# Man openssl enc

Man openssl ciphers

# **1 zjazd**

# Kryptografia – Funkcje hashujące

## Fukcja skrótu MD5:

https://www.youtube.com/watch?v=qHH7pHUYEEY&t=3s

Komenda nc 10.0.15.206 1337, kopiujemy później w drugim terminalu **echo -n „wklejamy skopiowane” | md5sum** i dostajemy odpowiedź na zadanie

## Funkcja skrótu SHA1:

https://www.youtube.com/watch?v=GcUuKxWN9Wk

Komenda nc 10.0.15.206 1337, kopiujemy później w drugim terminalu **echo -n „wklejamy skopiowane” | sha1sum** i dostajemy odpowiedź na zadanie

## Funckja skrótu SHA3:

https://www.youtube.com/watch?v=0dfcmc4xQJw

Komenda nc 10.0.15.206 1337, kopiujemy później wchodzimy na stronę <https://md5calc.com/hash/sha3-512> wklejamy nasz string i dostajemy odpowiedź

## Łamanie skrótu MD5 metodą słownikową:

https://www.youtube.com/watch?v=adagCEdB5kI

Nc … dostajemy zakodowany ciąg , ściągamy plik rockyou.txt później : echo „zakodowany znak” > pass.txt następnie komenda **john –wordlist=rockyou.txt format=raw-md5 pass.txt.** Znów nc i podajemy hasło które otrzymaliśmy.

## Identyfikacja funkcji skrótu:

Łączymy się przez nc . odpalamy python 3. Odpalamy hashliub bibliotekę i do zmiennych podajemy wartości które dostaliśmy w terminalu”

Obraz zawierający tekst

Opis wygenerowany automatycznie

## Kod uwierzytalniący HMAC:

https://www.youtube.com/watch?v=syv4bvI73Bo

Nc… dalej **echo – n „podajemy wiadomość” | openssl dgst -sha512 -hmac „klucz wiadomości”** i ctrl + d

## Kolizja MD5:

https://www.youtube.com/watch?v=sRWcs0Z-Qzg&t=11s

Wchodzi w neta szukamy strony z md5 collision kopiujemy oba wrzucamy do pliku zapisujemy i łączymy się podając an parametr wejścia plik np. **nc 10.0.18.102 1337 < odp.txt**   
Plik w środku wygląda tak: Obraz 3

## TEST SPRAWDZAJĄCY!:

**2 zjazd**

# Kryptografia – Szyfrowanie symetryczne

## Odszyfrowanie danych szyfrem AES

https://www.youtube.com/watch?v=OEswuYvywdc&t=6s

Echo „tekst zaszyfrowanej wiadomości” > data.enc

Openssl enc -aes-256-ecv -d -a -in data.enc -K (i teraz podajemy klucz)

## Odszyfrowanie pliku szyfrem AES

https://www.youtube.com/watch?v=VBcAhNLoP9c&t=1s

Śćiagamy oba pliki

Cat cs.out

Cat pwd.pass -> wyświetla nam hasło które jest użyte do wytworzenia klucza

Komenda :

Openssl enc -aes-256-cbc -d -in cs.out -kfile pwd.pass -pbldf2

## Szyfrowanie danych algorytmem 3des

https://www.youtube.com/watch?v=EgFyOJsyk8w&t=3s

Tworzymy plik data.enc i wklejamy do niego zaszyfrowaną wiadomość

Openssl enc -d -a -in data.enc -k „podajemy hasło które dostaliśmy” -pbkdf2 -des-ede3

## Szyfrowanie z wykorzystaniem algorytmu PBKDF1

https://www.youtube.com/watch?v=xJq2fraUZDQ&t=2s

Tworzymy plik data.enc i wklejamy do niego zaszyfrowaną wiadomość która dostaliśmy

Openssl enc -d -aes-256-ecb -k „podajemy hasło” -in data.enc -a -ter „podajemy liczbę iteracji która dostaliśmy w zadaniu”

## Algorytm PBKDF2 z niestandardową liczbą iteracji

https://www.youtube.com/watch?v=ZyQQRmgsZ5U

Echo „zaszyfrowana wiadomość która dostajemy z zadania” > data.enc

Cat data.enc

Openssl enc -d -a -aes-256-cbc -in data.enc -k „podajemy hasło które dostaliśmy” -pbkdf2 -iter „suma iteracji jaką otrzymaliśmy w zadaniu”

## Identyfikacja algorytmu szyfrującego

https://www.youtube.com/watch?v=q\_EbNioPnn4&t=15s

Nc. Tworzymy plik wrzucamy w niego zaszyfrowaną wiadomość np. data.enc

Tworzymy plik i wrzucam hasło do niego np. pwd.pass

Openssl enc -ciphers kopiujemy wszystko z koncówką ecb i wrzucamy do jednego pliku nazywamy go np. ciphers.txt

Tworzymy skrypt w bashu przeszukujący wszystkie algorytmy:

While read cipher;

Do

Echo „---------‘’

Echo $cipher

Openssl enc -d -a $cipher -in data.enc -kfile pwd.pass -pbkdf2

Done;

## Szyfrowany plik zip

https://www.youtube.com/watch?v=SOALDU1Kgnw&t=3s

Ściągamy plik

zip2john 2a34fsdf34… czyli-> (nazwaplikuściągnietego) > john.pass

john –-wordlist=rockyou.txt john.pass (na pomarańczowo dostajemy hasłdo do pliku)

unzip nazwapliku.zip

cat secret.txt (plik txt po rozkaowaniu)

**3 Zjazd:**

**Szyfrowanie symetryczne - generowanie kluczy RSA**

**https://www.youtube.com/watch?v=18q1WOcyzWM&t=6s**

openssl genres -out priv.pem 4096

Wyświetlamy klucz: cat priv.pem

openssl rsa -in priv.pem -pbout -out pub.pem

Wyświetlamy klucz publiczny: cat pub.pem

Otwieramy gedit, wklejamy klucz publiczny, dajemy linię odstępu i wklejamy klucz prywatny. Zapisujemy np jako keys.pem

Wyświetlamy cat keys.pem

Kopiujemy całość i wklejamy jako odpowiedź do zadania

**3.2 Generowanie kluczy na krzywych eliptycznych**

<https://www.youtube.com/watch?v=JpFwTpIubBo&t=1s>

openssl ecparam -name prime256v1 -genkey -out priv1.pem

openssl ec -in priv1.pem -out pub1.pem -pubout

Kopiujemy do 1 pliku i wklejamy jako odpowiedź.

**3.3 Szyfrowanie kluczem publicznym**

<https://www.youtube.com/watch?v=bR74jBpT2G0>

Kopiujemy klucz publiczny i zapisujemy go w pliku np pub3.pem

Kopiujemy tekst do zakodowania i zapisujemy go w pliku np data.txt

openssl pkeyutl -encrypt -in data.txt -inkey pub3.pem -pubin -out data.enc -pkeyopt rsa\_padding\_mode:oaep

Następnie: base64 -w0 data.enc

**3.4 Certyfikat z zadanym polem Nazwa pospolita**

<https://www.youtube.com/watch?v=TaLfj2O4VAE&t=1s>

Tworzymy katalog ca (mkdir ca) i wchodzimy do niego

openssl genrsa -out ca.key 2048

openssl req -new -x509 -key ca.key -out ca.crt

Uzupełniamy dane byle jak i w miejscu Common Name wklejamy tekst z zadania

openssl x509 -in ca.crt -text

Kopiujemy certyfikat i wklejamy jako odpowiedź

**3.5 Certyfikat na podstawie zapytania**

<https://www.youtube.com/watch?v=CCimhx4Qe8o>

Jeżeli nie ma:

Tworzymy katalog ca, wchodzimy

openssl genrsa -out ca.key 2048

openssl req -new -x509 -key ca.key -out ca.crt

Uzupełniamy dane byle jak i w miejscu Common Name wklejamy tekst z zadania (poprzedniego)

Tworzymy client.csr i wklejamy tam certyfikat z zadania

openssl x509 -req -CAkey ca.key -CA ca.crt -in client.csr -CAcreateserial -out client.crt

Cat client.crt

Kopiujemy i wklejamy jako odpowiedź.

**3.6. Generowanie klucza PGP**

<https://www.youtube.com/watch?v=77PLPCd250g&t=2s>

gpg --gen-key

Jako imię i nazwisko wklejamy tekst z zadania

Dowolny adres email

Wybieramy dalej

gpg --export --armor „tekst z zadania”

Kopiujemy i wklejamy jako odpowiedź

**3.7. Szyfrowanie za pomocą klucza PGP**

<https://www.youtube.com/watch?v=7pyJb6hy1eo>

Wklejamy klucz z zadania do pub.pem

gpg --show-key pub.pem

gpg --import pub.pem

echo -n „text z adania| gpg --encrypt --armor --recipient contact@cyberskiller.com

Kopiujemy i wklejamy jako odpowiedź

**3.8 Odszyfrowanie za pomocą klucza PGP**

<https://www.youtube.com/watch?v=0oxzn0b75KY&t=3s>

gpg --gen-key

Wpisujemy dowolne dane

Email np test@cyberskiller.com

Ustawiamy hasło

gpg --export --armor [test@cyberskiller.com](mailto:test@cyberskiller.com)

Jako odpowiedź wklejamy wygenerowany klucz publiczny

Dostajemy PGP MESSAGE

Wklejamy ją do data.enc

Wpisujemy:

gpg --decrypt data.enc

Dostajemy odpowiedź do zadania