1110111011011

1001100

1100110011100

1010101010101010

00011001001

O łamaniu haseł

Kryptografia i kryptoanaliza. Bezpieczeństwo haseł. Brute force. Metoda słownikowa

Hacking i Cyberbezpieczeństwo | Giganci Programowania

Opracowanie: Michał Suchoń

Przypomnienie

1. Co to VPN, a co to Proxy? Jaka jest różnica między nimi?
2. W jaki sposób wejść do Deep Web'u. Czy to bezpieczne i czy legalne?
3. A co to Tor?

Przypomnienie

1. Co to VPN, a co to Proxy? Jaka jest różnica między nimi?

   Proxy – z ang. *Pośrednik; usługa serwera pośredniczącego, zapobiega bezpośredniemu łączeniu się serwer np. celem uzyskania dostępu do strony internetowej (serwer docelowy widzi adres IP proxy – nie nasz). Związana głównie z protokołem HTTP.**

   VPN – *Virtual Private Network – ang. wirtualna sieć prywatna*; usługa tunelowania połączeń, gdzie naszym „wyjściem na świat" jest serwer VPN, który każde przesyłane dane przekazuje przez sieć rozlegle rozmieszczonych (na całym świecie!) urządzeń trasujących (np. routery / serwery). Docelowe urządzenia „widzą" adres IP serwera VPN, nie nasz.

   Podstawową różnicą między VPN'em a Proxy jest to, że Pośrednik działa w sposób niezabezpieczony i w ograniczeniu głównie do protokołu HTTP; VPN zaś może używać różne protokoły i porty, a sieć tunelowa jest zaszyfrowana.

2. W jaki sposób wejść do Deep Web'u. Czy to bezpieczne i czy legalne?

3. A co to Tor?

Przypomnienie

1. Co to VPN, a co to Proxy? Jaka jest różnica między nimi?

   Proxy – z ang. *Pośrednik; usługa serwera pośredniczącego, zapobiega bezpośredniemu łączeniu się serwer np. celem uzyskania dostępu do strony internetowej (serwer docelowy widzi adres IP proxy – nie nasz). Związana głównie z protokołem HTTP.**

   VPN – *Virtual Private Network – ang. wirtualna sieć prywatna*; usługa tunelowania połączeń, gdzie naszym „wyjściem na świat" jest serwer VPN, który każde przesyłane dane przekazuje przez sieć rozlegle rozmieszczonych (na całym świecie!) urządzeń trasujących (np. routery / serwery). Docelowe urządzenia „widzą" adres IP serwera VPN, nie nasz.

   Podstawową różnicą między VPN'em a Proxy jest to, że Pośrednik działa w sposób niezabezpieczony i w ograniczeniu głównie do protokołu HTTP; VPN zaś może używać różne protokoły i porty, a sieć tunelowa jest zaszyfrowana.

2. W jaki sposób wejść do Deep Web'u. Czy to bezpieczne i czy legalne?

   Wejście do Deep Web'u jest możliwe przez sieć Tor, z którą możemy się połączyć poprzez TorBrowser. Korzystanie z sieci Tor jest w pełni legalne w Polsce, lecz może się wiązać z wieloma zagrożeniami podczas nieostrożnego korzystania z sieci cebulowej.

3. A co to Tor?

Przypomnienie

1. Co to VPN, a co to Proxy? Jaka jest różnica między nimi?

   Proxy – z ang. *Pośrednik*;* usługa serwera pośredniczącego, zapobiega bezpośredniemu łączeniu się serwer np. celem uzyskania dostępu do strony internetowej (serwer docelowy widzi adres IP proxy – nie nasz). Związana głównie z protokołem HTTP.

   VPN – *Virtual Private Network – ang. wirtualna sieć prywatna*; usługa tunelowania połączeń, gdzie naszym „wyjściem na świat" jest serwer VPN, który każde przesyłane dane przekazuje przez sieć rozlegle rozmieszczonych (na całym świecie!) urządzeń trasujących (np. routery / serwery). Docelowe urządzenia „widzą" adres IP serwera VPN, nie nasz.

   Podstawową różnicą między VPN'em a Proxy jest to, że Pośrednik działa w sposób niezabezpieczony i w ograniczeniu głównie do protokołu HTTP; VPN zaś może używać różne protokoły i porty, a sieć tunelowa jest zaszyfrowana.

2. W jaki sposób wejść do Deep Web'u. Czy to bezpieczne i czy legalne?

   Wejście do Deep Web'u jest możliwe przez sieć Tor, z którą możemy się połączyć poprzez TorBrowser. Korzystanie z sieci Tor jest w pełni legalne w Polsce, lecz może się wiązać z wieloma zagrożeniami podczas nieostrożnego korzystania z sieci cebulowej.

3. A co to Tor?

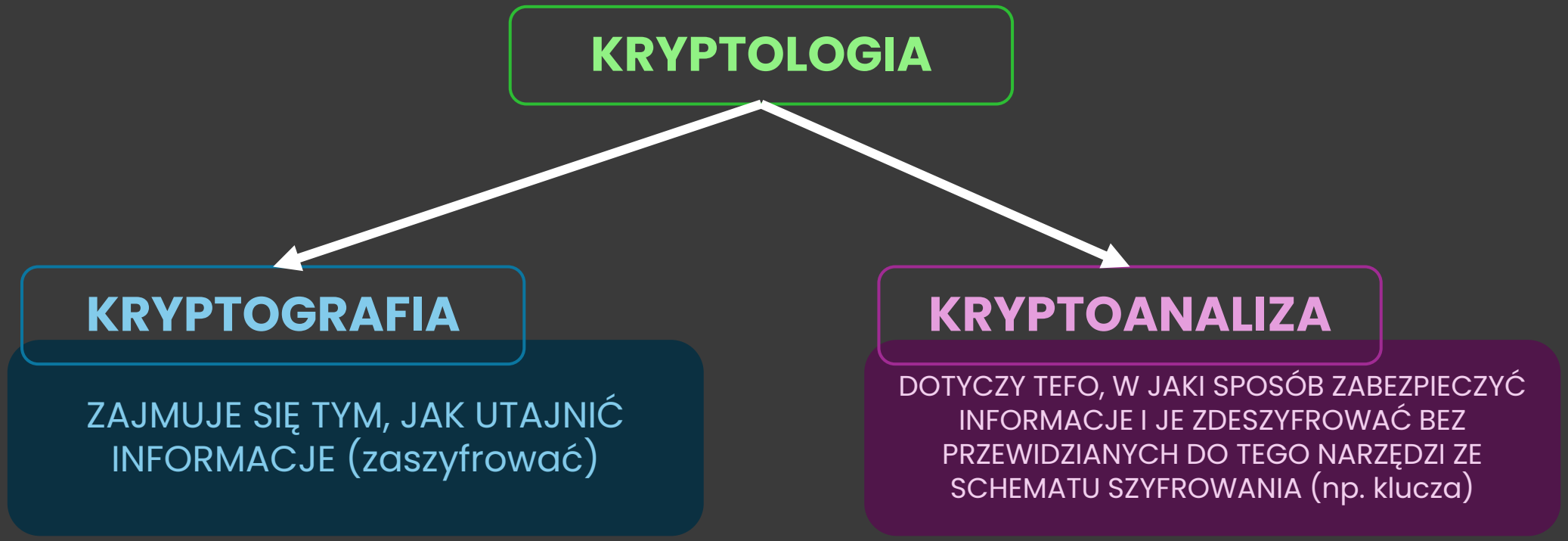
   Tzw. *sieć cebulowa*, opiera się o tunelowanie sieci w taki sposób, aby wszelki ruch sieciowy z naszego komputera pozostał ukryty.

Czym jest Kryptologia?

Kryptologia

Dziedzina nauki, badająca w jaki sposób przekazywać informacje w sposób zabezpieczony przed niepowołanym dostępem. Jest silnie skorelowana z matematyką i informatyką, ale także blisko związana z teorią informacji, inżynierią systemów oraz cyberbezpieczeństwem (bez niej nie jest w stanie istnieć!)

Dziedziny kryptologii

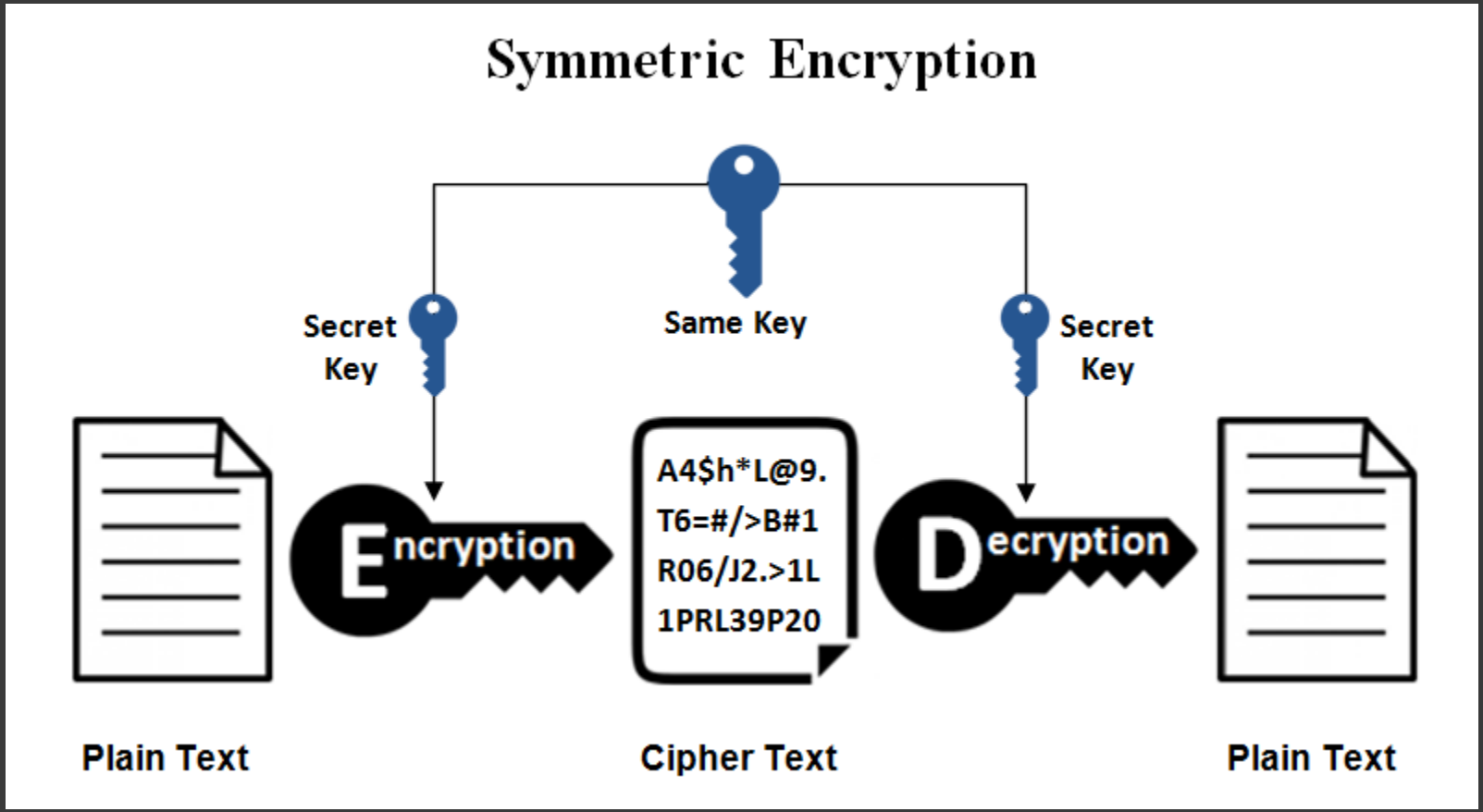


Jak zaszyfrować informacje?

rodzaje mechanizmów szyfrowania

Kryptografia symetryczna

POLEGA NA ZASZYFROWANIU WIADOMOŚCI PRZY UŻYCIU TEGO SAMEGO KLUCZA



https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

Czas na małą zagadkę…

cnc oc mqwc

Czas na małą zagadkę…

cnc oc mqwc

Podpowiedź (1): *veni, vidi, vici…*

Czas na małą zagadkę…

cnc oc mqwc

Podpowiedź (1): *veni, vidi, vici…*
Podpowiedź (2): *Cesarz Imperium Rzymskiego*

Czas na małą zagadkę…

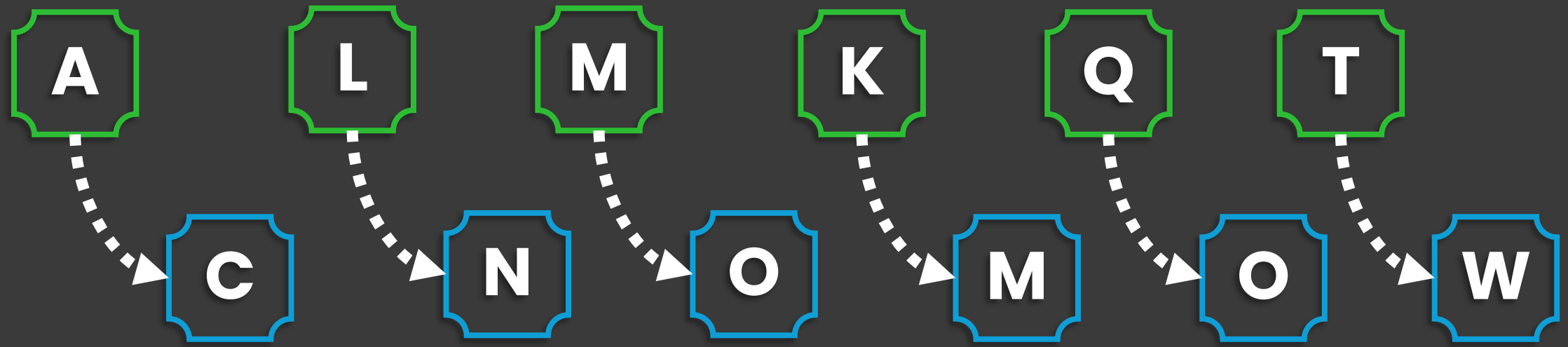
cnc oc mqwc

Podpowiedź (1): *veni, vidi, vici…*
Podpowiedź (2): *Cesarz Imperium Rzymskiego*
Podpowiedź (3): *Julisz Cezar*

SZYFR CEZARA

cnc oc mqwc

PRZESUNIĘCIE O 2 LITERY



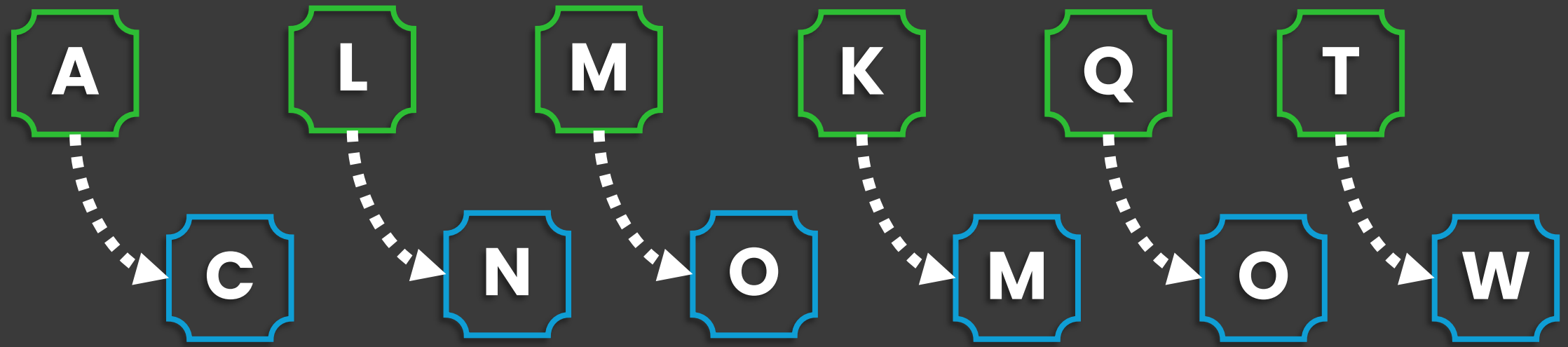
*Polega na przesunięciu liter o **n kroków**, dzięki czemu pierwotna treść informacji staje się trudna do odczytania osobie, nieznającej mechanizmu. Podobno stworzone przez samego cesarza rzymskiego Juliusza Cezara jako sposób na zabezpieczenie rozkazów dla oddziałów żołnierzy, tak by w razie porwania posłańca przez nieprzyjaciela, wróg nie był w stanie odczytać planów bojowych.*

SZYFR CEZARA | rozwiązanie

a l a ma k o t a

cnc oc mqwc

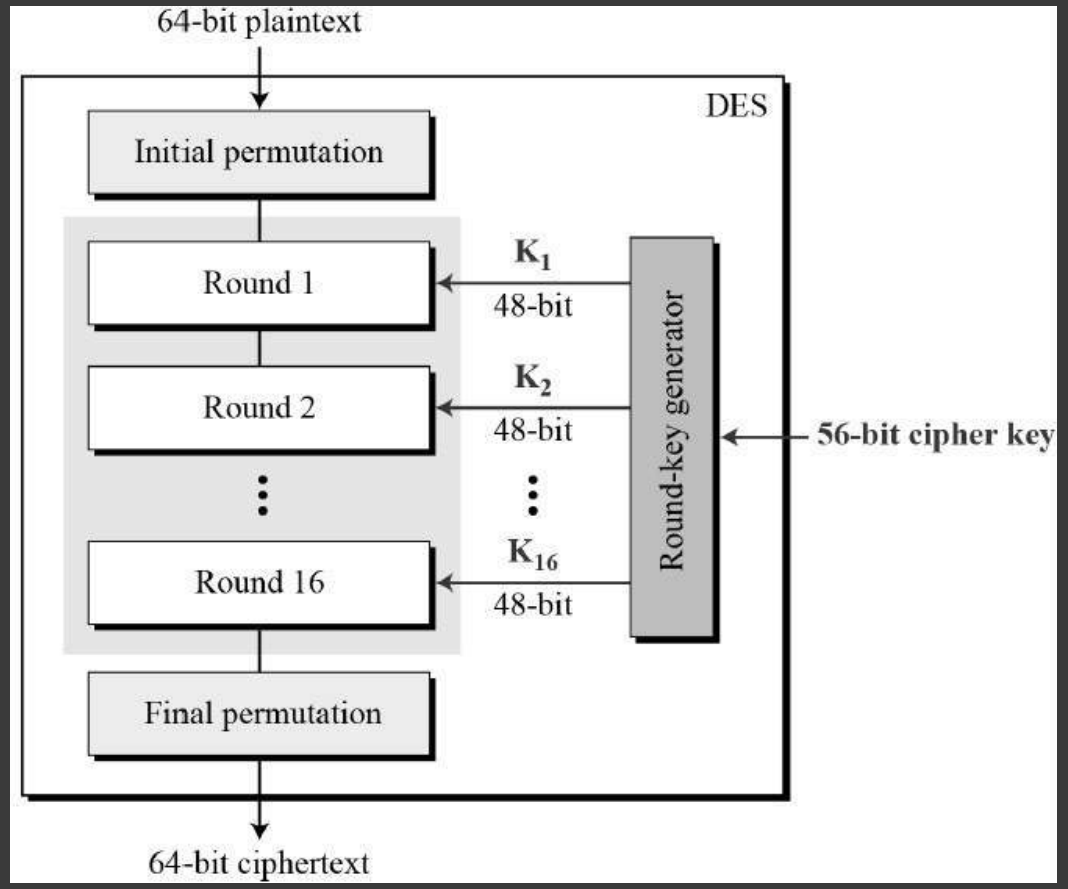
PRZESUNIĘCIE O 2 LITERY

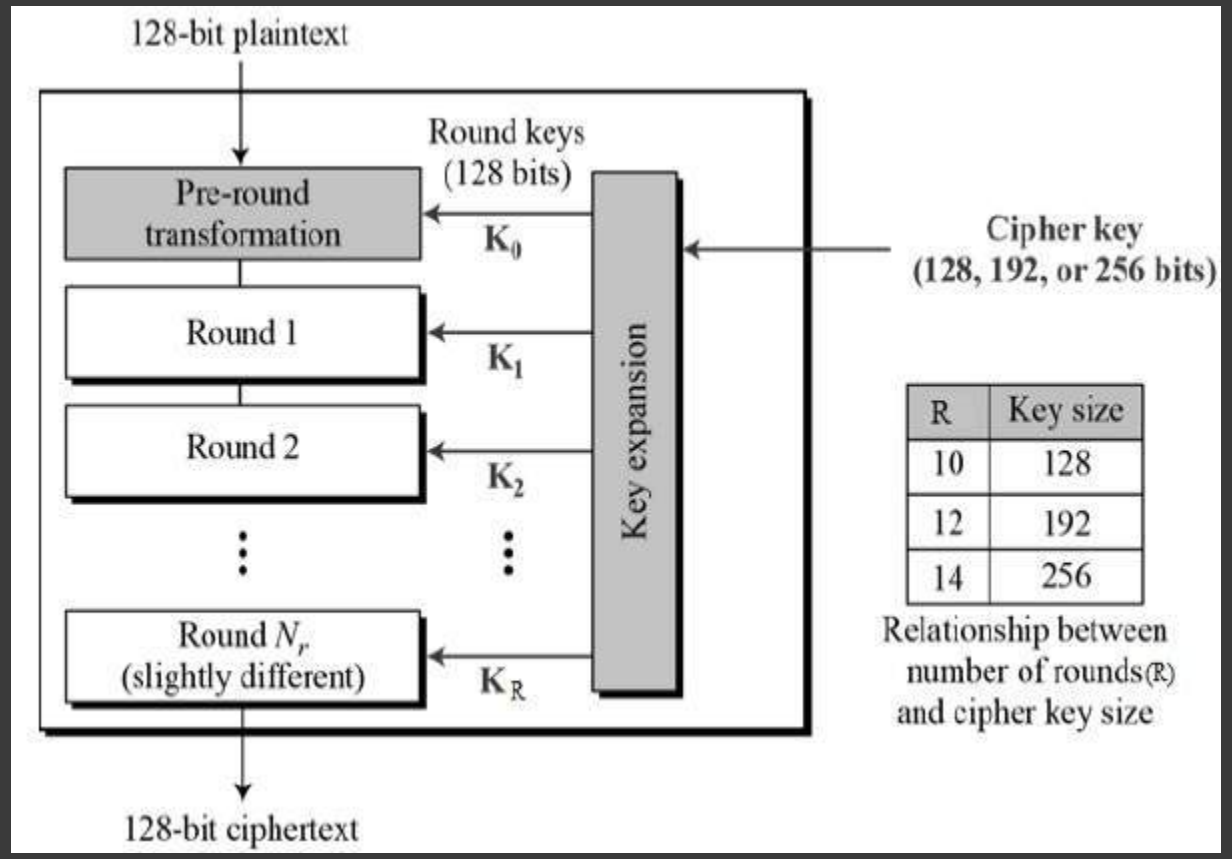


*Polega na przesunięciu liter o **n kroków**, dzięki czemu pierwotna treść informacji staje się trudna do odczytania osobie, nieznającej mechanizmu. Podobno stworzone przez samego cesarza rzymskiego Juliusza Cezara jako sposób na zabezpieczenie rozkazów dla oddziałów żołnierzy, tak by w razie porwania posłańca przez nieprzyjaciela, wróg nie był w stanie odczytać planów bojowych.*

SZYFRY BLOKOWE

SPOSÓB SZYFROWANIA, KTÓREGO ZAŁOŻENIEM JEST DZIELENIE JAWNEJ WIADOMOŚCI NA DWA BLOKI, A NASTĘPNIE SZYFROWANIE W TZW. *RUNDACH* PRZY UŻYCIU KLUCZA. NA KOŃCU, BLOKI SĄ ŁĄCZONE W ZASZYFROWANĄ CAŁOŚĆ. NAJPOPULARNIEJSZE SZYFRY BLOKOWE TO **DES** ORAZ **AES**





R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

DES i AES **DES** (64-bitowy ciąg; klucz 56-bitowy)

1. PERMUTACJA POCZĄTKOWA przestawienie bitów w bloku zgodnie z tzw. tablicą permutacji

2. PODZIELENIE BLOKU na dwa osobne po 32 bity

3. DOBÓR 56 BITÓW Z 64 BITÓW KLUCZA wybrane bity są dzielone na dwa kolejne bloki po 28 bitów

4. 16 RUND OPERACJI za pomocą tzw. *Funkcji Feistela* przy użyciu klucza

5. ŁĄCZENIE WSZYSTKICH PODZIELONYCH BLOKÓW DANYCH W 16 RUNDACH (nadal w dwóch osobnych blokach) za pomocą operacji XOR (1 1 lub 0 0 => 0; 1 0 lub 0 1 => 1)

6. PERMUTACJA KOŃCOWA łączenie danych w jeden cały łańcuch znaków

DES i AES **AES (128-bitowy ciąg; klucz 128/192/256-bitowy)**

1. PRZYGOTOWANIE PODKLUCZY generowanie podklucza początkowego oraz po jednym dla każdej rundy szyfrującej

2. RUNDA INICJUJĄCA każdy bajt w bloku danych jest dodawany do przypisanego mu bajtowi podklucza za pomocą operacji XOR

3. RUNDY SZYFRUJĄCE **9** powtórzeń dla klucza **128-bit** ; **11** powtórzeń dla klucza **192-bit** ; **11** powtórzeń dla klucza **192-bit**

4. RUNDA KOŃCZĄCA kroki te same co w rundach szyfrujących z pominięciem operacji mnożenia kolumn (Operacja MC)

Wyzwanie nr 1 | Base64

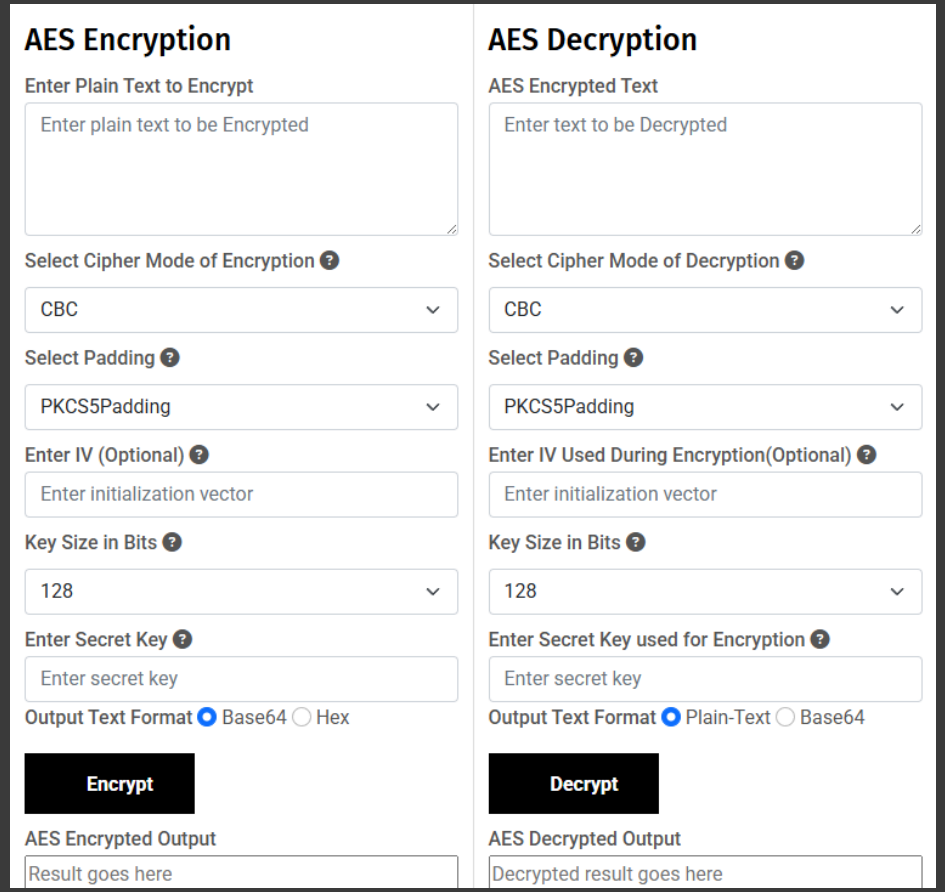
1. Wejdź na stronę https://www.base64encode.org/
2. Wybieramy na górze „Encode” na górze
3. Wpisujemy treść do zaszyfrowania (np. giganci)
4. Klikamy Encode
5. Otrzymamy ciąg Z2lnYW5jaQ**==**
6. Dodajemy do wiadomości 1 -> *giganci1* EFEKT: Z2lnYW5jaTE**=**
7. Dodajemy kolejną literę, np. b -> *giganci1b* EFEKT: Z2lnYW5jaTFi
8. Znaki równości zniknęły. Gdy dodamy kolejny znak, zostaną dodane kolejne dwa znaki równości (bloki)
9. Kopiujemy zaszyfrowaną wiadomość, wybieramy na górze „Decode” i sprawdzamy, czy otrzymamy wyjściową treść.

WYJAŚNIENIE:

Blok gdy zostanie zapełniony (nie będzie pustego znaku oznaczonego znakiem =), przy kolejnym dodaniu znaku dodawane są kolejne puste znaki do ich ewentualnego zapełnienia.

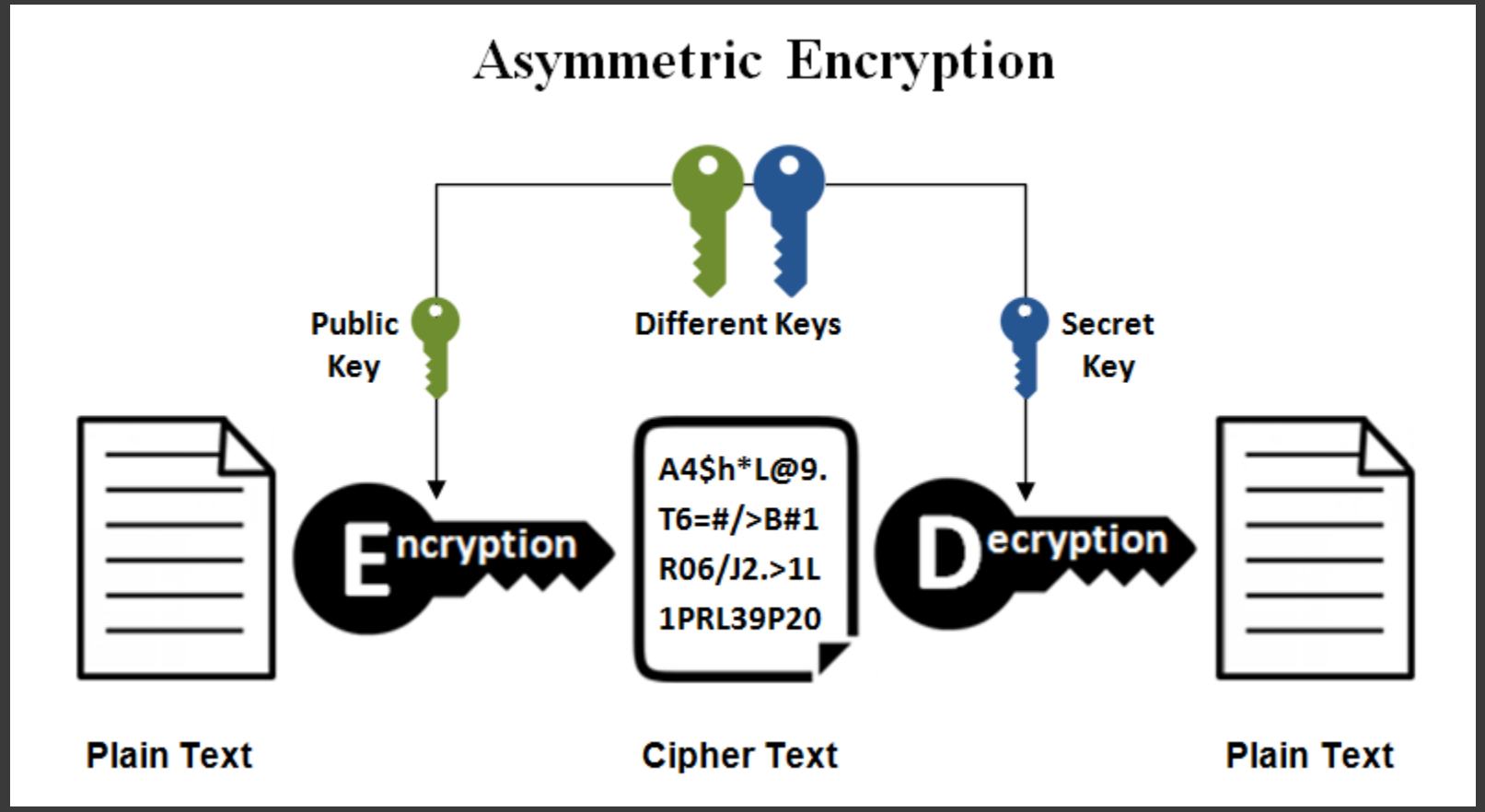
Wyzwanie nr 2 | AES

1. Wygeneruj klucz szyfrujący 256 bitowy (*CodeIgniter Encryption Keys*) ze strony: https://randomkeygen.com/
2. Przejdź do strony szyfrującej wiadomości: https://www.devglan.com/online-tools/aes-encryption-decryption
3. Po lewej możemy zaszyfrować wiadomość, po prawej deszyfrujemy zaszyfrowaną (W sekcji „Enter Secret Key" należy podać klucz, wygenerowany ze strony wyżej)



Kryptografia asymetryczna

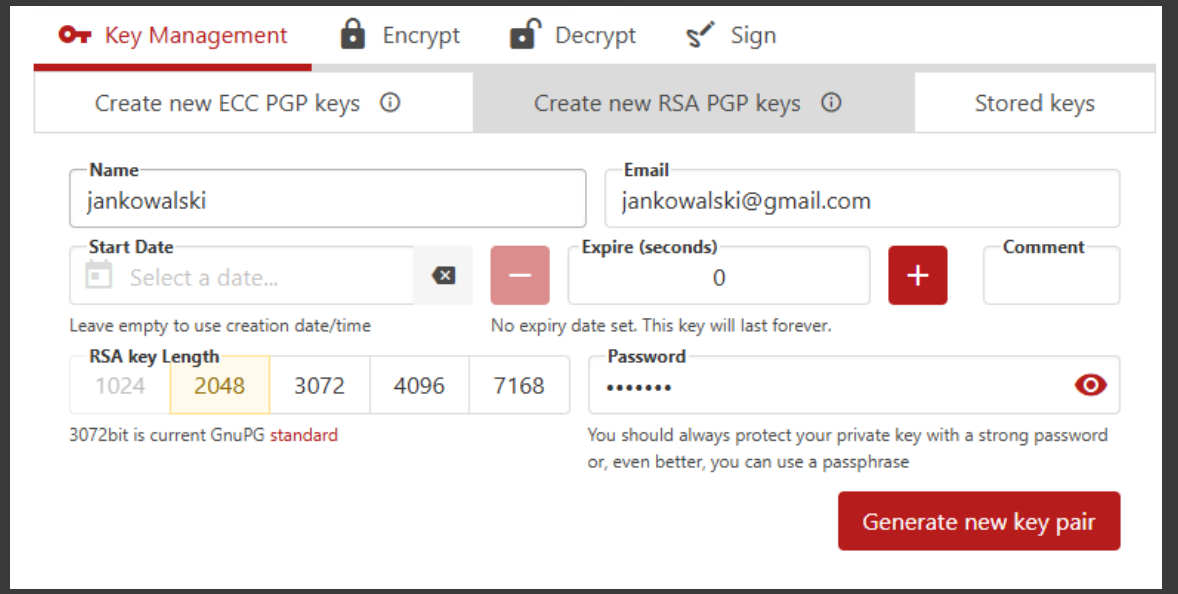
SZYFROWANIE I DESZYFROWANIE OPIERA SIĘ O DWA KLUCZA – **PUBLICZNY** (NADAWCY, DO ZASZYFROWANIA) ORAZ **PRYWATNY** (ODBIORCY, DO DESZYFROWANIA)



https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

Wyzwanie nr 3 | POCZTA PGP

1. Wchodzimy na stronę https://codref.org/tools/pgp/
2. W zakładce „Key Management" wybieramy opcję *Create new RSA PGP keys*. Wypełniamy pola: *name* (nazwa klucz), *email* (wiadomo), RSA key Length (długość klucza publicznego), password (hasło zabezpieczające).
3. Klikamy *Generate new key pair*
4. W zakładce *Encrypt* szyfrujemy wiadomość, w *Decrypt* deszyfrujemy



Zady i walety

~~(…że wiecie, zamiast wady i zalety haha)~~

KRYPTOGRAFIA SYMETRYCZNA

Aye!

- ✓ szybszy niż asymetryczny (używa tylko jednego klucza)
- ✓ Korzysta z uwierzytelniania za pomocą hasła
- ✓ Jest prosty w implementacji
- ✓ przez szybkość zdecydowanie zalecany przy dużych zbiorach danych jak pliki, bazy danych, kopie zapasowe…

Nay!

- × Używa tego samego klucza do szyfrowania i deszyfrowania, co stwarza ryzyko ujawnienia klucza osobom trzecim (a znajomość go to jak posiadanie przycisku atomowego…)
- × Nie jest skalowalny, tzn. udostępniane klucze wielu osobom mogą być nie przystosowane pod różnorodnych użytkowników (i ich zapotrzebowania)

Zady i walety

~~(…że wiecie, zamiast wady i zalety haha)~~

KRYPTOGRAFIA ASYMETRYCZNA

Aye!

- ✓ Posiada dwa odrębne od siebie klucze, z czego jeden (publiczny) jest ogólnodostępny i przystosowany do rozpowszechnienia i drugi (prywatny), przeznaczony jedynie dla konkretnego usera
- ✓ Idealnie nadaje się w dużych sieciach, dzięki czemu łatwiej komunikować się przy posiadanej parze kluczy.

Nay!

- × Szyfrowanie asymetryczne jest wolniejsze niż symetryczne pod względem wydajności
- × Znacznie trudniejsze do implementacji
- × Wykorzystuje klucze o dużej długości

Co lepsze?

Co lepsze?

To nie ma tak, że dobrze albo że niedobrze…

Hashowanie

Metoda zabezpieczenia informacji (głównie haseł), poprzez przypisanie dowolnie dużemu ciągowi zestaw pseudolosowych znaków za pomocą tzw. funkcji hashującej (skrótu / mieszającej) poprzez jeden z dostępnych algorytmów, np.: SHA1, SHA256, SHA512, MD5 (niepolecane), bcrypt – obecnie najlepsze. Główną cechą hashowania jest to, że zahashowane hasło nie da się odwrócić („zdeshashować” do tekstu jawnego); można wyłącznie porównać ciągi.

Metody łamania haseł

ALGORYTM SIŁOWY (BRUTE FORCE) – metoda sprawdzania wszystkich możliwych kombinacji znaków w celu znalezienia hasła

ALGORYTM SŁOWNIKOWY (DICTIONARY) – metoda sprawdzania kombinacji haseł za pomocą podzbiorów haseł, pochodzących np. z wycieków baz danych

SPRAWDZANIE SIŁY HASŁA: http://password-checker.online-domain-tools.com/

Wyzwanie nr 4 | HASHOWANIE

Wykorzystując język C#, napisz program do hashowania hasła

ROZWIĄZANIE

Wyzwanie nr 5 | SZUKANIE HASHA

Wykorzystując język C#, napisz program do znajdowania *hasha* podanego hasła jawnie (Podajemy pin np. 5 cyfrowy w zmiennej i próbujemy go znaleźć, przeszukując możliwe hashe). Uwaga! Program wykorzystuje metody zaimplementowane w poprzednim ćwiczeniu!

ROZWIĄZANIE

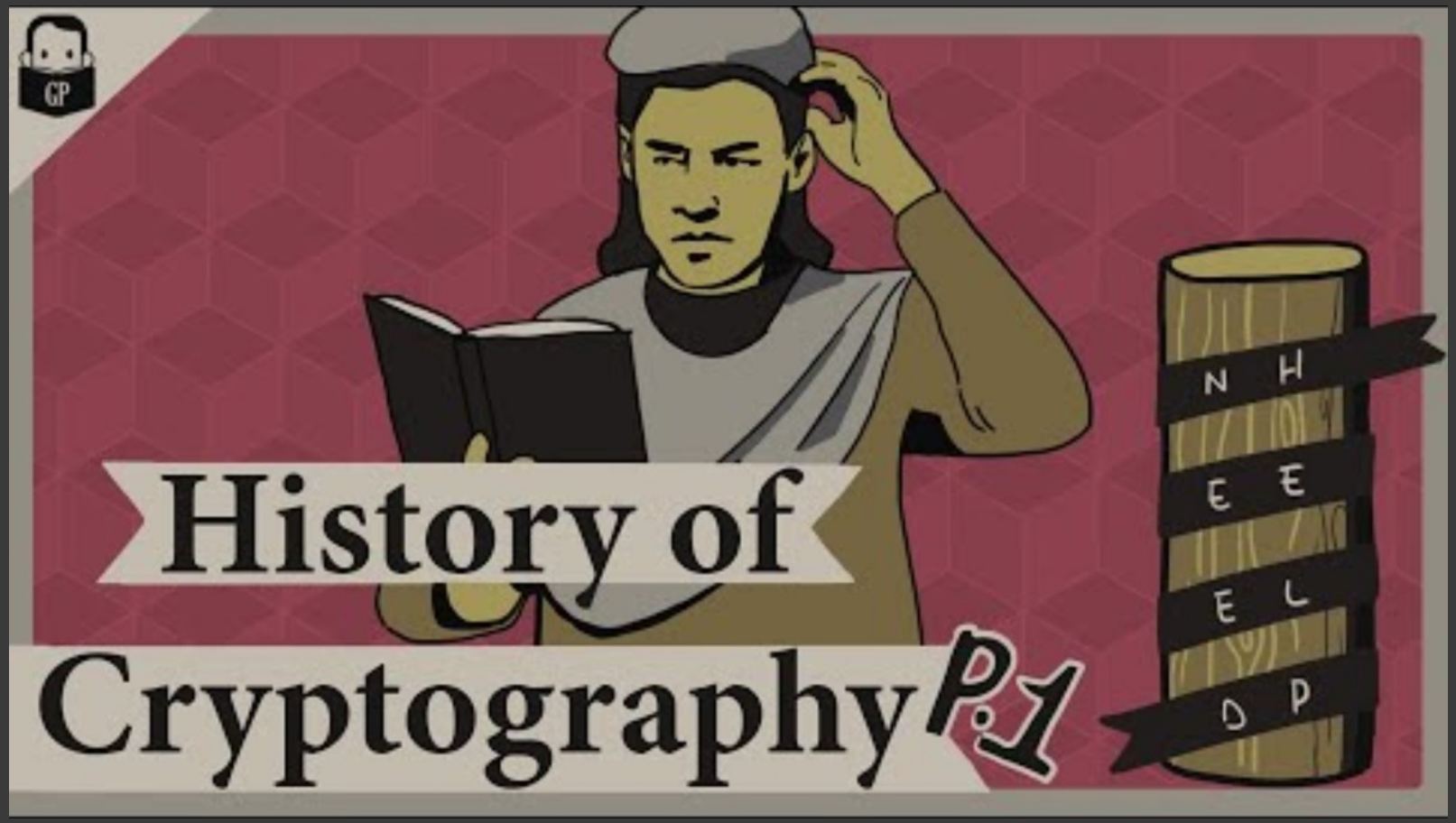
Podsumowanie

1. Czym się różni kryptografia symetryczna od asymetrycznej? W jaki sytuacjach sprawdzą się oba rodzaje?
2. Na czym opiera się szyfrowanie blokowe?
3. Po co hashuje się hasła? Co sprawia, że jest to metoda skuteczna?

Polecam!

ŹRÓDŁO:
https://www.youtube.com/watch?v=_GxNLWBHxvI&pp=ygUcc3p5ZnJvd2FuaWUgIG1hdGV1c3ogY2hyb2Jvaw%3D%3D

Polecam!



ŹRÓDŁO:
https://www.youtube.com/watch?v=9pp9YpginNg&pp=ygUVaGlzdG9yaWEga3J5cHRvZ3JhZmlp

‹ POWRÓT DO ZADANIA

Rozwiązanie - 4

1. Uruchamiamy aplikację konsolową z frameworkiem .Net -> *Console App (.Net Framework)*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Security.Cryptography;

namespace Hashowanie
{
    internal class Program
    {


        static void Main(string[] args)
        {

        }
    }
}
```

Ważne do zaimportowania!

Kod: **https://pastebin.com/a5rPNvVb**

‹ POWRÓT DO ZADANIA

Rozwiązanie - 4

2. Tworzymy metodę hashującą podane hasło (tworzymy ją nad Mainem, wewnątrz klasy Program) -> *HashString(string)*

```
static public string HashString(string str)
{
    byte[] data = Encoding.UTF8.GetBytes(str);
    byte[] hash;

    using (SHA512 shaM = new SHA512Managed())
    {
        hash = shaM.ComputeHash(data);
    }
    string result = ByteArrayToString(hash);
    return result;
}
```

Kod: **https://pastebin.com/a5rPNvVb**

‹ POWRÓT DO ZADANIA

Rozwiązanie - 4

3. Zwracana wartość jest jednak zwracana jako tablica bajtów. Musimy więc stworzyć metodę, która zamieni nam tablicę bajtów na ciąg heksadecymalnego (z którego przejdziemy do stringa)-> *ByteArrayToString(byte)*

```
static public string ByteArrayToString(byte[] b)
{
    StringBuilder hex = new StringBuilder(b.Length * 2);
    foreach(byte element in b)
    {
        hex.AppendFormat("{0:x2}", element);
    }
    return hex.ToString();
}
```

Kod: **https://pastebin.com/a5rPNvVb**

‹ POWRÓT DO ZADANIA

Rozwiązanie - 4

4. Na końcu, w głównej pętli programu Main wywołujemy funkcję, wczytując hasło jawnie od użytkownika -> *Sprawdzamy, czy działa!*

```
        static void Main(string[] args)
        {
            Console.WriteLine("Napisz hasło (jawnie): ");
            string hasloJawnie = Console.ReadLine();
            Console.WriteLine($"Przed hashowaniem: {hasloJawnie}");
            Console.WriteLine($"Po hashowaniu: {HashString(hasloJawnie)}");
            Console.ReadKey();
        }
```

Kod: **https://pastebin.com/a5rPNvVb**

‹ POWRÓT DO ZADANIA

Rozwiązanie - 5

Cały „myk” polega na użyciu pętli iterującej po liczbach 5-cyfrowych (od 10000 do 99999), a następnie porównywanie hashy z podanym (hash naszego pinu) -> *Jeśli program znajdzie taki sam hash, zwróci go w postaci pinu, którego poszukujemy*

```
        static void Main(string[] args)
        {
            string stringKtoryHasuje = "58221";
            Console.WriteLine(HashString(stringKtoryHasuje));


            string znaleziony = "brak";
            string hash =
"13588a9cb59d457179a0b294857359a881d090870479ec48474336d64dc8cd6470bc7543fbf19beceb3d1
bda81081134af152fe76ade1d1251abeca9d1949f7e";

            for(int i = 10000; i <= 99999; i++)
            {
                if(HashString(i.ToString()) == hash)
                {
                    znaleziony = i.ToString();
                    break;
                }
            }

            Console.WriteLine($"Znaleziony numer to: {znaleziony}");
            Console.ReadKey();
        }
```

Kod: **https://pastebin.com/1ByANx6U**