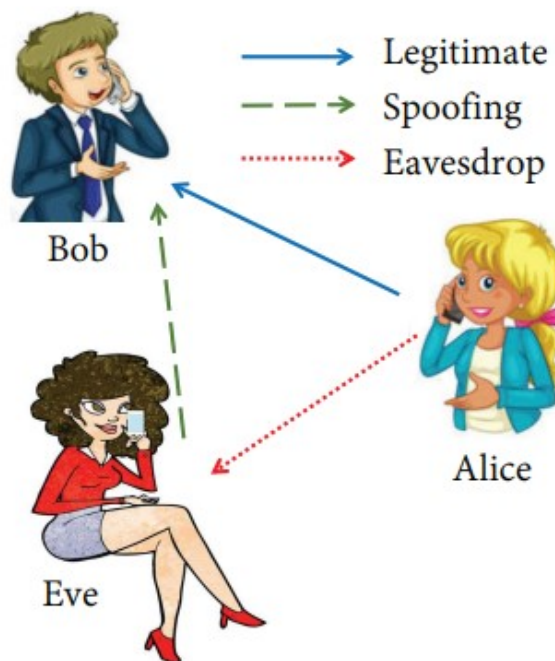


Alicja, Bolek i Ewa

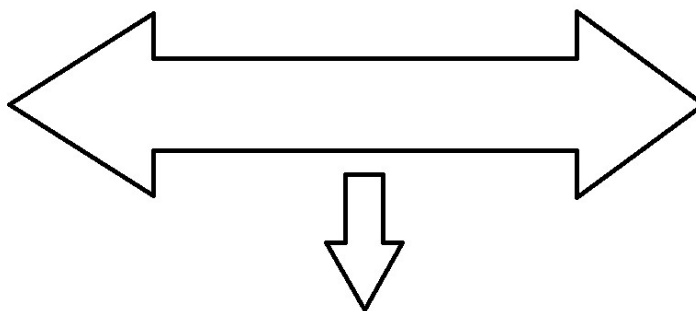
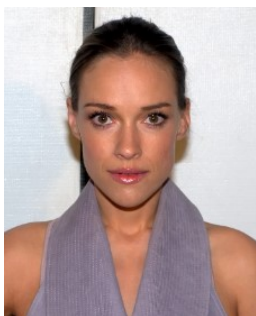


Alicja, Bolek i Ewa

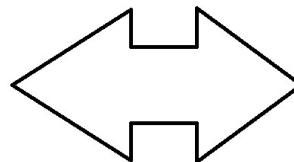
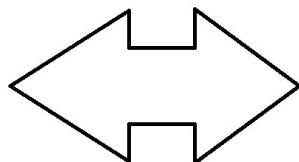
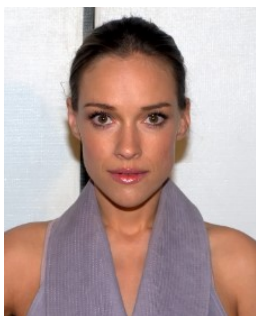
<https://doi.org/10.3390/s21165379>



Zagrożenia - Atak pasywny



Zagrożenia - Atak aktywny



Man in the middle
(MITM)

zastosowania

1. Poufność danych przechowywanych
2. Poufność danych przesyłanych
3. Generowanie ciągów pseudolosowych

Rys historyczny

Ok. I wiek p.n.e

Szyfr Cezara (zwany jest też **szyfrem przesuwającym**, **kodek Cezara** lub **przesunięciem Cezariańskim**) – jedna z najprostszych technik szyfrowania. Jest to rodzaj szyfru podstawieniowego, w którym każda litera tekstu jawnego (niezaszyfrowanego) zastępowana jest inną, oddaloną od niej o stałą liczbę pozycji w alfabecie, literą (szyfr monoalfabetyczny), przy czym kierunek zamiany musi być zachowany. Nie różni się przy tym liter dużych i małych. Nazwa szyfru pochodzi od Juliusza Cezara, który prawdopodobnie używał tej techniki do komunikacji ze swymi przyjaciółmi [Wiki].

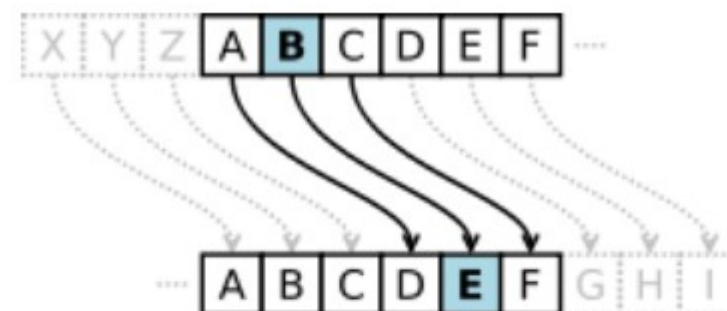
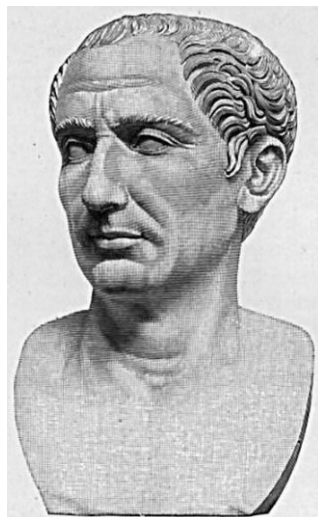
Enigma (z gr. αινιγμα „zagadka”) – niemiecka przenośna elektromechaniczna maszyna szyfrująca [Wiki]

W 1976 roku amerykańscy kryptografowie Whitfield Diffie i Martin Hellman opublikowali pracę "Nowe kierunki w kryptografii", w której przedstawili koncepcję **wymiany kluczy publicznych**.

Rys historyczny

Ok. I wiek p.n.e

Szyfr Cezara



Szyfr Cezara zastępuje każdą literę tekstu jawnego inną, przesuniętą względem litery kodowanej o stałą liczbę pozycji w alfabecie. Na rysunku szyfr z przesunięciem równym 3, tak więc **B** w tekście jawnym jest podmieniane w szyfrogramie na **E** (rozpatrywany jest alfabet łaciński).

Rys historyczny

Enigma (z gr. αἰνigma „zagadka”) – niemiecka przenośna elektromechaniczna maszyna szyfrująca [Wiki]



- 1932 r. Polacy opracowali efektywne metody deszyfrowania ówczesnej wersji Enigmy.
- Przed wybuchem wojny „wzmocniono” Enigme
- Anglicy złamali „wzmocnioną” Enigmę wykorzystując wiedzę przekazaną przez Polaków.

Rys historyczny

W 1976 roku amerykańscy kryptografowie Whitfield Diffie i Martin Hellman opublikowali pracę "Nowe kierunki w kryptografii", w której przedstawili koncepcję **wymiany kluczy publicznych**.

644

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-22, NO. 6, NOVEMBER 1976

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

Abstract—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a com-

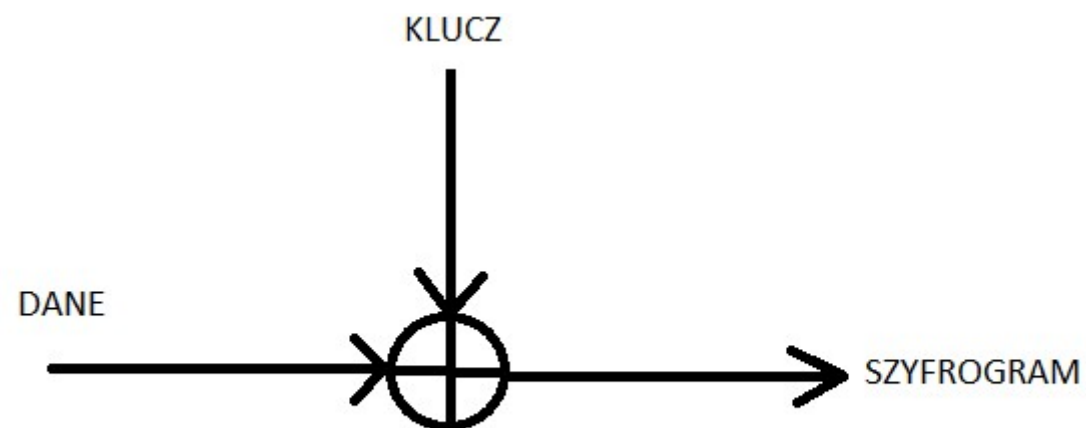
Rys historyczny

W 1976 roku amerykańscy kryptografowie Whitfield Diffie i Martin Hellman opublikowali pracę "Nowe kierunki w kryptografii", w której przedstawili koncepcję **wymiany kluczy publicznych**.

Kryptografia asymetryczna została oficjalnie wynaleziona przez cywilnych badaczy Martina Hellmana, Whitfielda Diffie w 1976 roku. Prawie równolegle prototyp podobnego systemu stworzył Ralph Merkle – w 1974 roku zaproponował algorytm wymiany kluczy nazwany puzzleami Merkle'a[1]. Dopiero pod koniec XX wieku brytyjska służba wywiadu elektronicznego GCHQ ujawniła, że pierwsza koncepcja systemu szyfrowania z kluczem publicznym została opracowana przez jej pracownika Jamesa Ellisa już w 1965 roku, a działający system stworzył w 1973 roku Clifford Cocks, również pracownik GCHQ[2]. Odkrycia te były jednak objęte klauzulą tajności do 1997 roku.

Szyfrowanie

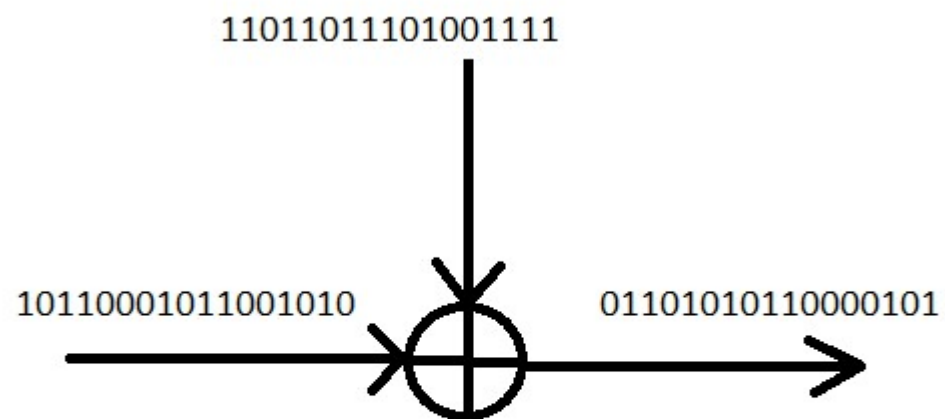
XOR



Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Szyfrowanie

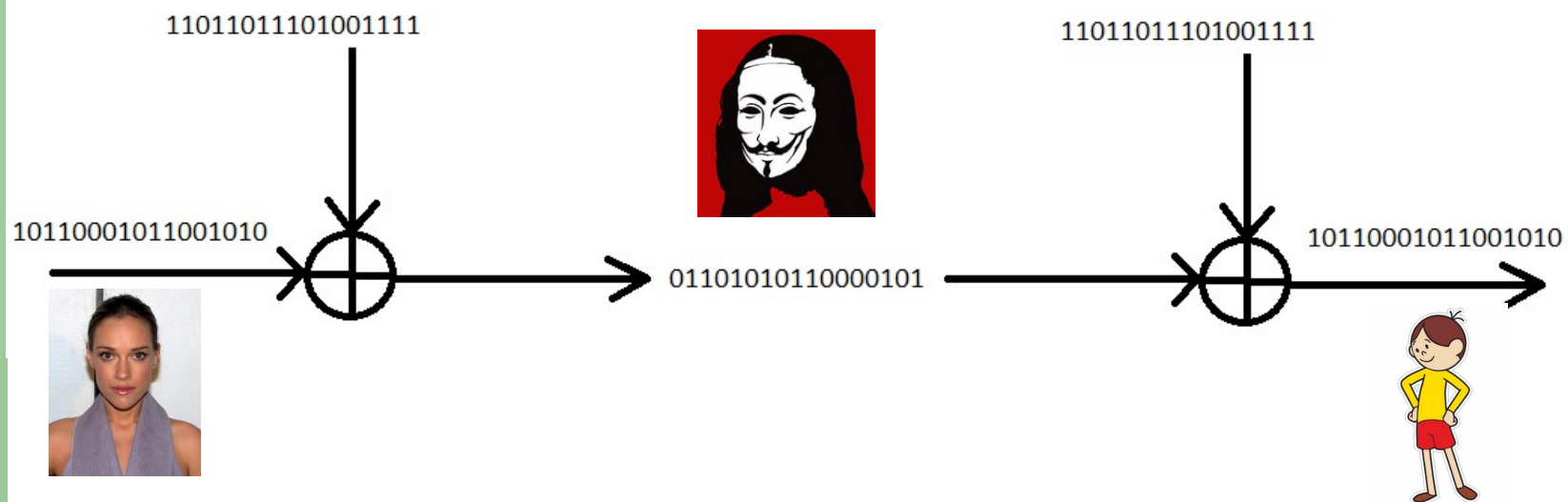
XOR



$$\begin{array}{r} \oplus \quad 11011011101001111 \\ \quad 10110001011001010 \\ \hline \quad 01101010110000101 \end{array}$$

Szyfrowanie

XOR



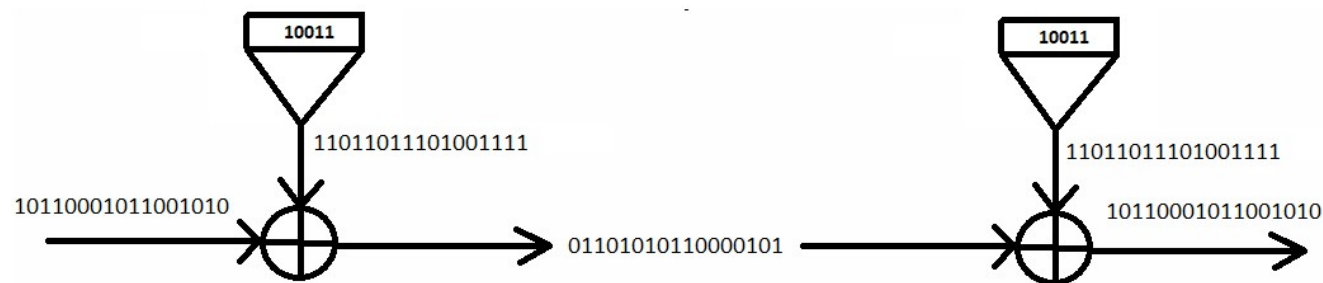
Szyfr Vernama – szyfr idealny

Tryby pracy szyfrów

- strumieniowy
- książka elektroniczna
- licznikowy
- łańcuchowy

Tryby pracy szyfrów

- strumieniowy

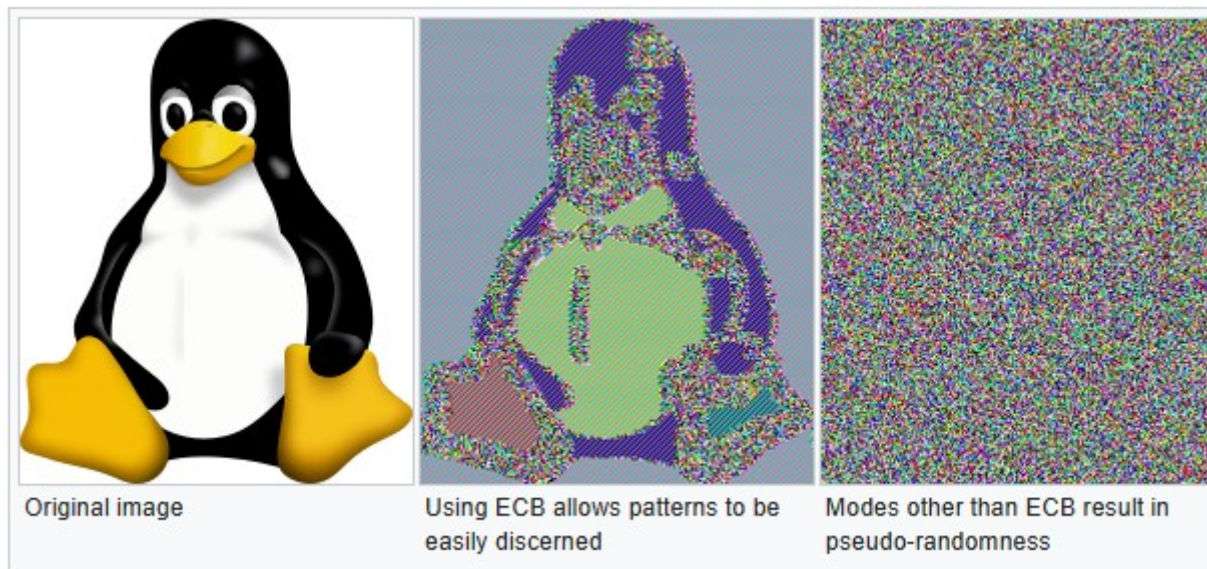


Openssl enc -list

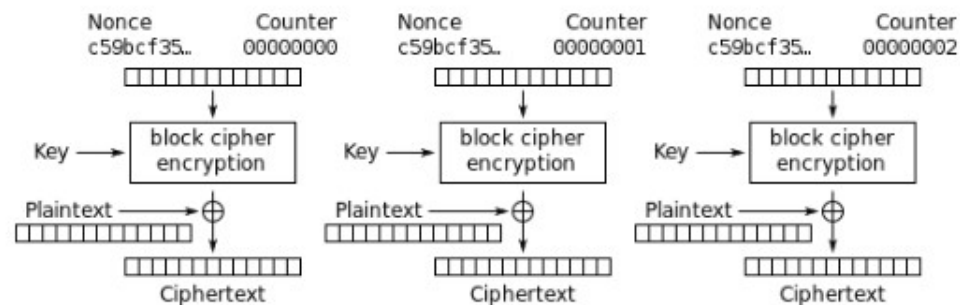
-aes-128-cbc	-aes-128-cfb	-aes-128-cfb1
-aes-128-cfb8	-aes-128-ctr	-aes-128-ecb
-aes-128-ofb	-aes-192-cbc	-aes-192-cfb
-aes-192-cfb1	-aes-192-cfb8	-aes-192-ctr
-aes-192-ecb	-aes-192-ofb	-aes-256-cbc
-aes-256-cfb	-aes-256-cfb1	-aes-256-cfb8
-aes-256-ctr	-aes-256-ecb	-aes-256-ofb
-aes128	-aes128-wrap	-aes192
-aes192-wrap	-aes256	-aes256-wrap
-aria-128-cbc	-aria-128-cfb	-aria-128-cfb1
-aria-128-cfb8	-aria-128-ctr	-aria-128-ecb
-aria-128-ofb	-aria-192-cbc	-aria-192-cfb
-aria-192-cfb1	-aria-192-cfb8	-aria-192-ctr
-aria-192-ecb	-aria-192-ofb	-aria-256-cbc
-aria-256-cfb	-aria-256-cfb1	-aria-256-cfb8
-aria-256-ctr	-aria-256-ecb	-aria-256-ofb
-aria128	-aria192	-aria256
-bf	-bf-cbc	-bf-cfb
-bf-ecb	-bf-ofb	-blowfish
-camellia-128-cbc	-camellia-128-cfb	-camellia-128-cfb1
-camellia-128-cfb8	-camellia-128-ctr	-camellia-128-ecb
-camellia-128-ofb	-camellia-192-cbc	-camellia-192-cfb
-camellia-192-cfb1	-camellia-192-cfb8	-camellia-192-ctr
-camellia-192-ecb	-camellia-192-ofb	-camellia-256-cbc
-camellia-256-cfb	-camellia-256-cfb1	-camellia-256-cfb8
-camellia-256-ctr	-camellia-256-ecb	-camellia-256-ofb
-camellia128	-camellia192	-camellia256
-cast	-cast-cbc	-cast5-cbc
-cast5-cfb	-cast5-ecb	-cast5-ofb
-chacha20	-des	-des-cbc
-des-cfb	-des-cfb1	-des-cfb8
-des-ecb	-des-edc	-des-edc-cbc
-des-edc-cfb	-des-edc-ecb	-des-edc-ofb
-des-edc3	-des-edc3-cbc	-des-edc3-cfb
-des-edc3-cfb1	-des-edc3-cfb8	-des-edc3-ecb
-des-edc3-ofb	-des-ofb	-des3
-des3-wrap	-desx	-desx-cbc

Tryby szyfrowania blokowego

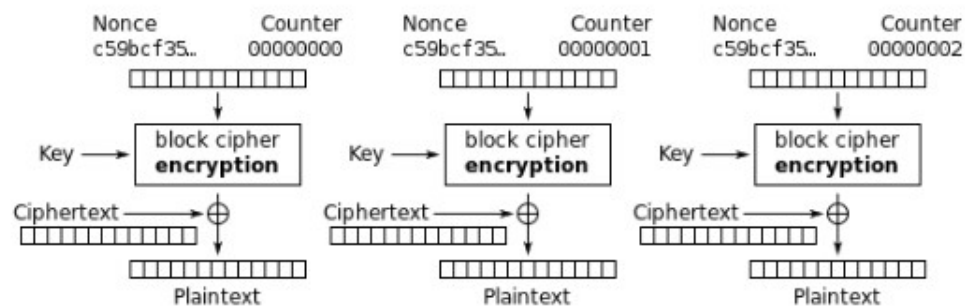
https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation



Tryby szyfrowania blokowego /strumieniowego

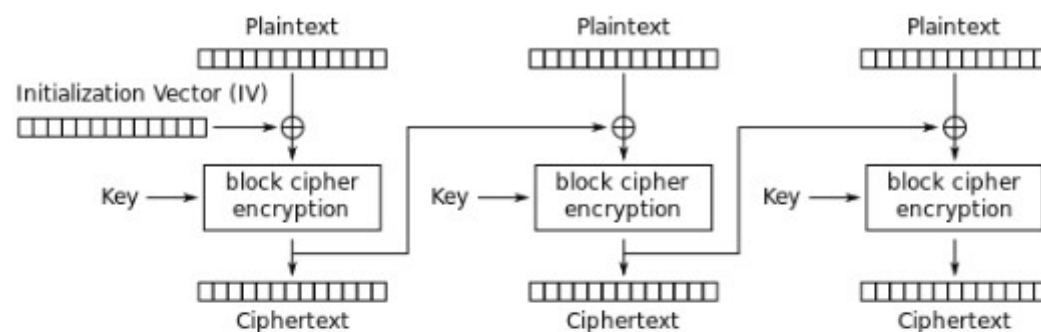


Counter (CTR) mode encryption

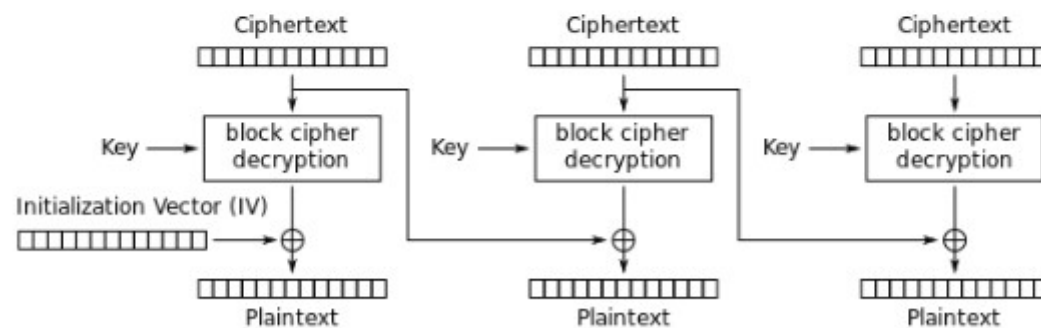


Counter (CTR) mode decryption

Tryby łańcuchowy

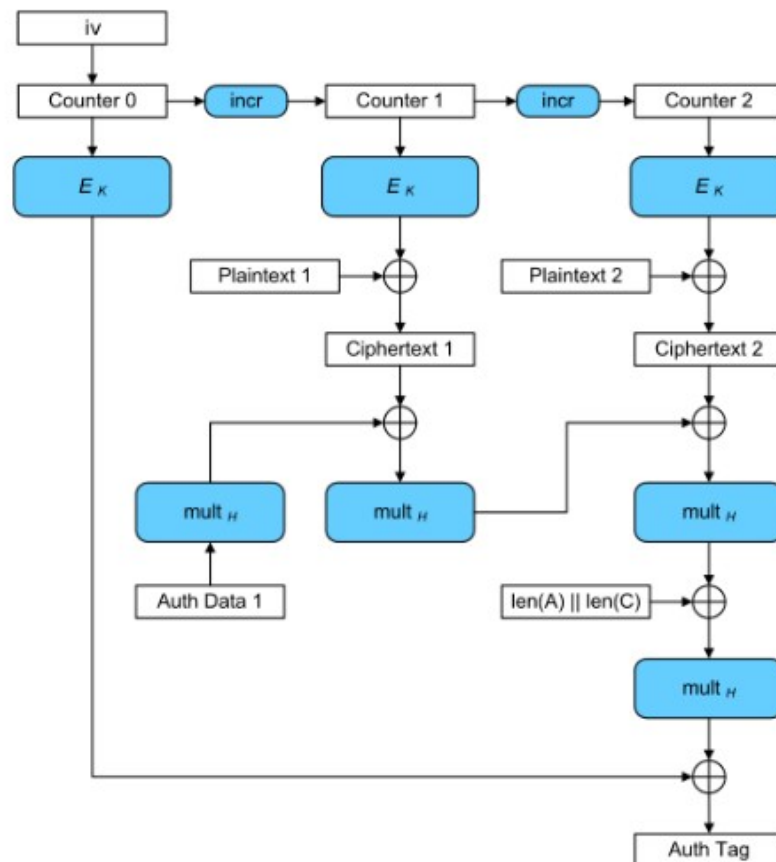


Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Tryby szyfrowania z uwierzytelnieniem



openssl enc -help

Usage: enc [options]

Valid options are:

- help Display this summary
- list List ciphers
- ciphers Alias for -list
- in infile Input file
- out outfile Output file
- pass val Passphrase source
- e Encrypt
- d Decrypt
- p Print the iv/key
- P Print the iv/key and exit
- v Verbose output
- nopad Disable standard block padding
- salt Use salt in the KDF (default)
- nosalt Do not use salt in the KDF
- debug Print debug info
- a Base64 encode/decode, depending on encryption flag
- base64 Same as option -a
- A Used with -[base64]a] to specify base64 buffer as a single line
- bufsize val Buffer size
- k val Passphrase
- kfile infile Read passphrase from file
- K val Raw key, in hex
- S val Salt, in hex
- iv val IV in hex
- md val Use specified digest to create a key from the passphrase
- iter +int Specify the iteration count and force use of PBKDF2
- pbkdf2 Use password-based key derivation function 2
- none Don't encrypt
- * Any supported cipher
- rand val Load the file(s) into the random number generator
- writerand outfile Write random data to the specified file
- engine val Use engine, possibly a hardware device

Zad 1

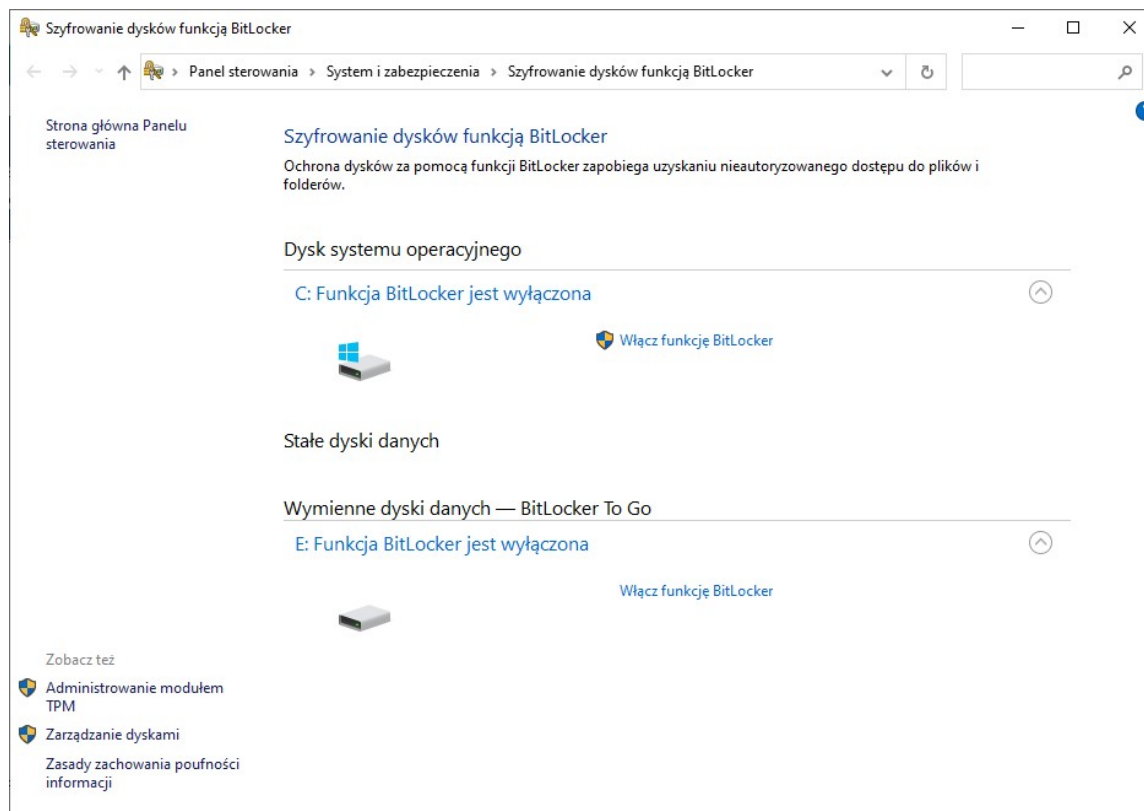
- Zaszzyfrować: plaintext.txt

```
openssl enc -e -pbkdf2 -aes-256-cbc -in  
plaintext.txt -out encrypted.txt
```

```
openssl enc -d -pbkdf2 -aes-256-cbc -in  
encrypted.txt -out decrypted.txt
```


Szyfrowanie dysków

- Bitlocker




Szyfrowanie dysków



- Bitlocker

Trusted Platform Module (TPM), znany również jako **ISO/IEC 11889**) to międzynarodowy standard bezpiecznego mikroprocesora kryptograficznego, dedykowanego mikrokontrolera zaprojektowanego do zabezpieczania sprzętu za pomocą zintegrowanych kluczy kryptograficznych. Termin ten może również odnosić się do chipa zgodnego ze standardem.

Szyfrowanie dysków

- Bitlocker



  Szyfrowanie dysków funkcją BitLocker (E:)

Wybierz sposób odblokowywania tego dysku

☐ Użyj hasła w celu odblokowania dysku
Hasło powinno zawierać wielkie i małe litery, cyfry, odstępy i symbole.

Wprowadź hasło

Ponownie wprowadź hasło

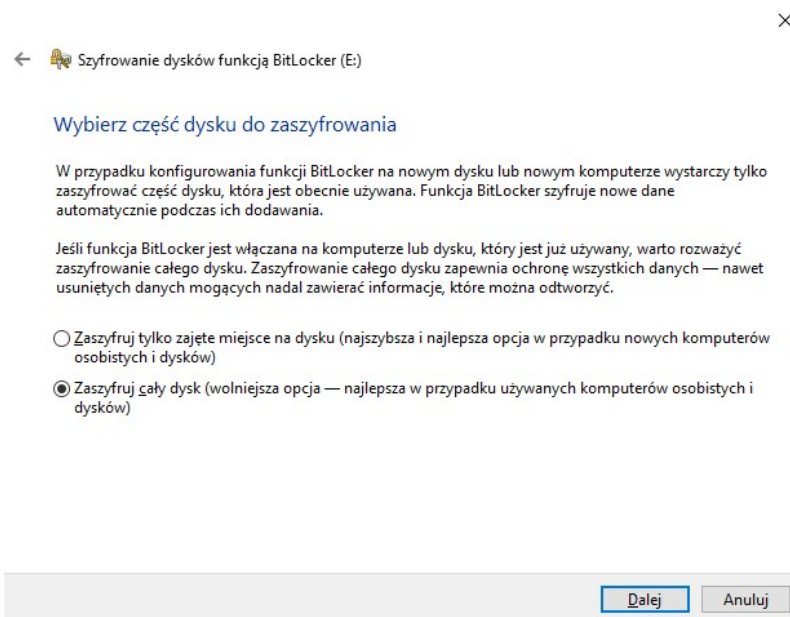
☐ Użyj mojej karty inteligentnej, aby odblokować dysk
Należy włożyć kartę inteligentną. Podczas odblokowywania dysku konieczne będzie podanie numeru PIN tej karty inteligentnej.

Dalej

Anuluj

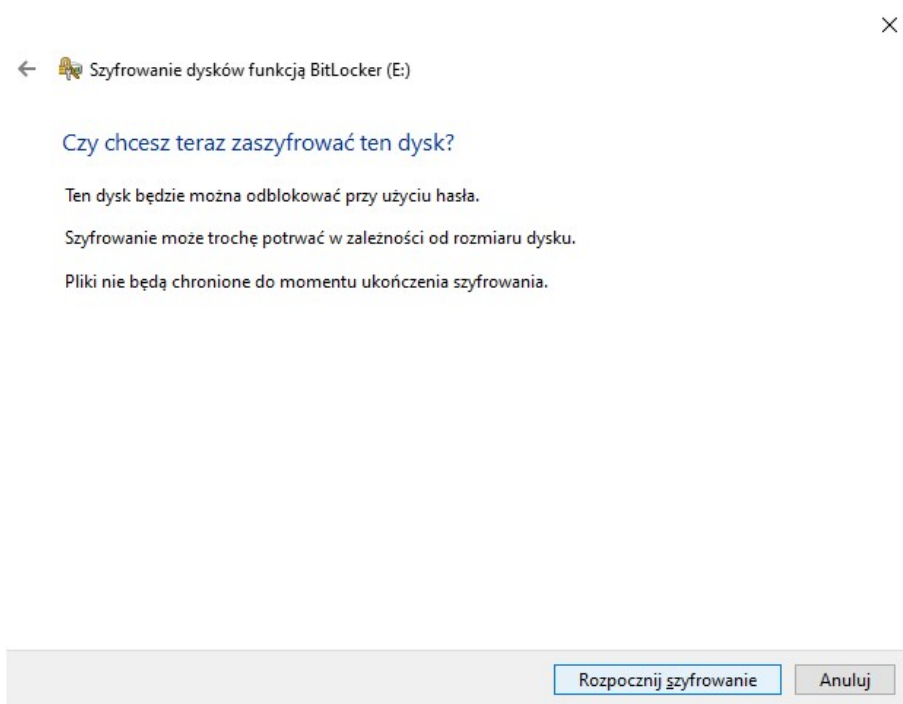
Szyfrowanie dysków

- Bitlocker



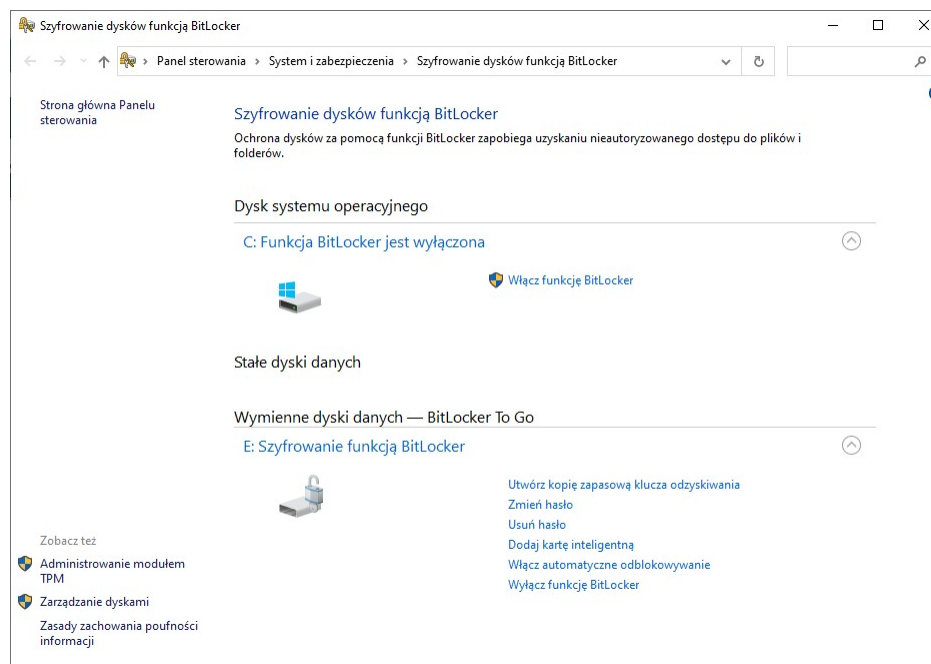
Szyfrowanie dysków

- Bitlocker










Szyfrowanie dysków

- Bitlocker



Szyfrowanie dysków

- Bitlocker

- >  Dysk lokalny (C:)
-  Dysk USB (E:)
-  Dysk USB (E:)
- >  Sieć
- ▼  Linux
 - >  docker-desktop-data
 - >  Ubuntu-20.04

BitLocker (E:)

Wprowadź hasło, aby odblokować ten dysk.

[Więcej opcji](#)

Odblokuj

- Bitlocker

```
LBA:0                                blok: 0
00000000: E8 58 90 2D 46 56 45 2D      46 53 2D 00 02 08 08 00    ex -FVE-FS-....
00000010: 00 00 02 00 00 F8 00 01      3F 00 FF 00 00 00 00 00    ....?..?..?..
00000020: 00 00 08 00 E0 1F 00 00      00 00 00 00 00 00 00 00    .....f.....
00000030: 01 00 06 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
00000040: 80 00 29 00 00 00 00 4E      4F 20 4E 41 4D 45 20 20    €. )....NO NAME
00000050: 20 20 46 41 54 33 32 20      20 20 33 C9 8E D1 BC FA    FAT32 3EŽNL0
00000060: 7B 8E C1 8E D9 BD 00 7C      AO FB 7D BA 7D BB FC AC    {ZÁŽŮ". ů }<-
00000070: 98 40 74 0C 48 74 0E B4      0E BB 07 00 CD 10 EB EF    @t.Ht.'...Ī.ed
00000080: A0 FD 7D EB E6 CD 16 CD      19 00 00 00 00 00 00 00    ý}ěčí.ī.....
00000090: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
000000A0: 3B DE 67 49 29 2E D8 4A      83 99 F6 A3 39 E3 D0 01    ;ǫgI).Ř ĭō9āĐ.
000000B0: 00 60 84 0C 00 00 00 00      00 60 B5 0C 00 00 00 00    ..'.....μ.....
000000C0: 00 B0 8A 0F 00 00 00 00      00 00 00 00 00 00 00 00    .°š.....
000000D0: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
000000E0: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
000000F0: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
00000100: 0D 0A 52 65 6D 6F 76 65      20 64 69 73 6B 73 20 6F    ..Remove disks o
00000110: 72 20 6F 74 68 65 72 20      6D 65 64 69 61 2E FF 0D    r other media."
00000120: 0A 44 69 73 6B 20 65 72      72 6F 72 FF 0D 0A 50 72    ".Disk error"..Pr
00000130: 65 73 73 20 61 6E 79 20      6B 65 79 20 74 6F 20 72    ess any key to r
00000140: 65 73 74 61 72 74 0D 0A      00 00 00 00 00 00 00 00    estart.....
00000150: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
00000160: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
00000170: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
00000180: 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00    .....
00000190: 00 00 00 00 00 00 00 00      78 78 78 78 78 78 78 78    .....xxxxxxxxxx
000001A0: 78 78 78 78 78 78 78 78      78 78 78 78 78 78 78 78    xxxxxxxxxxxxxxxxxx
000001B0: 78 78 78 78 78 78 78 78      78 78 78 78 78 78 78 78    xxxxxxxxxxxxxxxxxx
000001C0: 78 78 78 78 78 78 78 78      78 78 78 78 78 78 78 78    xxxxxxxxxxxxxxxxxx
000001D0: 78 78 78 78 78 78 78 78      78 78 78 78 78 78 78 78    xxxxxxxxxxxxxxxxxx
000001E0: 78 78 78 78 78 78 78 78      FF FF FF FF FF FF FF FF    xxxxxxxx
000001F0: FF FF FF FF FF FF FF FF      FF FF FF 00 1F 25 5A AA    ..,US

LBA:1                                blok: 1
00000200: 2D 88 24 87 30 0F 4E 8A      88 15 32 90 73 08 35 E2    - $#0.NŠ .2 s.5ā
00000210: 68 6A 55 25 92 D6 54 FF      56 02 52 3A B4 F8 97     hJūX;ŌT'V.R;U;0
00000220: 38 75 AE D1 BF 97 B6 0F      5C C6 C8 14 FC 05 93 BC   ;Ńēž-g.ŷČĈ.uŕŦ
00000230: 1F 84 E8 86 3A 71 D8 D9      32 C6 9A 55 30 EE 60 32   ,ĉt;qR0ÚČŠ0u1-2
00000240: EC C3 B1 EE 11 A0 35 AE      6A EC C9 72 E9 0D BB A7   ŠĀİ.5 ſojēÉrē.ŵş
00000250: B9 08 CB FS 14 6E E8 7E      7F 74 46 BA 10 8D 9F 9C   q.AĖ.nčŹtFş.Tēz
00000260: 88 F3 3C 6E C6 D9 82 5A      49 26 54 22 70 E0 5E 11   <&cŋCŲ.Zİ&Z.pŕŕ.
00000270: 60 51 36 F3 F2 00 05 D1      EE 6F 77 6B 3A F7 3F 28   'Q6Gñ..Niowk:>+{
00000280: 2D 9F 8F AO 4A 26 94 98      0B B8 9E CA E1 11 C7 A7   -žŹ J&n;. ŽēĎ.CŞ
00000290: 73 10 5F 5D 31 F0 8D 7A      DA 36 24 D6 1F F6 C2 8E   s_ŶlđŹŭ6Ś0.ôĀZ
000002A0: 74 FF 01 F1 14 30 10 F9      1D 38 44 70 44 78 4A C5   t'.ñ.0.Ź.8DPdpxŹ
000002B0: C9 88 7D 7F 3A 95 08 B1      D0 61 27 F5 48 03 4E 10   ÉŶŶ) :•.âDa'ĤH.N
000002C0: F7 A5 69 92 AC BA 18 68      EA 94 0F 5E 5B 82 74 81   +ŝıŴş.s.hăŶ.İ.t.
000002D0: 31 8E 48 83 E3 DD 9F 6C      56 6C B1 4E 55 4D 84 E8   lŻH äŸŹŶlŷ+NUMŷĊ
000002E0: F2 7C 45 DB C5 EA 51 0F      CD 72 08 58 DA 10 19 88   ħŶEÜŁęŲ.İŕ.XŬ.
000002F0: 0C FE FE 28 4A 75 83 58      C0 25 1A 6A 4C 58 25 AB   .İŢ(ŲŲ [R&.ŶLX&
00000300: 3E 29 CA 73 74 E3 C8 24      19 32 3C 05 A3 19 B0 16   )ĄSt&ČŞ.2<.č.Ű
00000310: B0 14 09 18 DD 0D 7B F4      E6 57 AE D6 18 87 98 18   .Ų.Ŷ.(ôĥwŲŲ.).
00000320: B7 AA BF 81 7D B9 41 18      08 27 06 C3 C1 03 A8 44   .ŞŹ}qā..ĀĀ.D
00000330: AA 11 7E 7C 9F 71 22 75      68 1B 8B 2D F1 48 0D 1A   ſ_ŶŶ"q"uh..-ŔK..
00000340: 2B 9A 40 48 BF 87 3F 0D      F3 4D CE 63 72 CF 19 DA   +@M&Ŷ.Ų.6ŨİcŕDŲ
00000350: 6D F2 55 0D 01 26 8C 12      C6 F1 DB F5 9E 07 9D 2D   mŨŬ. &Ŝ.ČŇŮŹŷ.--
00000360: 96 69 79 59 9B 1F DF A8      7B C5 C0 49 60 E8 97 D8   -iyY}'B{'[ŔİŶĊ-ĊŔ
00000370: B8 B2 27 22 93 D9 3F 25      14 A0 DA 2D 3E E8 5F EC   ..ŲŲŲ.Ų.Ų->ĉĊŶ
00000380: 1F E7 B1 OA B8 CB 37 E5      B2 CF 10 7C 83 9C 3F 11   .çŷ.ĒŶŶ.ŲŶ) ŜŶ.
00000390: FB EF 11 58 3A 65 71 13      82 1B C8 82 0E 16 A7 11   řŲŶ.X.e.q.;.Č.Ŷ.Ŷ
000003A0: F1 8C C7 CE 6C E9 49 CB      1F 24 00 86 10 31 F6 6E   řŠĆİĽİĚŶ$.Ŷ.t.1a
```


Szyfrowanie z wykorzystaniem algorytmów asymetrycznych

- **Klucz publiczny** używany jest do zaszyfrowania informacji, **klucz prywatny** do jej odczytu. Ponieważ klucz prywatny jest w wyłącznym posiadaniu adresata informacji, tylko on może ją odczytać. Natomiast klucz publiczny jest udostępniony każdemu, kto zechce zaszyfrować wiadomość.
- Ponieważ kryptografia asymetryczna jest o wiele wolniejsza od tradycyjnej, nie szyfruje się wiadomości za pomocą kryptosystemów asymetrycznych. Zamiast tego szyfruje się jedynie klucz a następnie wykorzystuje się szyfrowanie symetryczne (strumieniowe lub blokowe)

Szyfrowanie z wykorzystaniem algorytmów asymetrycznych

- RSA
- `openssl genpkey -algorithm RSA -out key1.pem`
- `openssl pkey -in key.pem -pubout -out pub_key1.pub`
- `openssl asn1parse -in pub_key1.pub`
- `openssl rand -hex 32 > klucz.hex`
- `openssl pkeyutl -encrypt -in klucz.hex -pubin -inkey pub_key1.pub -out ciph_key1.hex`
- `openssl pkeyutl -decrypt -in ciph_key1.hex -inkey key1.pem -out deciph_key1.hex`

Szyfrowanie pakietów

- SSL/TLS
- IPSec

Szyfrowanie pakietów

SSL/TLS



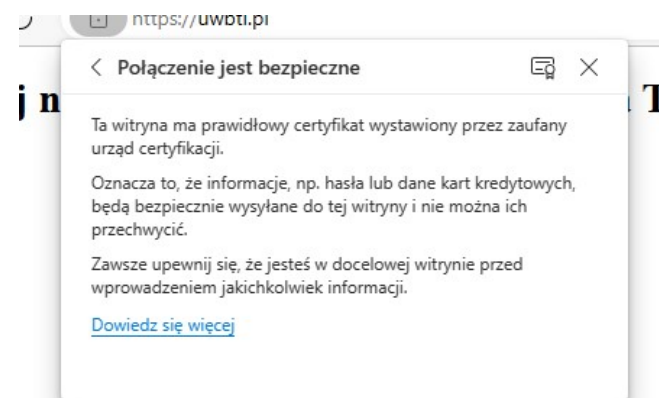
Witaj na stronie zajęć z Bezpieczeństwa Teleinformatycznego

[Lab 1.](#)

[Lab 2.](#)

[Lab 3.](#)

[Lab 4.](#)



Szyfrowanie pakietów

IPSec

FortiClient
File Help

FortiClient VPN

Upgrade to the full version to access additional features and receive technical support.

New VPN Connection

VPN: SSL-VPN **IPsec VPN** XML

Connection Name:

Description:

Remote Gateway: + Add Remote Gateway

Authentication Method: Pre-shared key **Pre-shared key** X.509 Certificate

Authentication (XAuth): ☒ Prompt on login ☐ Save login ☐ Disable

Failover SSL VPN: [None]

+ Advanced Settings

Cancel Save

Szyfrowanie pakietów

SSL/TLS i IPsec to dwa różne protokoły służące do zabezpieczania komunikacji sieciowej. Oto kilka istotnych różnic między nimi:

- Warstwa protokołu: SSL/TLS to protokół szyfrowania stosowany na warstwie aplikacji, podczas gdy IPsec działa na warstwie sieciowej.
- Cel: SSL/TLS jest powszechnie stosowany do zabezpieczania komunikacji w sieci WWW, czyli do zabezpieczania przeglądania stron internetowych, przesyłania poczty e-mail czy innych aplikacji internetowych. Z kolei IPsec jest zazwyczaj stosowany do zabezpieczania połączeń VPN, czyli do tworzenia tunelu w celu przesyłania danych między dwoma lub więcej sieciami.
- Przepustowość: IPsec jest zazwyczaj bardziej wydajny od SSL/TLS, ponieważ działa na warstwie sieciowej i może szyfrować cały ruch sieciowy. SSL/TLS działa na warstwie aplikacji, co oznacza, że musi szyfrować każde zapytanie i odpowiedź oddzielnie, co może prowadzić do nieco wolniejszej przepustowości.
- Konfiguracja: Konfiguracja IPsec jest zazwyczaj bardziej złożona niż konfiguracja SSL/TLS, ponieważ IPsec wymaga skonfigurowania tunelu między dwoma sieciami, a także ustalenia kluczy szyfrujących. W przypadku SSL/TLS większość konfiguracji odbywa się automatycznie i użytkownik nie musi ręcznie konfigurować kluczy szyfrujących.



Szyfrowanie



THE END