

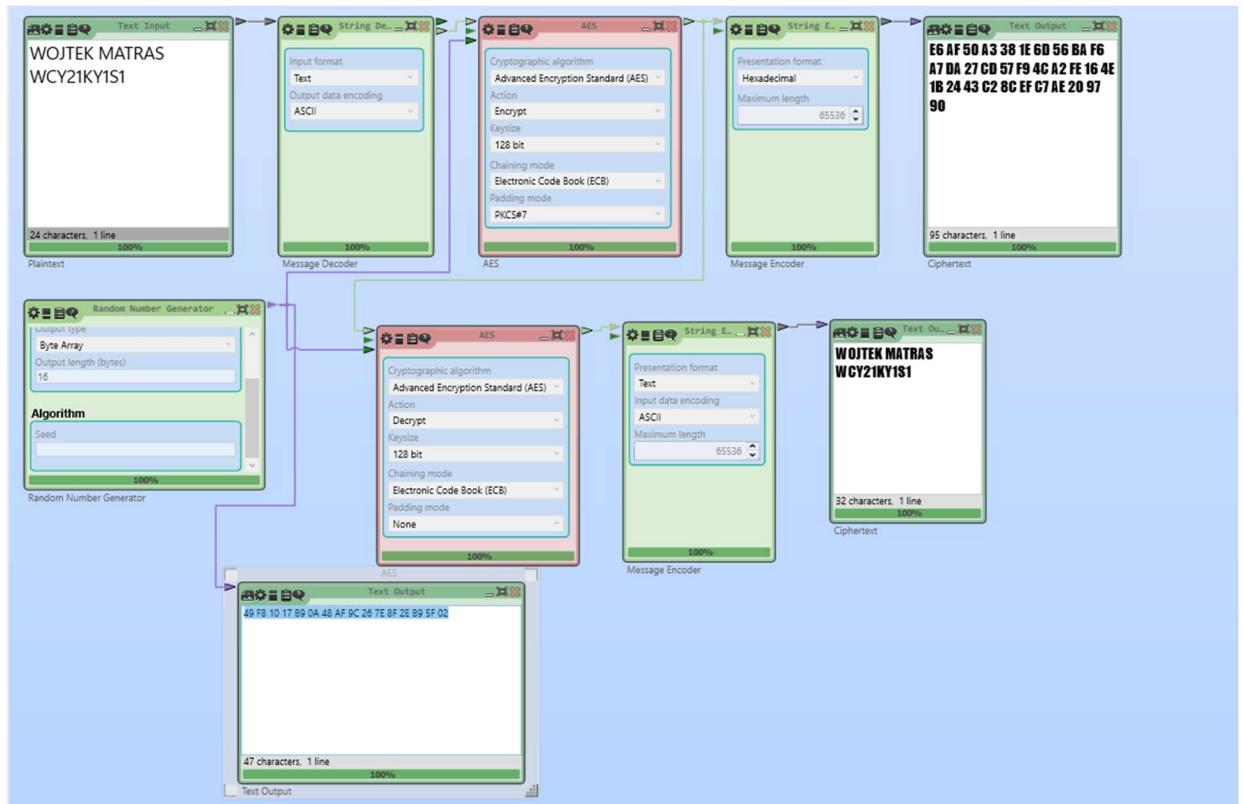
Opracuj model realizujący szyfrowanie i deszyfrowanie wybranym (dowolnym) kluczem o wszystkich trzech długościach (128, 192 i 256 bitów) szyfrem blokowym (standardem) AES w trybie ECB tekstu jawnego zadanej na dwa sposoby: tekstowo (Text Input) i jako plik (File Input). Następnie używając jako klucza losowego ciągu odpowiedniej długości (uzyskanego przy pomocy bloku „Random Number Generator”) zaszyfruj i odszyfruj:

- w trybie szyfrowania tekstu swoje imię i nazwisko trzema różnymi długościami klucza,

$16*8=128$  klucz 49 F8 10 17 B9 0A 48 AF 9C 26 7E 8F 2E B9 5F 02

Szyfrowany tekst: WOJTEK MATRAS WCY21KY1S1

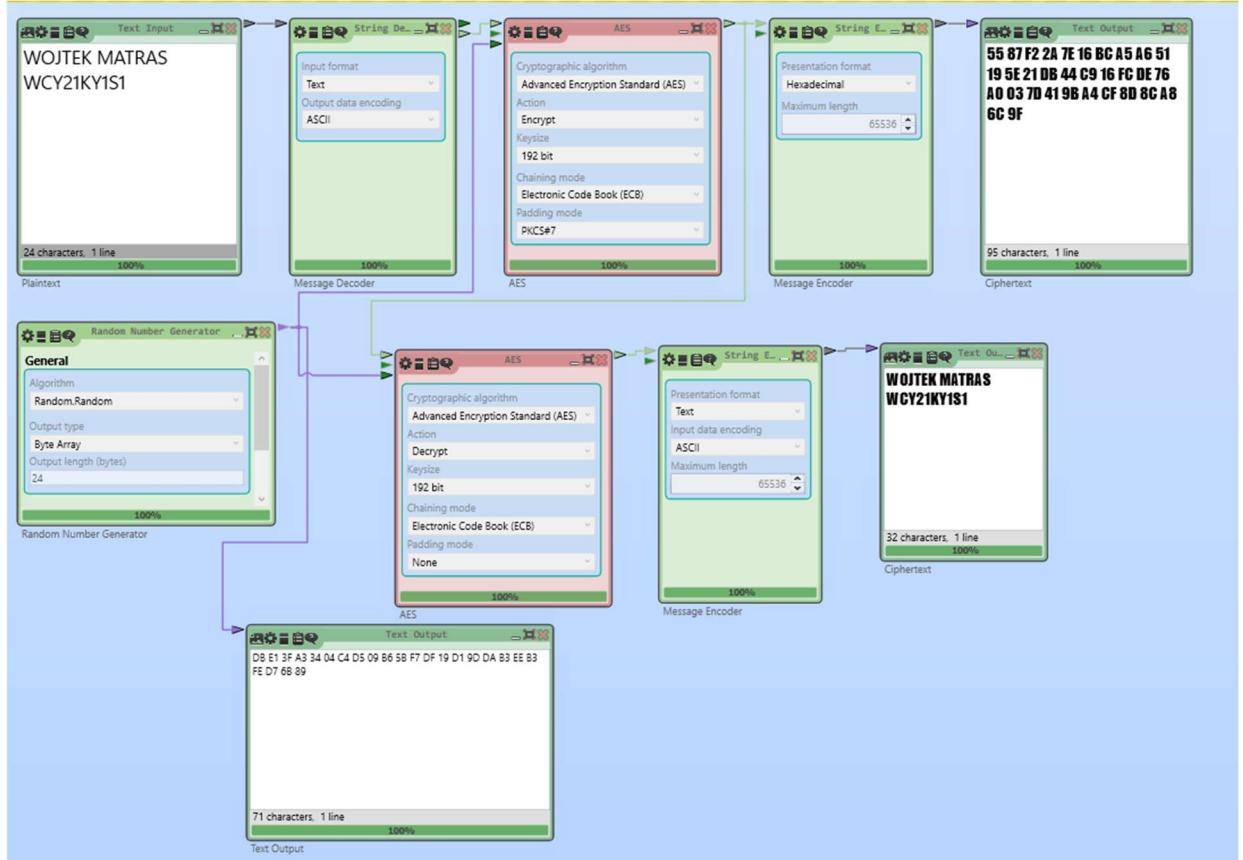
Zaszyfrowany tekst widać na screenie byte array z 95 znakami, w prawym górnym rogu



$24*8=192$  klucz DB E1 3F A3 34 04 C4 D5 09 B6 5B F7 DF 19 D1 9D DA B3 EE B3 FE D7 6B 89

Szyfrowany tekst: WOJTEK MATRAS WCY21KY1S1

Zaszyfrowany tekst widać na screenie byte array z 95 znakami, w prawym górnym rogu



$32 \times 8 = 256$ bitowy klucz : 3E 17 BA 96 AE 04 CD 3B 99 86 9E 56 F0 AD BF 3A 6F  
B7 4D 25 55 17 5D CC 6E 8B 09 14 57 9A B0 36

Szyfrowany tekst: WOJTEK MATRAS WCY21KY1S1

Zaszyfrowany tekst: 57 4F 4A 54 45 4B 20 4D 41 54 52 41 53 20 57 43 59 32 31 4B  
59 31 53 31 08 08 08 08 08 08 08 08

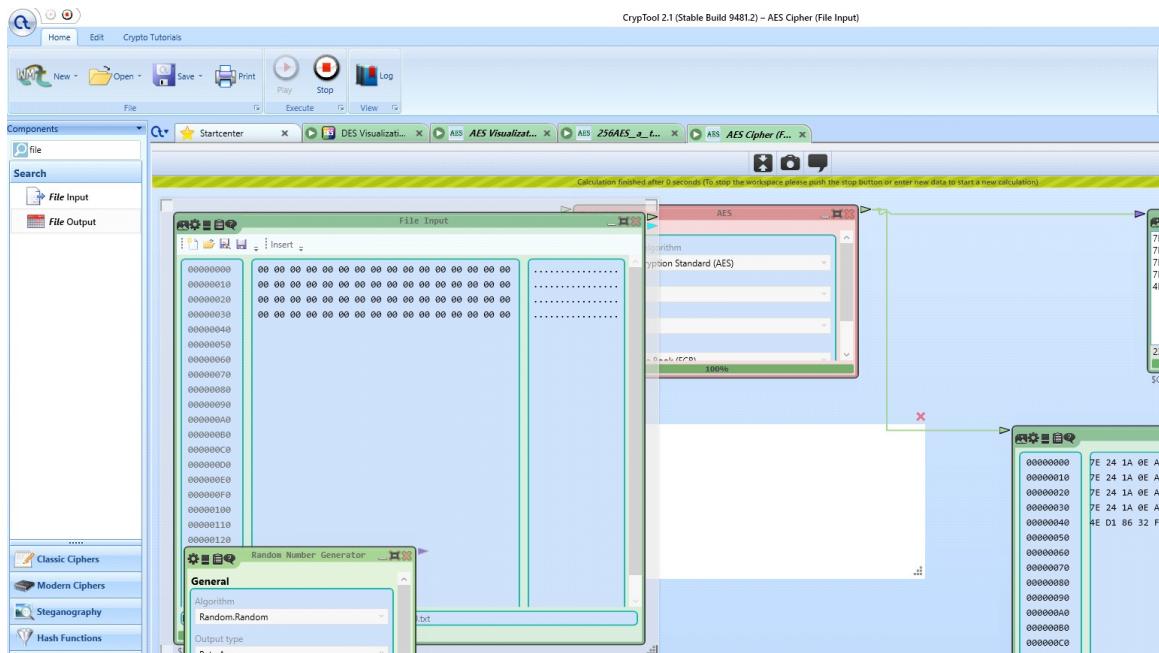


- b) w trybie szyfrowania plików załączone pliki: plain00.txt, plainFF.txt, plainCTR.txt, plainLW.txt, plainBW.txt. Blok FileInputStream powiększ, tak aby na zrzucie ekranu widać było całą zawartość każdego pliku. Omów krótko jakie własności mają te pliki, a jakie

ich szyfrogramy.

klucz 128 BITOWY (Random Number Generator)

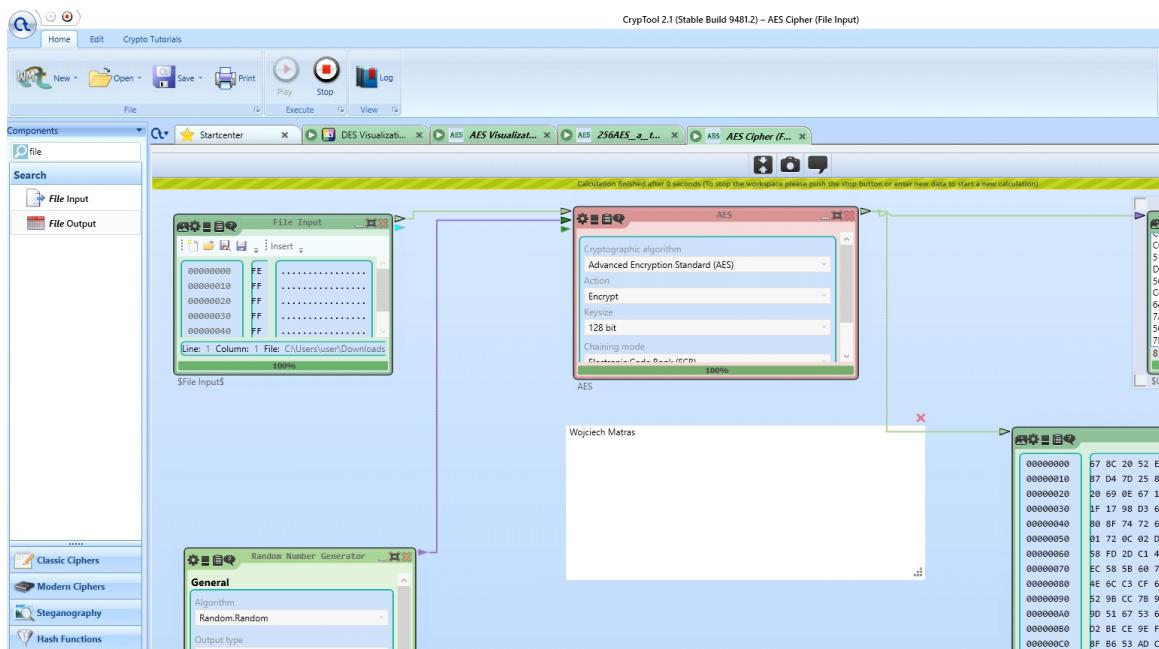
plain00:



text wyjściowy:

7E 24 1A 0E A8 70 1B DE 71 48 33 97 E5 93 F1 08 7E 24 1A 0E A8 70 1B DE 71 48 33 97 E5 93 F1 08 4E D1 86 32 F2 68 93 11 7E 71 42 10 E5 08 55 B8

plainBW.txt

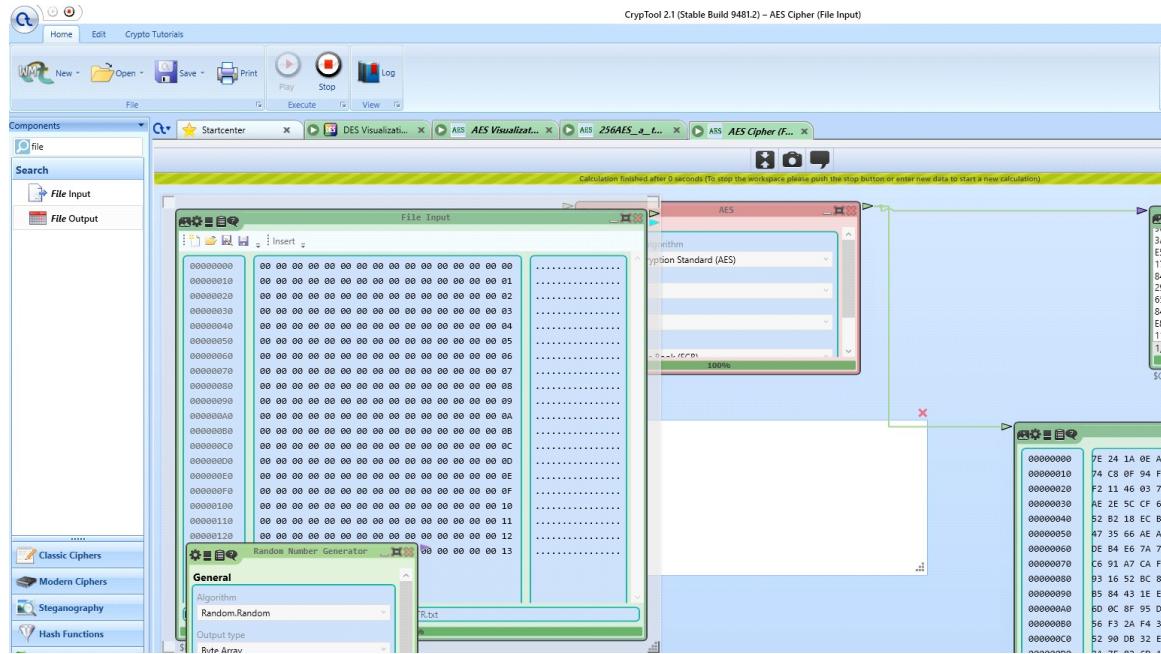


text wyjściowy:

67 8C 20 52 ED E8 13 19 71 56 58 BD D2 17 DD 54 87 D4 7D 25 80 79 83 22 0C BC 8E 28 A8 E6 97 50 20 69 0E 67 1B 5A B5 19 6B 5C 3F 09 72 47 E6 B4 1F 17 98 D3 64 74 41 C0 32 6C 40 E5 C7 2C 7E 56 80 8F 74 72 67 B5 D1 18 85 DF 6D 96 E3 3D F7 D9 01 72 0C 02 D5 4D DE D8 11 EB 89 7A 10 51 DB 7D 58 FD 2D C1 4A C5 76 4B 49 02 BD 60 5F B9 A7 F9 EC 58 5B 60 7D 99 99 C3 0D 2B 01 DF 93 E8 B2 6E 4E 6C C3 CF 6D 68 AE 95 16 FF 8D 98 4F 8A 25 55 52 9B CC 7B 9E 82 F9 F6 10 D6 E4 DB A2

E7 F1 69 9D 51 67 53 6F 88 8C 6D 64 B4 C3 8F 1E C2 D9 F4 D2 BE CE 9E FD 31 EC D8 1B 95 A9 E1 4A  
3F E7 5C 8F B6 53 AD C5 24 65 8F C5 B7 80 50 12 F6 C4 0A 79 CC D0 38 49 E7 B7 C4 82 9D D9 18 64  
03 D7 E9 92 48 DC A8 EA 60 B7 59 BD 13 7A AB 99 DC CF BE 42 63 79 B2 E5 35 B4 94 69 5C 24 15  
4B CC 4B 08 4E D1 86 32 F2 68 93 11 7E 71 42 10 E5 08 55 B8

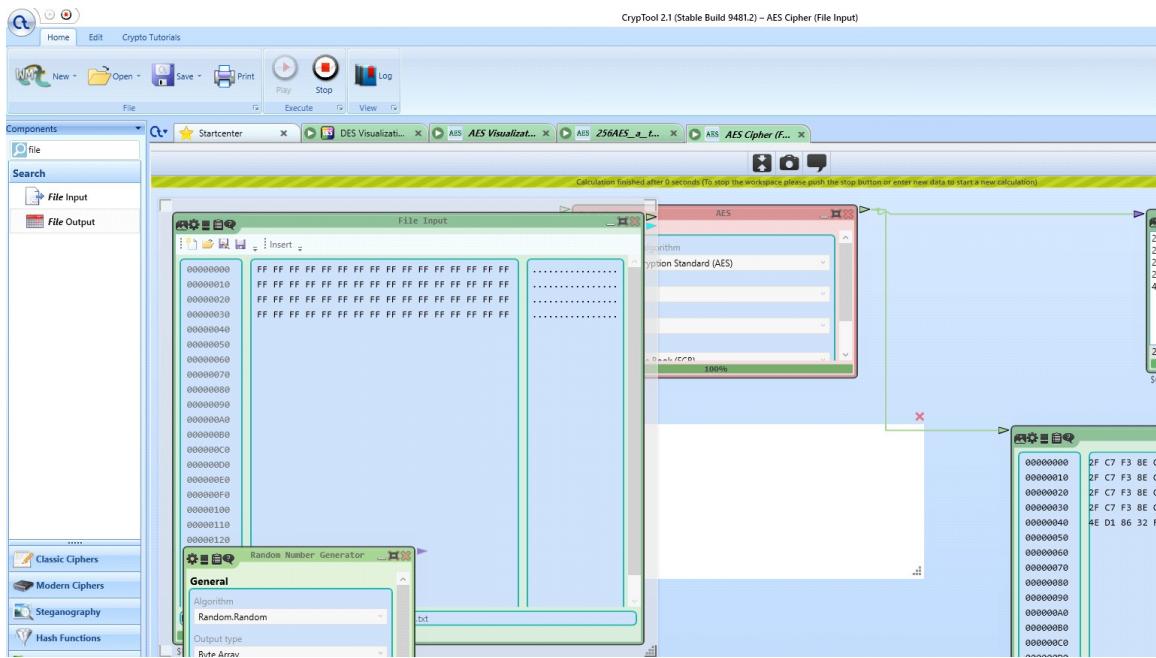
plainCTR:



text wyjsciowy:

7E 24 1A 0E A8 70 1B DE 71 48 33 97 E5 93 F1 08 74 C8 0F 94 F2 C6 48 A6 3D 08 EB 53 F1 8C D3 F3  
F2 11 46 03 77 D9 F3 BE 7E 11 96 97 0E 05 2D D6 AE 2E 5C CF 60 68 32 AF DE F9 AA 1F 35 F1 E0 41  
52 B2 18 EC B0 FB CF B8 CA E8 16 6B C0 CF 06 E5 47 35 66 AE A7 CE A7 2A 13 D2 B4 23 AF 4E 1E D9  
DE B4 E6 7A 75 44 64 0F 62 49 03 4F C9 E5 C2 BD C6 91 A7 CA FE 00 5B 72 31 7C 56 73 67 E2 07 DD  
93 16 52 BC 8D 8F BA 08 39 F1 C0 38 EA F9 C8 B3 B5 84 43 1E E7 14 04 43 13 1D 83 77 15 2A 7C C8  
6D 0C 8F 95 D4 48 2E 03 BD 74 4C DA 34 78 74 7E 56 F3 2A F4 36 2A FE D6 E9 A0 81 33 85 B9 FB F1  
52 90 DB 32 E4 89 9E 76 8E F4 2A 37 75 95 CC 74 3A 7F 82 6D A9 E7 B2 9B 10 09 FF D9 DF 15 A4 E5  
2E CC A0 78 DE 0F 32 4F 99 E7 69 EB 09 85 17 6B 64 69 FC 2C 17 E2 C2 22 16 93 9A 52 A5 84 BD 49  
22 CC 91 BD 23 1D 4B 9F 50 AB 5A 29 2F 85 57 02 B0 B3 3C B1 03 97 16 16 DF AC 65 BF 74 B4 C5  
A2 4F CB 49 92 FD 7D 63 6F 9B 84 C8 28 BE 3D 08 67 AC 99 A6 B3 4B 37 63 B3 E8 91 AF B0 D2 FC 8F  
98 4E D1 86 32 F2 68 93 11 7E 71 42 10 E5 08 55 B8

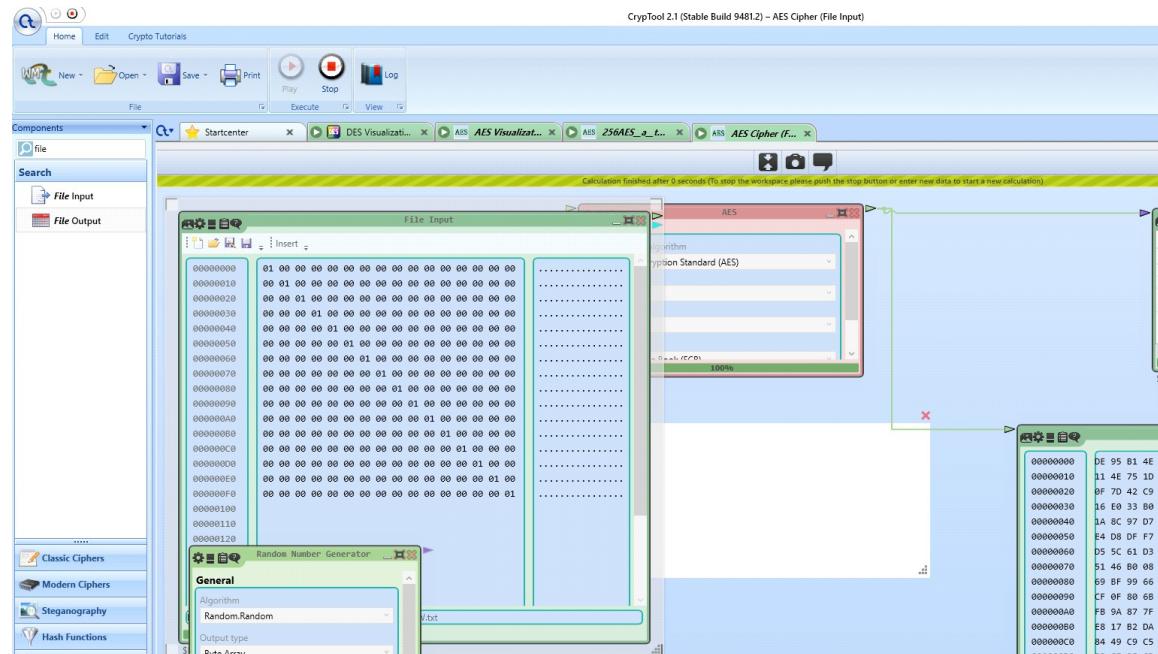
plainFF.txt



text wyjściowy:

```
2F C7 F3 8E C4 F3 0E 10 20 36 5C D0 CA 7A B7 59 2F C7 F3 8E C4 F3 0E 10 20 36 5C D0 CA 7A B7 59
2F C7 F3 8E C4 F3 0E 10 20 36 5C D0 CA 7A B7 59 2F C7 F3 8E C4 F3 0E 10 20 36 5C D0 CA 7A B7 59
4E D1 86 32 F2 68 93 11 7E 71 42 10 E5 08 55 B8
```

plaintLW:



text wyjściowy:

```
DE 95 B1 4E 7B 3C CF 40 6A C7 4E 59 B8 98 12 99 11 4E 75 1D DC 9F 0C E8 D8 B5 E4 71 C6 30 A2 CD
0F 7D 42 C9 F0 CB EC BF E2 CE 2B 64 29 D0 2D 1E 16 E0 33 B0 1E 00 35 57 16 88 10 2B 93 EA D3 C5
1A 8C 97 D7 9B 1C A6 8E D6 1D 3F E2 15 EE CE 56 E4 D8 DF F7 53 84 AD 9C 3D D7 29 EC EB 59 EB
64 D5 5C 61 D3 A2 F8 E7 78 A6 E6 F4 62 12 31 60 6E 51 46 B0 08 07 DB 54 72 07 D3 94 BF F4 CF A9
BB 69 BF 99 66 A1 71 08 E8 CA 22 49 01 CE F1 57 AC CF OF 80 6B 2F 76 8F 22 82 AA 1C AA 88 19 01
B6 FB 9A 87 7F 98 8A B5 F6 CC DC B2 A0 D8 23 FB 4F E8 17 B2 DA CC 4D 18 D7 09 0B 40 69 9F 55
3B 8D 84 49 C9 C5 2A 74 78 53 D2 F3 CF 58 28 70 43 B6 B3 CE 9E 6D 3E 0A 80 50 3F 68 9B 5F E8 95
1D CE 6E 79 24 71 EF E1 1E 4F F3 84 37 6F D4 48 6C 75 74 C8 0F 94 F2 C6 48 A6 3D 08 EB 53 F1 8C
```

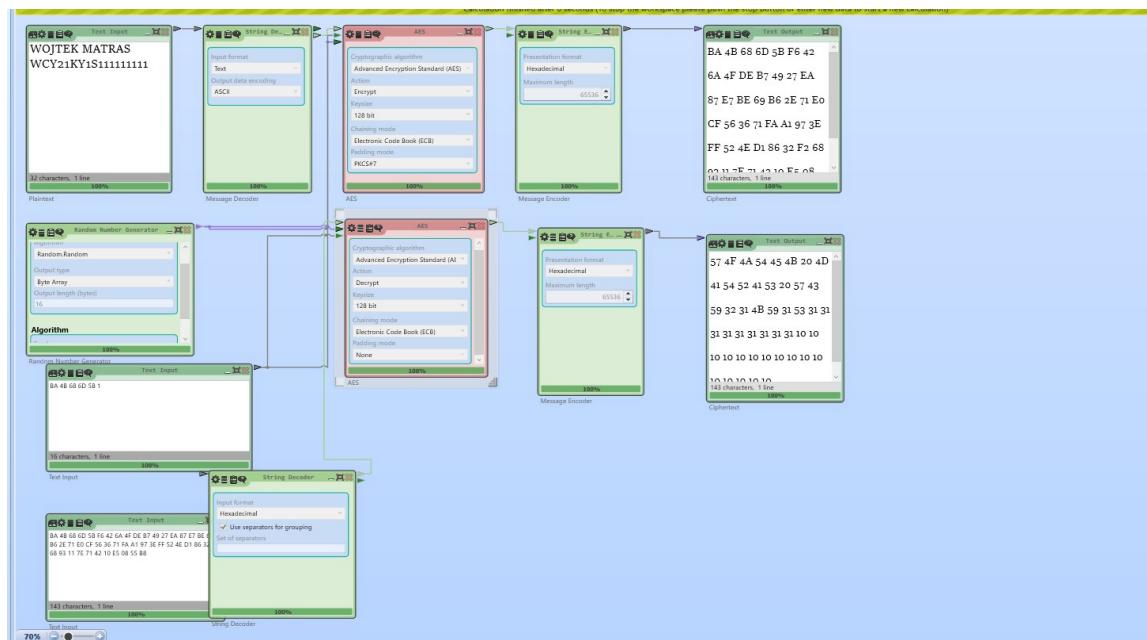
D3 F3 4E D1 86 32 F2 68 93 11 7E 71 42 10 E5 08 55 B8

pliki wejściowe i wyjściowe całkowicie od siebie się różnią. Mimo kodowania można zauważyc powtarzające się bloki przy tym samym tekscie wejściowym. Jednak gdy wiersz różni od siebie się minimalnie, wyjściowy tekst jest całkowicie różniacy się od siebie.

Metody dopełniania wiadomości:

Korzystając z funkcji skrótu SHA-2 albo SHA-3 z dowolną długością skrótu opracować model funkcji skrótu z kluczem (MAC) na bazie funkcji skrótu (HMAC) zgodnie ze standardem FIPS 198-1. Przy użyciu tego modelu wyznaczyć skróty wiadomości utworzonej z własnego imienia i nazwiska stosując losowy klucz odpowiedniej długości.

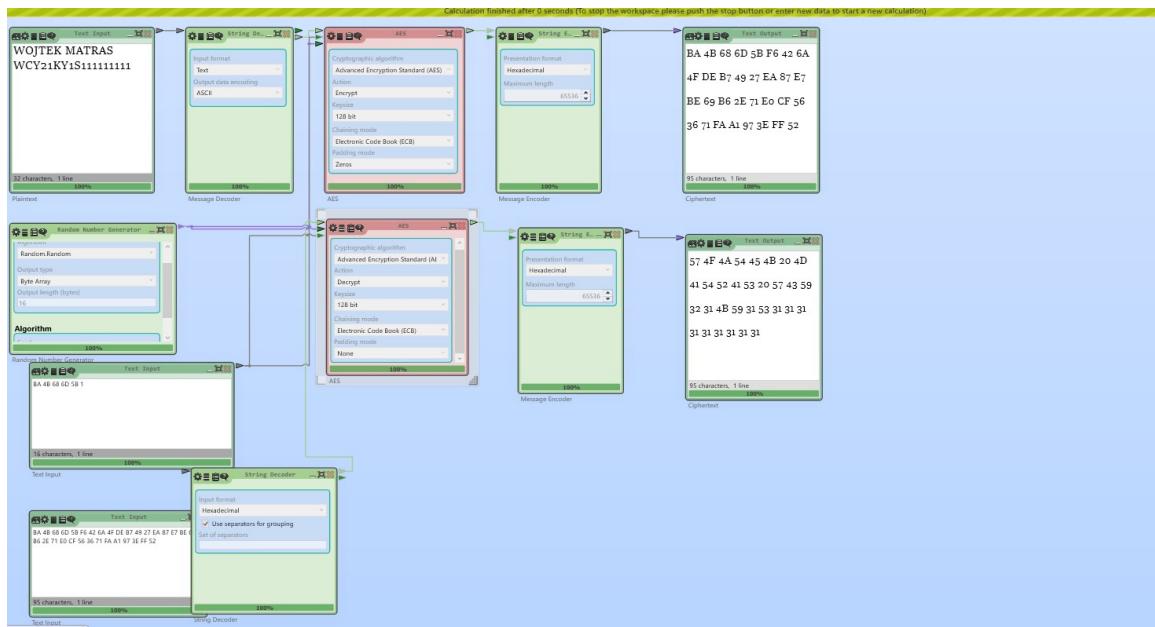
padding z PKCS7 do NONE



57 4F 4A 54 45 4B 20 4D 41 54 52 41 53 20 57 43 59 32 31 4B 59 31 53 31 31 31 31 31 31 31 31 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10

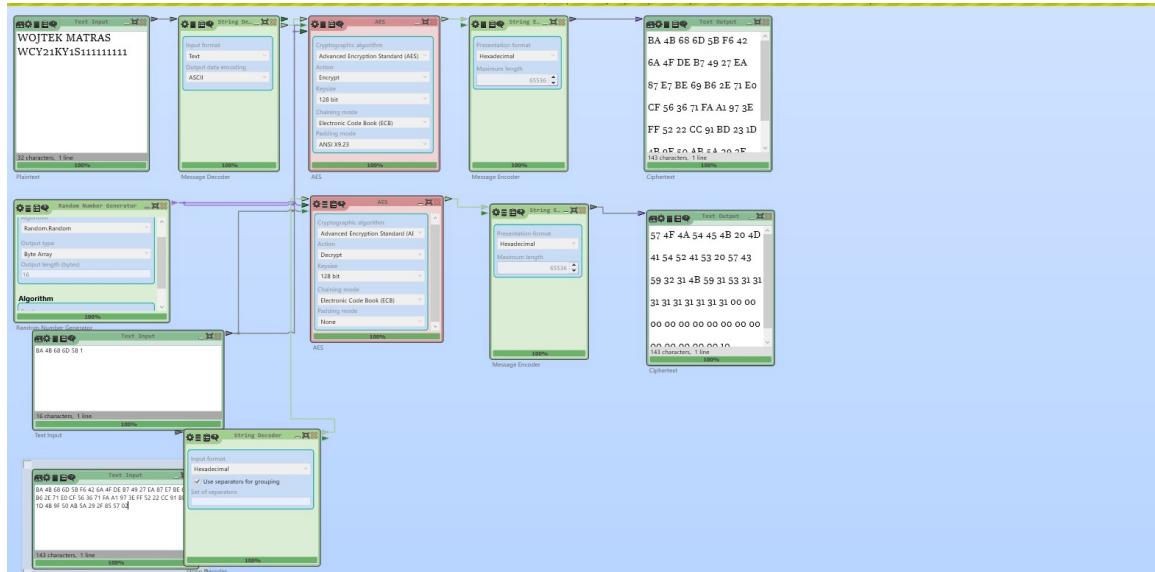
(dodatkowe 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10)

padding z Zeros na None



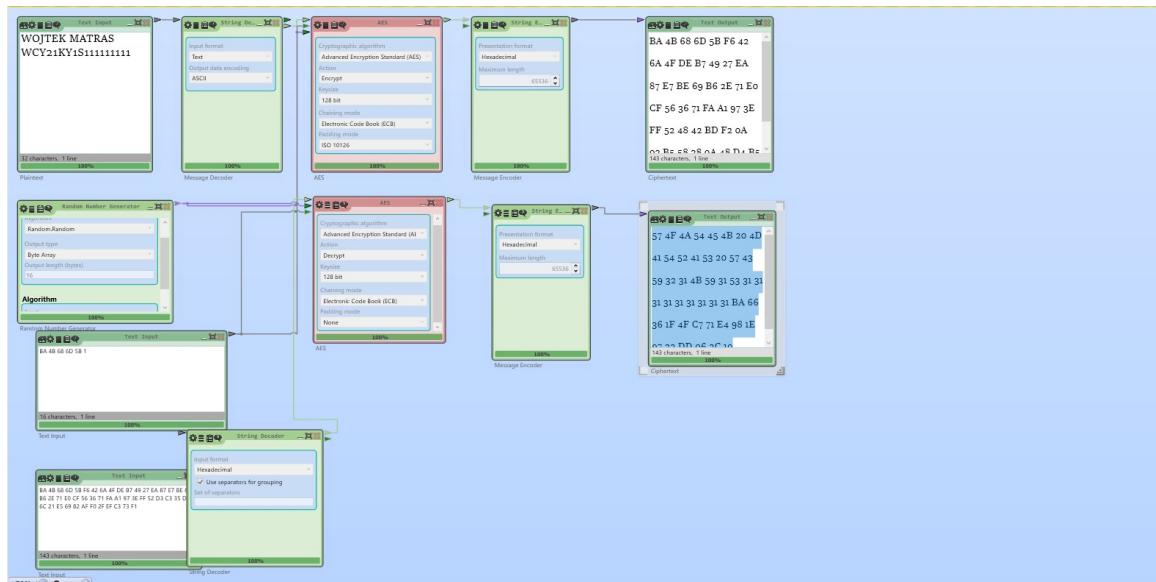
57 4F 4A 54 45 4B 20 4D 41 54 52 41 53 20 57 43 59 32 31 4B 59 31 53 31 31 31 31 31 31 31 31 31 31  
(brak różnicy)

padding z ANSI X9 na None



57 4F 4A 54 45 4B 20 4D 41 54 52 41 53 20 57 43 59 32 31 4B 59 31 53 31 31 31 31 31 31 31 00  
00 10 (dodatkowe 00 00 00 00 00 00 00 00 00 00 00 00  
00 10 )

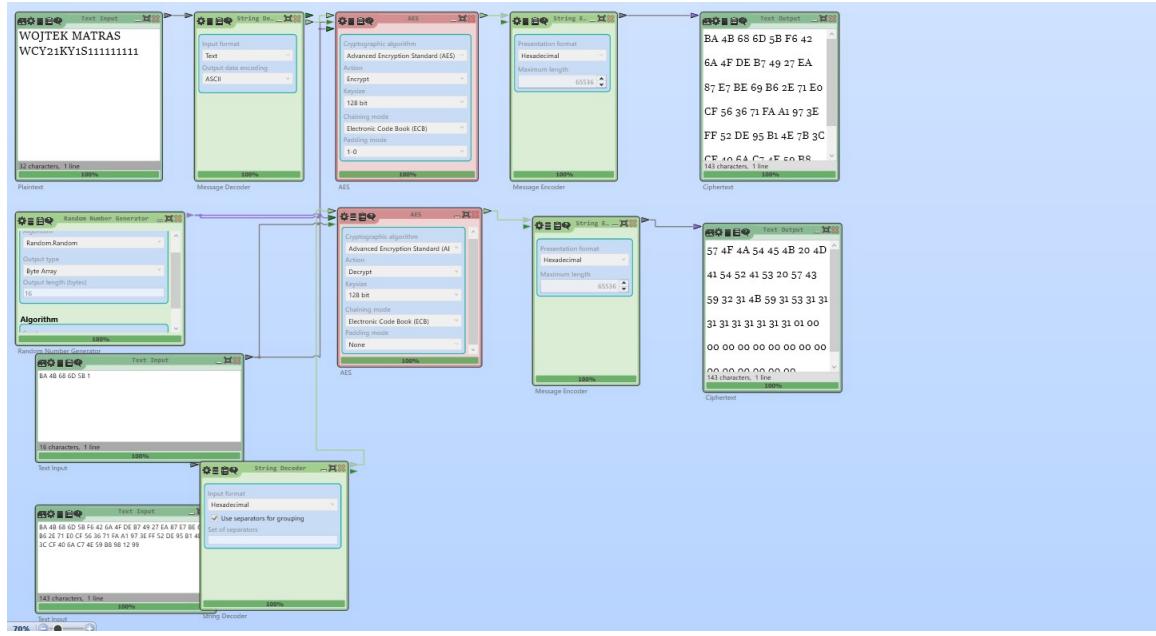
Padding z ISO na none



57 4F 4A 54 45 4B 20 4D 41 54 52 41 53 20 57 43 59 32 31 4B 59 31 53 31 31 31 31 31 31 31 31 BA  
66 36 1F 4F C7 71 E4 98 1E 97 32 DD 06 2C 10

( dodatkowe BA 66 36 1F 4F C7 71 E4 98 1E 97 32 DD 06 2C 10)

padding z 1-0 do None



57 4F 4A 54 45 4B 20 4D 41 54 52 41 53 20 57 43 59 32 31 4B 59 31 53 31 31 31 31 31 31 31 01  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Metody dopełnienia AES, takie jak none, PKCS7, Zeros, ANSI X9, ISO/IEC 7816-4, "ISO/IEC 10126" oraz "ISO/IEC 9797-1 method 2", są używane do wyrównywania bloków danych przed szyfrowaniem lub po deszyfrowaniu w trybie blokowym AES

Metody dopełnienia AES, takie jak "none", "PKCS7", "Zeros", "ANSI X9.23", "ISO/IEC 7816-4", "ISO/IEC 10126" oraz "ISO/IEC 9797-1 method 2", są używane do wyrównywania bloków danych przed szyfrowaniem lub po deszyfrowaniu w trybie blokowym AES (Advanced Encryption Standard). Oto krótki opis każdej z tych metod dopełnienia:

## 1. Metoda "None":

Ta metoda oznacza brak dopełnienia. Oznacza to, że dane muszą mieć dokładnie taki sam rozmiar jak rozmiar bloku AES. Jeśli dane nie są dostatecznie długie, muszą zostać uzupełnione na odpowiednią długość przed szyfrowaniem. Ta metoda nie dodaje żadnych dodatkowych bitów ani bajtów.

## 2. Metoda "PKCS7":

Jest to popularna metoda dopełnienia stosowana w trybach blokowych. W przypadku metody PKCS7, dane są dopełniane poprzez dodanie bajtów o wartości równej liczbie bajtów, które trzeba dodać, aby uzupełnić blok do pełnej długości. Na przykład, jeśli brakuje 4 bajtów do pełnego bloku 16 bajtowego, zostaną dodane 4 bajty o wartości 0x04.

## 3. Metoda "Zeros":

Metoda ta polega na uzupełnianiu brakujących bajtów zerami. Jeśli blok danych jest krótszy niż rozmiar bloku AES, brakujące bajty są uzupełniane zerami.

## 4. Metoda "ANSI X9.23":

Jest to metoda dopełnienia, która polega na dodawaniu bajtów o wartości 0x00, po których następuje bajt o wartości 0x80. Pozostałe bajty są dopełniane zerami. Jest to forma dopełnienia stosowana w niektórych starszych systemach.

## 5. Metoda "ISO/IEC 7816-4":

Ta metoda dopełnienia jest używana w kartach inteligentnych (smart card) zgodnych z normą ISO/IEC 7816-4. Oznacza to dodawanie jednego bajtu o wartości 0x80, a resztę bajtów uzupełnia się zerami.

## 6. Metoda "ISO/IEC 10126":

Ta metoda dopełnienia polega na losowym uzupełnianiu bajtów danych, z wyjątkiem ostatniego bajtu, który określa ilość uzupełnienia. Ostatni bajt określa wartość dopełnienia dla pozostałych bajtów.

## 7. Metoda "ISO/IEC 9797-1 method 2":

Jest to metoda dopełnienia, która polega na dodawaniu jednego bajtu o wartości 0x80, a pozostałe bajty są uzupełniane zerami.

## Podstawowe tryby pracy i ich własności

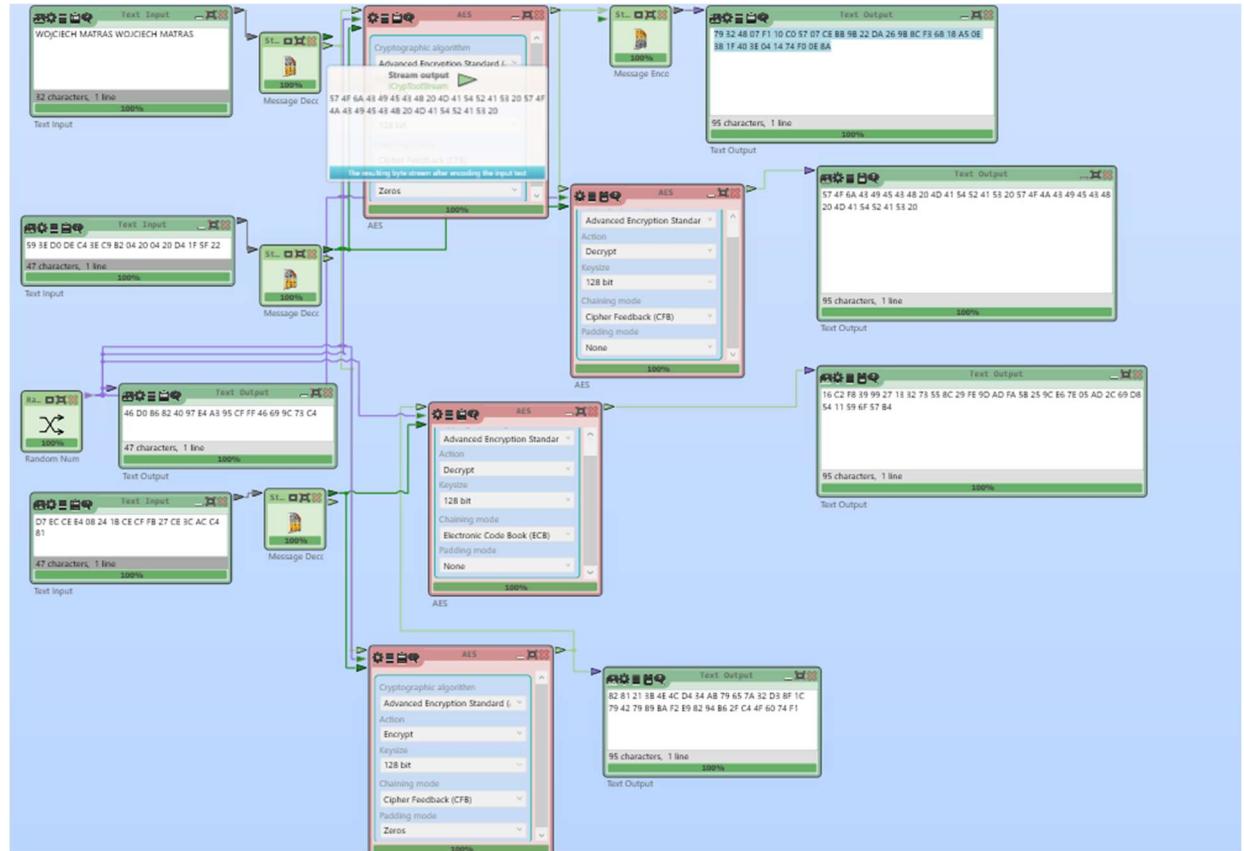
Opracuj model szyfrowania bloku danych (zapisanych szesnastkowo) algorytmem AES w trybach ECB, CBC, OFB, CFB (blok o rozmiarze 512 bitów powinien być szyfrowany równolegle - w jednym kroku), (do utworzenia wartości IV wykorzystaj blok „Random

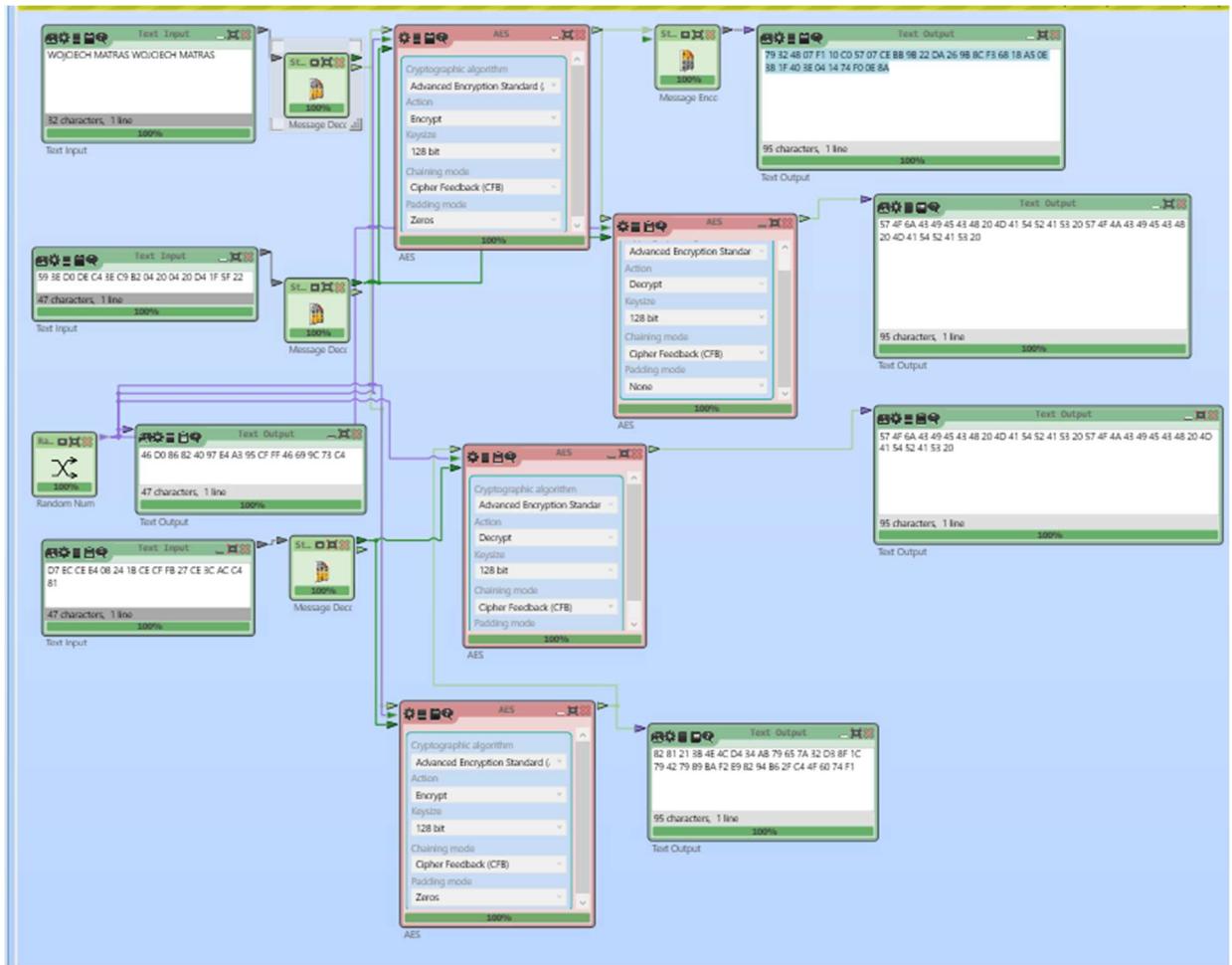
Number Generator”), a następnie zaszyfruj i odszyfruj odpowiednią liczbę bajtów za pomocą każdego trybu.

M: WOJCIECH MATRAS WOJCIECH MATRAS

Wektor: 46 D0 86 82 40 97 E4 A3 95 CF FF 46 69 9C 73 C4

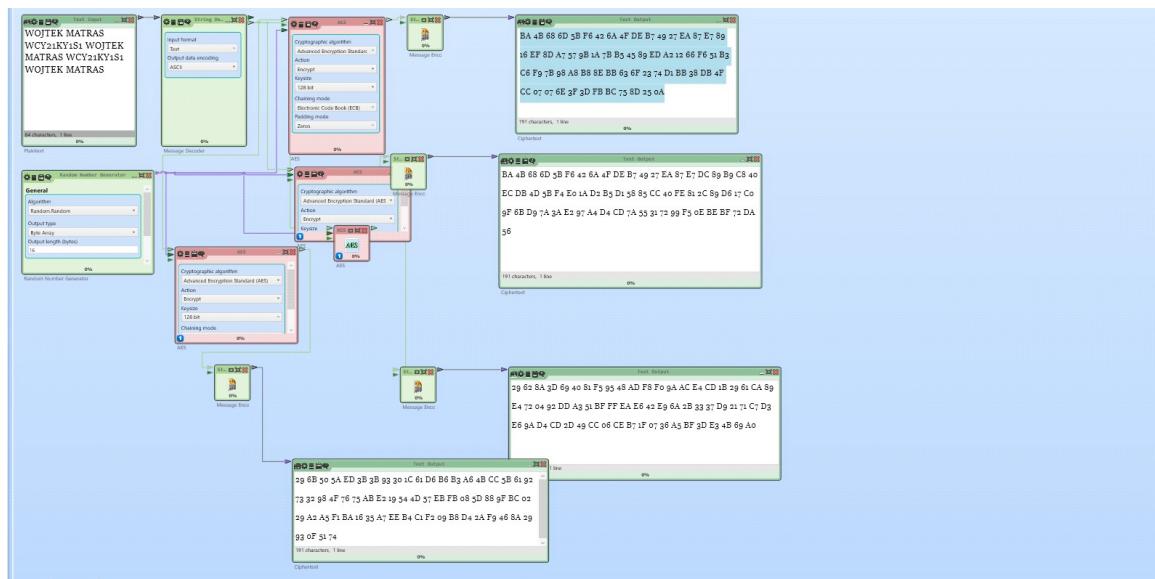
Key: 59 3E D0 DE C4 3E C9 B2 04 20 04 20 D4 1F 5F 22





M: WOJCIECH MATRAS WCY21KY1S1 WOJCIECH MATRAS WCY21KY1S1  
WOJCIECH MATRAS

Wektor: 1A 3D 40 57 6D E8 BD 36 50 49 66 40 6A D5 2B 2F



Blok o rozmiarze 64 znakow czyli 64 bajtów =  $64 \times 8 = 512$  bit

ECB:

BA 4B 68 6D 5B F6 42 6A 4F DE B7 49 27 EA 87 E7 89 16 EF 8D A7 57 9B 1A 7B B5 45 89 ED A2 12 66 F6  
51 B3 C6 F9 7B 98 A8 B8 8E BB 63 6F 23 74 D1 BB 38 DB 4F CC 07 07 6E 3F 3D FB BC 75 8D 25 0A

CBC:

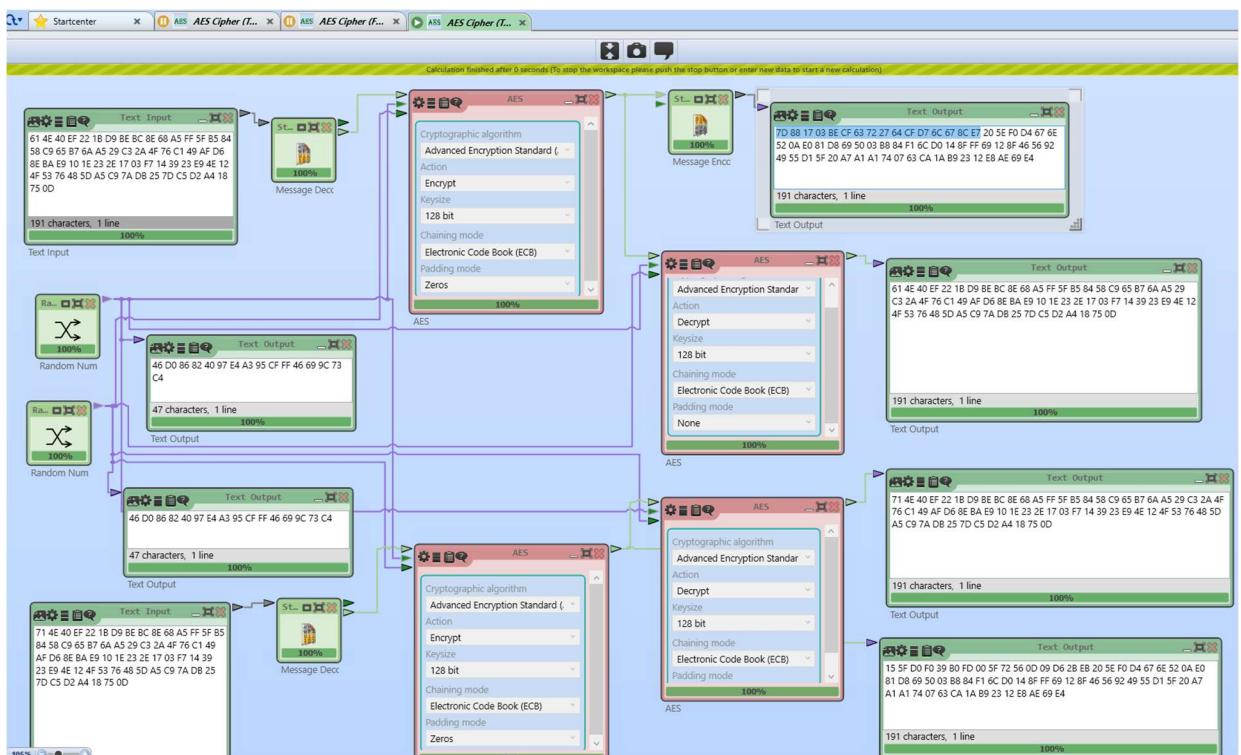
BA 4B 68 6D 5B F6 42 6A 4F DE B7 49 27 EA 87 E7 DC 89 B9 C8 40 EC DB 4D 5B F4 E0 1A D2 B5 D1 58 85  
CC 40 FE 81 2C 89 D6 17 C0 9F 6B D9 7A 3A E2 97 A4 D4 CD 7A 55 31 72 99 F5 0E BE BF 72 DA 56

CFB:

29 62 8A 3D 69 40 81 F5 95 48 AD F8 F0 9A AC E4 CD 1B 29 61 CA 89 E4 72 04 92 DD A3 51 BF FF EA E6  
42 E9 6A 2B 33 37 D9 21 71 C7 D3 E6 9A D4 CD 2D 49 CC 06 CE B7 1F 07 36 A5 BF 3D E3 4B 69 A0

OFB:

29 6B 50 5A ED 3B 3B 93 30 1C 61 D6 B6 B3 A6 4B CC 5B 61 92 73 32 98 4F 76 75 AB E2 19 54 4D 57 EB  
FB 08 5D 88 9F BC 02 29 A2 A5 F1 BA 16 35 A7 EE B4 C1 F2 09 B8 D4 2A F9 46 8A 29 93 0F 51 74



Dla wszystkich czterech opracowanych podstawowych trybów pracy zbadaj wpływ zmian tekście jawnym na szyfrogram. W tym celu zmień w tekście jawnym jeden bit i opisz jak zmieniały się szyfrogramy.

Zmiana 1 bitu: Przechodzę na inne dane ponieważ ciężko zmienić wartość o jeden bit w tekście string.(a raczej to nie możliwe)

Nowe dane:

M: 16 C2 F8 39 99 27 13 32 73 55 8C 29 FE 9D AD FA 5B 25 9C E6 7E 05 AD 2C 69 D8  
54 11 59 6F 57 B4

Zmiana na

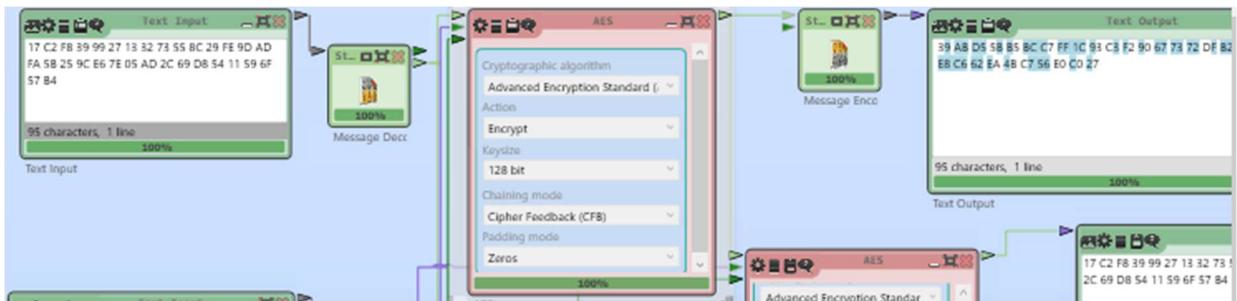
M: 17 C2 F8 39 99 27 13 32 73 55 8C 29 FE 9D AD FA 5B 25 9C E6 7E 05 AD 2C 69 D8  
54 11 59 6F 57 B4

Key: 59 3E D0 DE C4 3E C9 B2 04 20 04 20 D4 1F 5F 22

Wektor: 46 D0 86 82 40 97 E4 A3 95 CF FF 46 69 9C 73 C4

CFB

38 1E 18 52 45 27 8E 53 0E E4 CD 92 05 B4 2A FF D1 76 E0 CE 01 03 5B 01 BD AA 5B C2 E0 09 5B 7B  
 Zmiana 39 AB D5 5B B5 BC C7 FF 1C 93 C3 F2 90 67 73 72 DF B2 3F 8E A1 8B E8 C6 62 EA 4B C7 56 E0 C0 27



ECB:

93 E7 CB 39 41 8B ED 3E F4 45 0C 4A 0A 27 0F 5F 50 99 04 83 14 27 8A B9 37 90 96 10 4D C2 EC 5B  
 2A FF 29 61 2D 16 06 E7 C1 67 56 95 FF F4 97 54 50 99 04 83 14 27 8A B9 37 90 96 10 4D C2 EC 5B

CBC:

30 62 35 17 65 2E 37 33 15 65 67 DA 75 0F AD 85 EF 4C 63 A2 F7 4A F9 D3 00 82 FF E7 94 4C F4 43  
 D9 A1 A3 95 66 E2 73 2B F3 29 EE 3A 10 75 92 08 27 D8 61 52 24 8D 23 35 1E 98 BA 29 19 22 8E 14

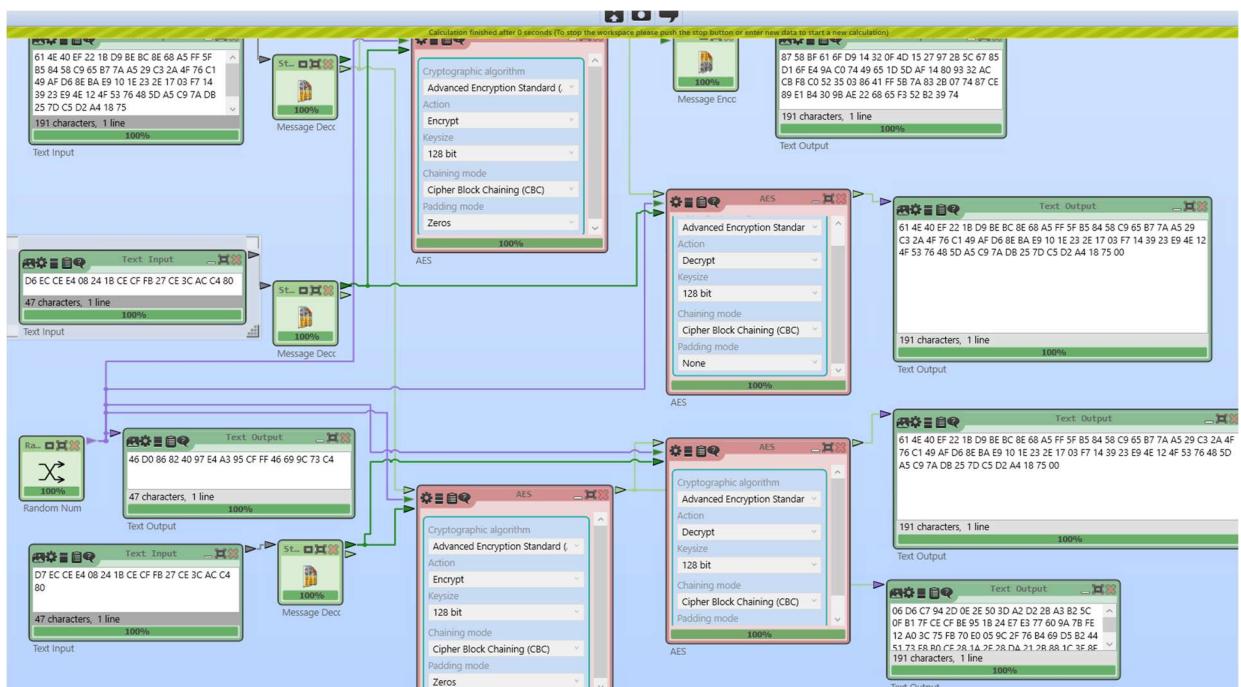
OFB:

38 4F F5 B1 D6 94 F9 35 23 29 A2 5C 0C EC 08 85 C9 ED 96 95 1E 0A 1F C3 A4 8B D5 04 81 7B 6F 09  
 39 4F F5 B1 D6 94 F9 35 23 29 A2 5C 0C EC 08 85 C9 ED 96 95 1E 0A 1F C3 A4 8B D5 04 81 7B 6F 09

Dla tych trybów pracy, które wykorzystują wartość inicjującą IV, zbadaj wpływ zmian tej wartości na uzyskany szyfrogram.

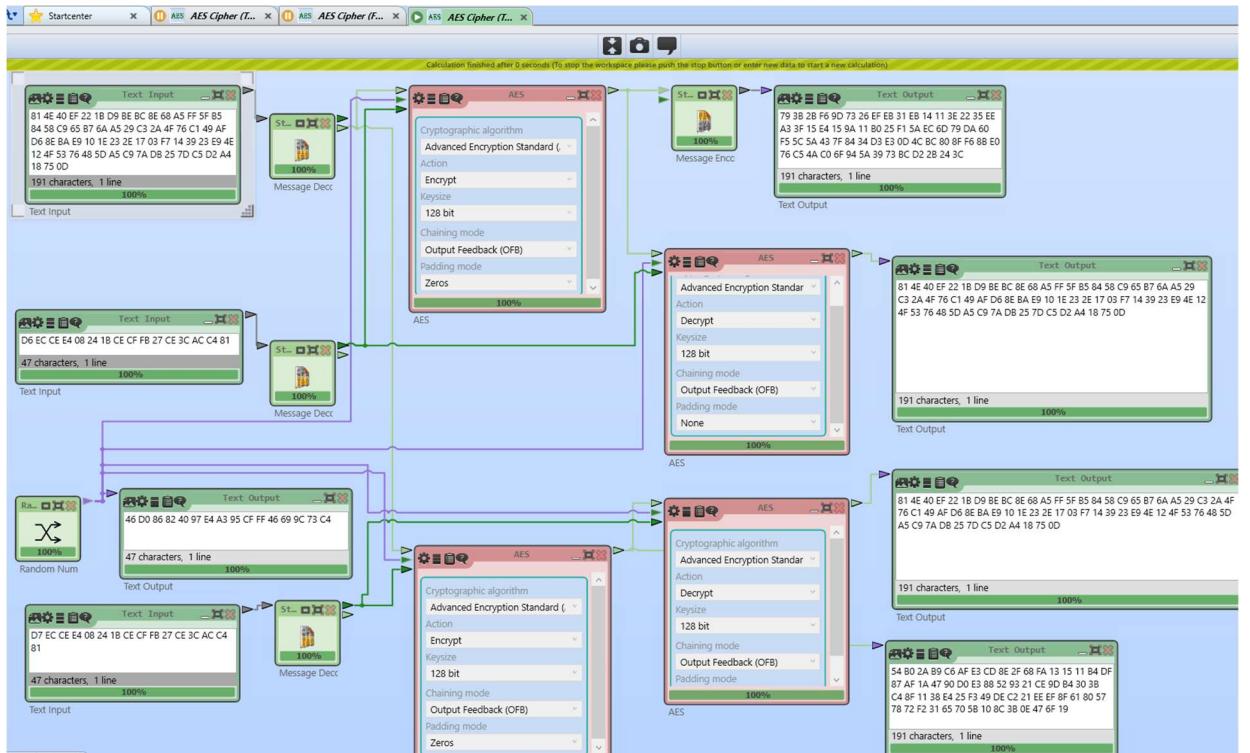
Porównaj zaimplementowane tryby szyfrowania pod względem wad i zalet oraz możliwości zastosowania.

Zmiana w drugim bajcie IV 6->7



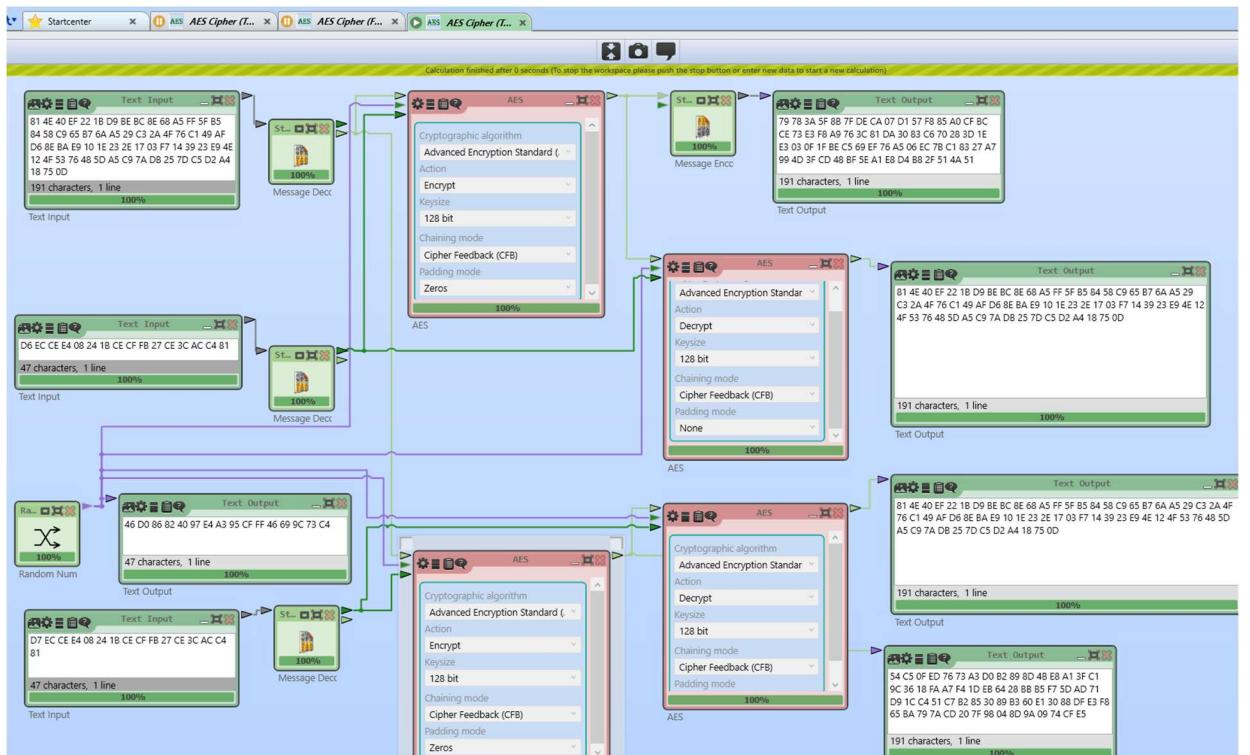
Całkowita zmiana

OFB:



## Całkowita zmiana

CFB:



### 1. ECB (Electronic Codebook):

Tryb ECB polega na niezależnym szyfrowaniu każdego bloku danych osobno.

Oznacza to, że ten sam blok danych otrzymuje ten sam zaszyfrowany blok.

Wadą tego trybu jest to, że powtarzające się bloki danych są widoczne w zaszyfrowanym tekście, co może prowadzić do utraty poufności i narażenia na ataki. ECB nie jest zalecany do szyfrowania dużych ilości danych.

### 2. CBC (Cipher Block Chaining):

W trybie CBC każdy blok danych jest łączony z poprzednim blokiem przed szyfrowaniem. Dodatkowo, przed pierwszym blokiem danych jest inicjalizator zwany wektorem inicjalizacji (IV). Szyfrowanie bloków jest zależne od wyniku poprzedniego bloku, co zapewnia większe bezpieczeństwo. Jednak wadą CBC jest brak równoległego przetwarzania bloków, ponieważ każdy blok zależy od poprzedniego. CBC jest bardziej odporny na ataki w porównaniu do ECB.

### 3. OFB (Output Feedback):

W trybie OFB używany jest szyfr blokowy jako generator strumienia klucza, który jest następnie kaskadowany z danymi wejściowymi. Generowany strumień klucza jest stosowany do operacji XOR z danymi wejściowymi, co prowadzi do uzyskania zaszyfrowanego tekstu. W trybie OFB, podobnie jak w CBC, używa się wektora inicjalizacji. Jedną z zalet trybu OFB jest możliwość równoległego przetwarzania danych, ponieważ generowany strumień klucza nie zależy od poprzednich bloków. Jednak wadą jest brak kontroli integralności danych.

### 4. CFB (Cipher Feedback):

W trybie CFB, podobnie jak w OFB, szyfr blokowy jest używany jako generator strumienia klucza. Jednak w tym trybie generowany strumień klucza jest używany w procesie szyfrowania poprzedniego bloku, a nie w operacji XOR z danymi wejściowymi. Dzięki temu CFB jest bardziej odporny na błędy transmisji danych w porównaniu do OFB. Wadą CFB jest brak równoległego przetwarzania bloków.

- a) Przy zmianie jednego znaku wartości inicjującej IV w każdym trybie szyfrowania szyfrogramy różniły się od siebie. Oznacza to, że zmiana wartości inicjującej IV znacząco wpływa na zmianę szyfrogramu.

Wpływ trybu pracy na propagację błędów transmisji

Dla wszystkich czterech opracowanych podstawowych trybów pracy zbadaj wpływ zmian w szyfrogramie na odzyskany tekst jawny (propagacja błędów transmisji). W tym celu zmień w szyfrogramie jeden bit i opisz jak zmieniły się odszyfrowane teksty jawne. Na podstawie tych obserwacji opisz jak wygląda propagacja błędów w poszczególnych trybach, tzn. jak różne błędy w odebranym szyfrogramie wpływają na tekst odszyfrowany.

M: 11 08 C7 65 68 3E 64 86 01 9E 6A 32 8A E7 CB 0D 75 43 36 09 BD 55 4E 20 0F 0A  
14 26 C1 68 08 C7 65 68 3E 64 86 01 9E 6A 32 8A E7 CB 0D 75 43 36 CB 5C DB D4 D3  
23 F4 E8 11 0F A3 8B 67 A5 10 74

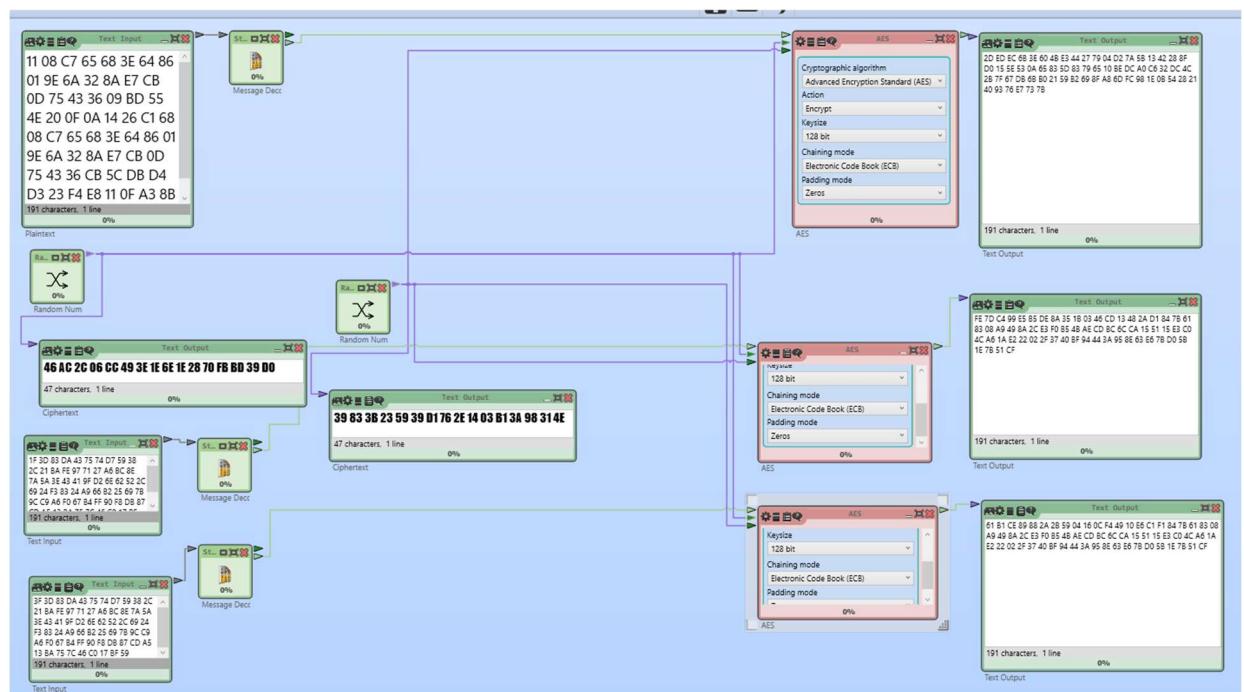
Wektor: 46 AC 2C 06 CC 49 3E 1E 6E 1E 28 70 FB BD 39 D0

Klucz: 39 83 3B 23 59 39 D1 76 2E 14 03 B1 3A 98 31 4E

**Zmiana w szyfrogramie**

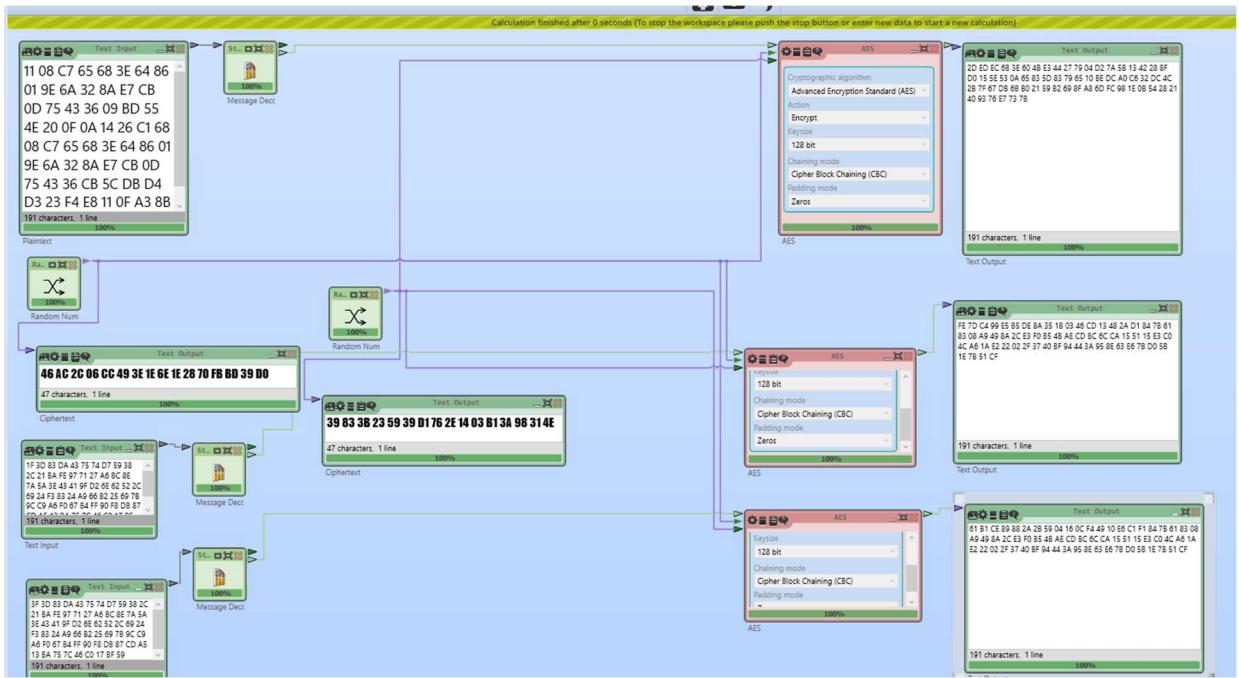
**3->1**F 3D 83 DA 43 75 74 D7 59 38 2C 21 BA FE 97 71 27 A6 BC 8E 7A 5A 3E 43 41  
 9F D2 6E 62 52 2C 69 24 F3 83 24 A9 66 B2 25 69 7B 9C C9 A6 F0 67 B4 FF 90 F8 DB  
 87 CD A5 13 BA 75 7C 46 C0 17 BF 59

ECB



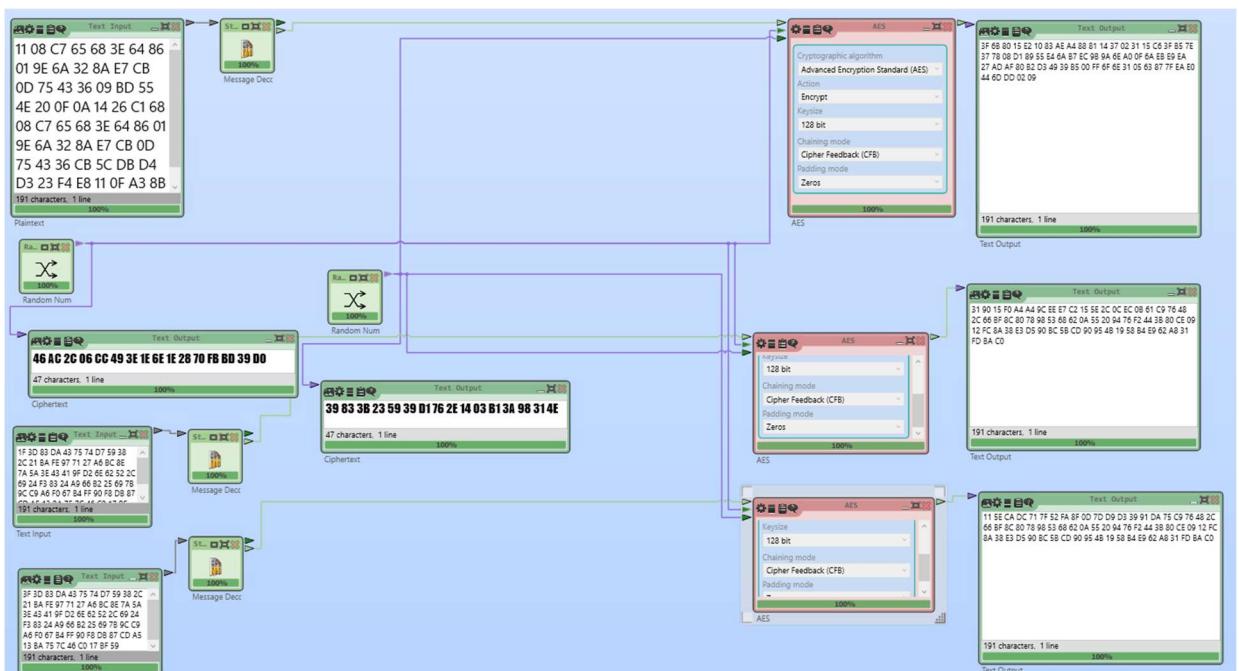
Zmiana jednego bitu powoduje zmianę całego bloku zawierającego ten bit w odszyfrowanym tekście jawnym.

CBC



Zmiana jednego bitu powoduje zmianę całego bloku posiadającego ten bit w odszyfrowanym tekście jawnym.

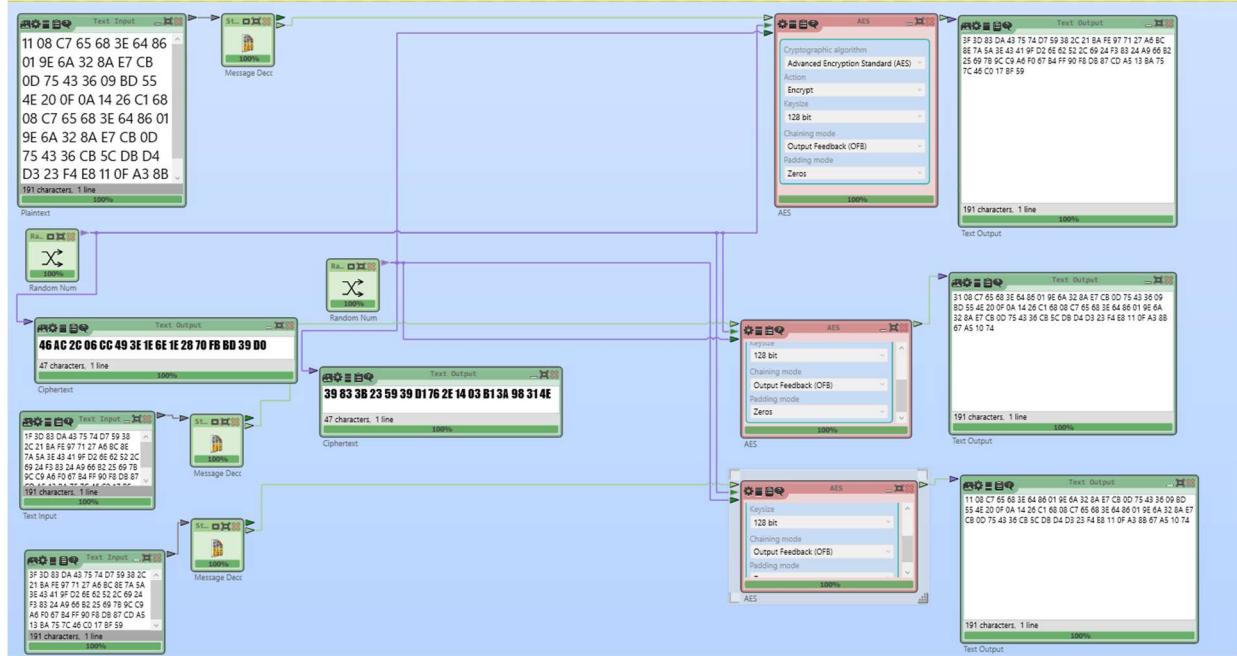
## CFB



Zmiana jednego bitu powoduje zmianę tego samego bitu oraz wszystkich

kolejnych bitów w odszyfrowanym tekście jawnym.

OFB:



Zmiana tych samych bitów w szyfrogramie powoduje zmianę tego samego bitu w odszyfrowanym tekście. W naszym wypadku tylko 1 tablica zmienia się z 11 na 31

Wypisz nazwy innych szyfrów blokowych, które są dostępne w programie CrypTool 2.1.

- c) Camellia Cipher
- d) Blowfish Cipher
- e) Twofish Cipher
- f) Threefish Cipher
- g) RC2
- h) DES