

Politechnika Wrocławska, Informatyka Stosowana

# BLOKOWE ALGORYTMY SZYFROWANIA

Cyberbezpieczeństwo, Laboratorium nr.3 - raport

Autor: Aleksander Stepaniuk  
Nr. Indeksu: 272644

## **Zad 1. Przykładowe algorytmy blokowe.**

Teksty których użyłem do analizy kolejnych algorytmów:

### **Tekst 1:**

Litera „n” powtórzona 2000 razy.

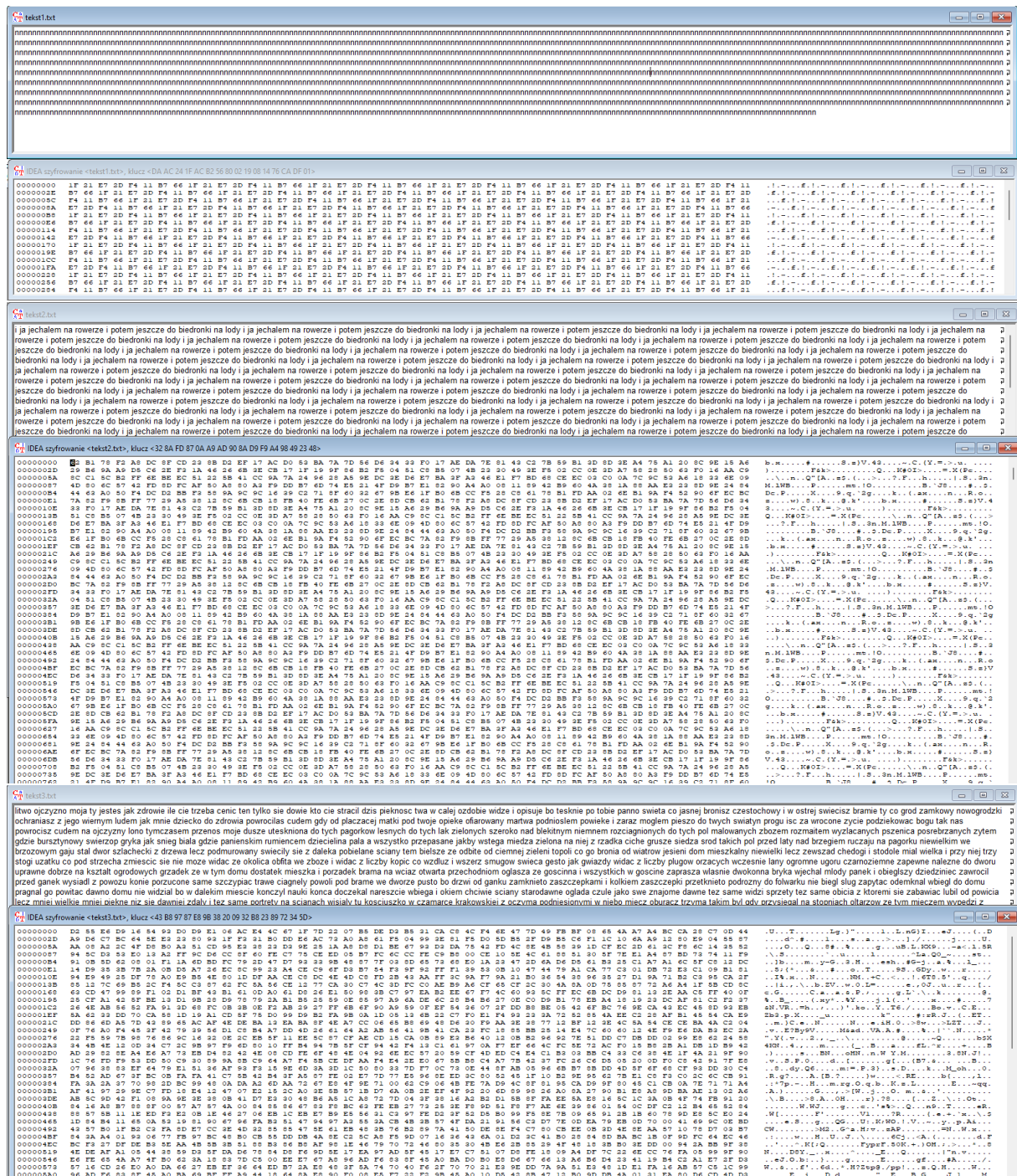
### **Tekst 2:**

Tekst „i potem jeszcze do biedronki na lody” powtórzony 1000 razy.

### **Tekst 3:**

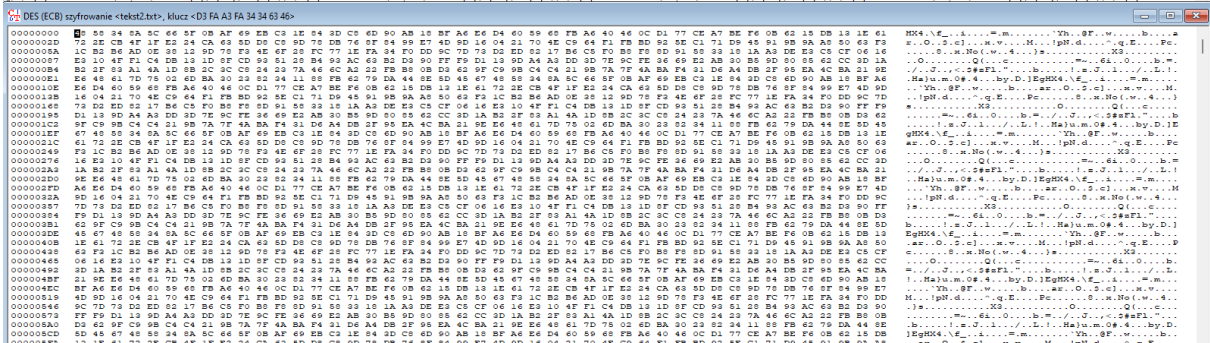
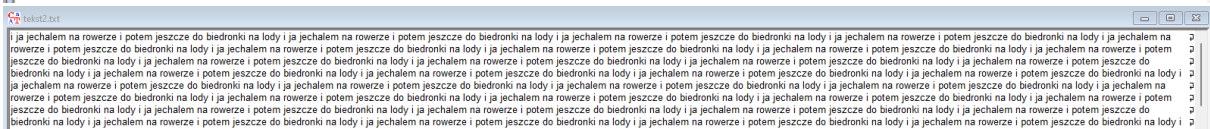
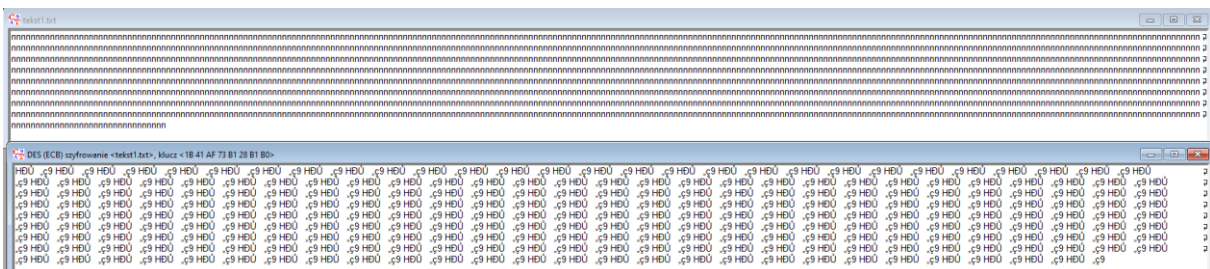
5000 pierwszych znaków pana Tadeusza („litwo ojczyzno moja ty jesteś jak zdrowie ile cie trzeba cenic ten tylko sie dowie kto cie stracil...”)

IDEA:

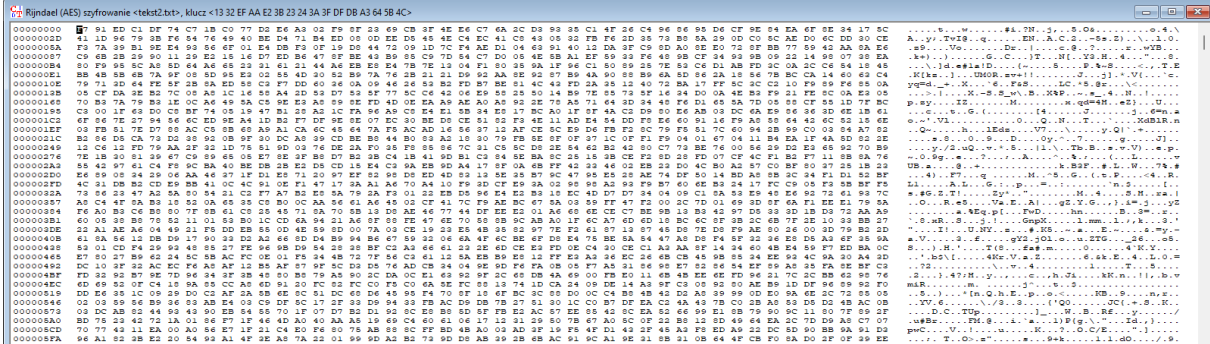
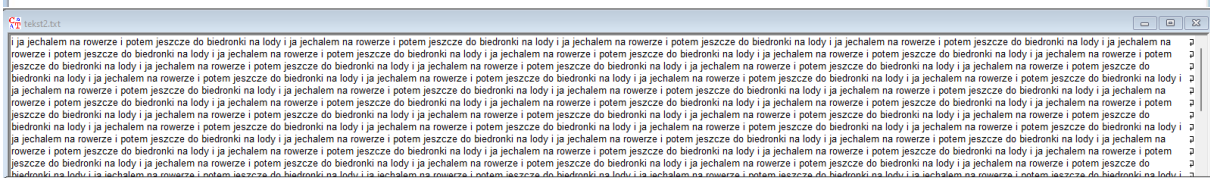
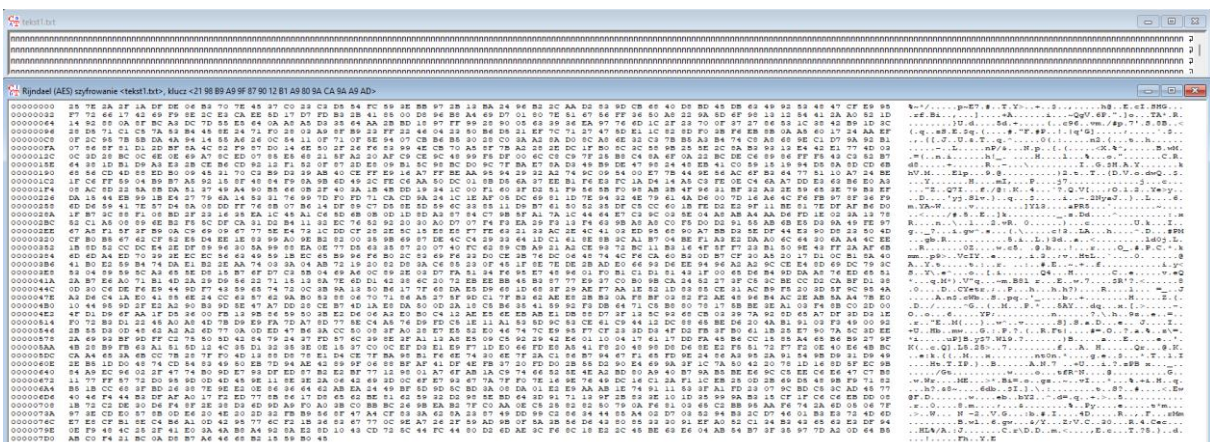




## DES (z ECB):



## AES (z CBC): (128 bitów klucz)

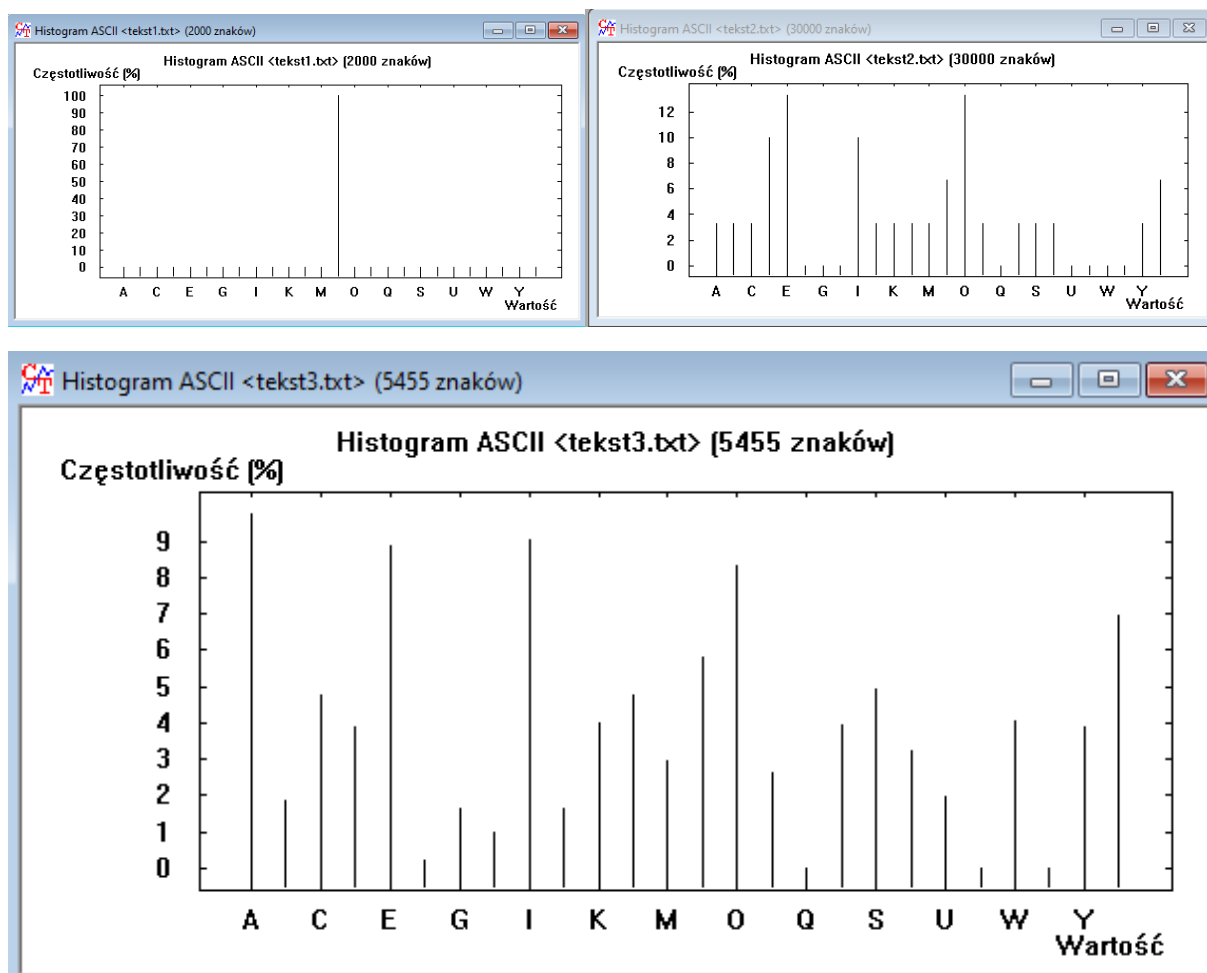


## Zadanie 1.1;

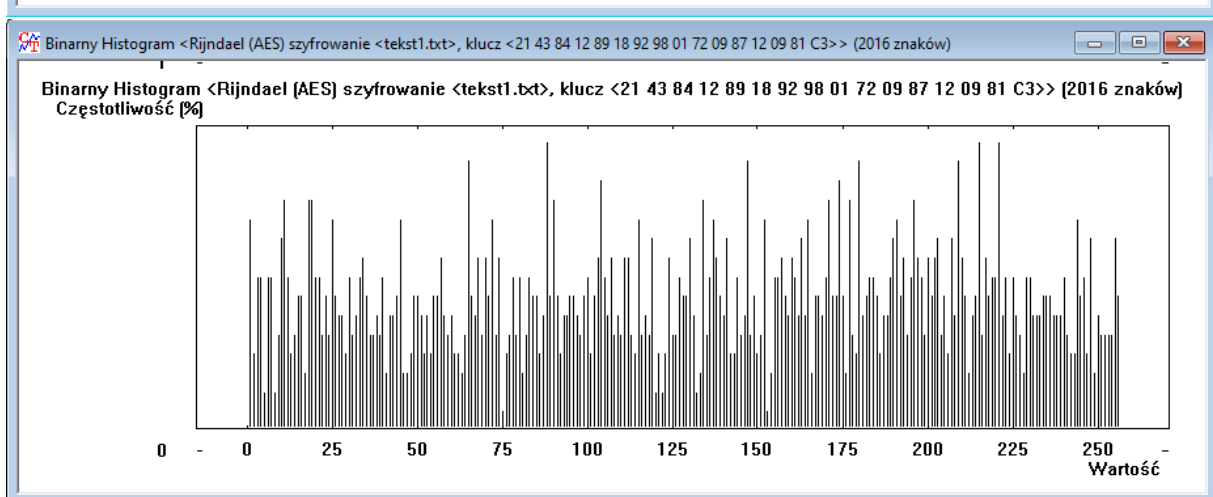
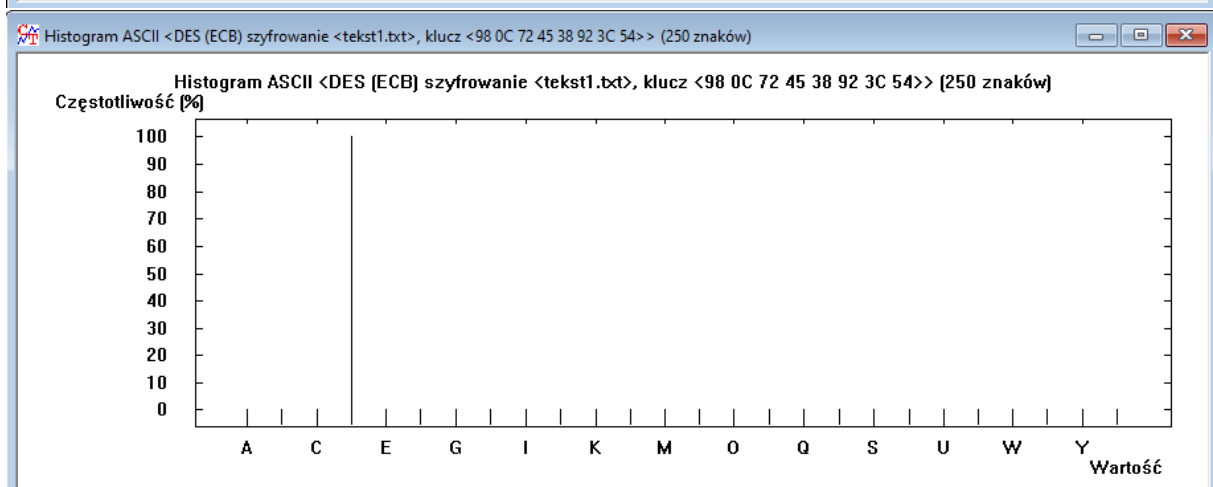
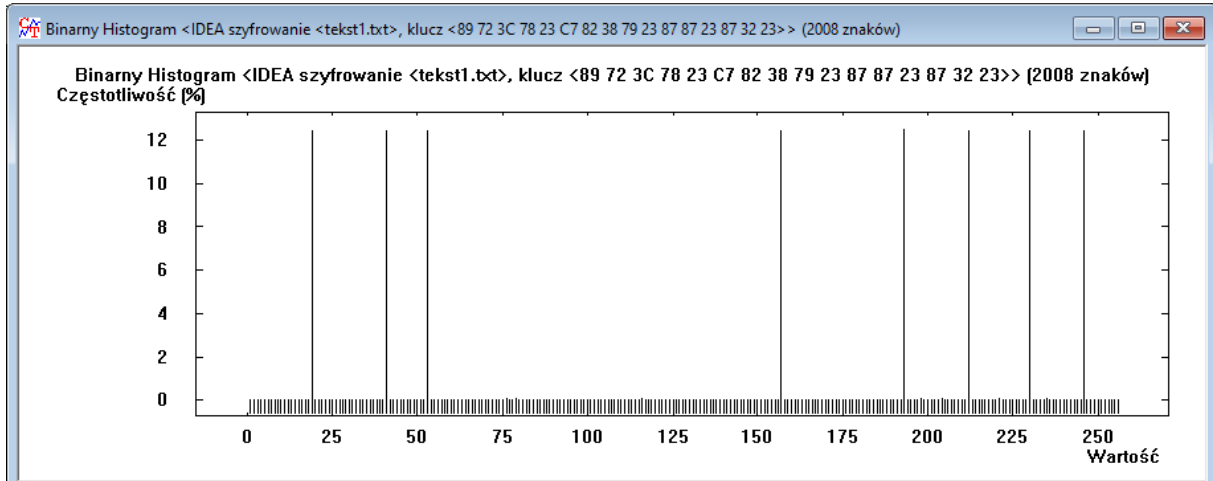
W tabeli poniżej znormalizowano entropię wszystkich tekstów w taki sposób, aby maksymalna możliwa wartość wynosiła 4,70.

Entropia tekstów zależnie od algorytmu		
Algorytm	Tekst jawny	Tekst zaszyfrowany
IDEA	0,00	1,78
	3,92	4,19
	4,25	4,67
DES (ECB)	0,00	1,00
	3,92	4,24
	4,25	4,68
AES (CBC 128 bit klucz)	0,00	4,64
	3,92	4,69
	4,25	4,67

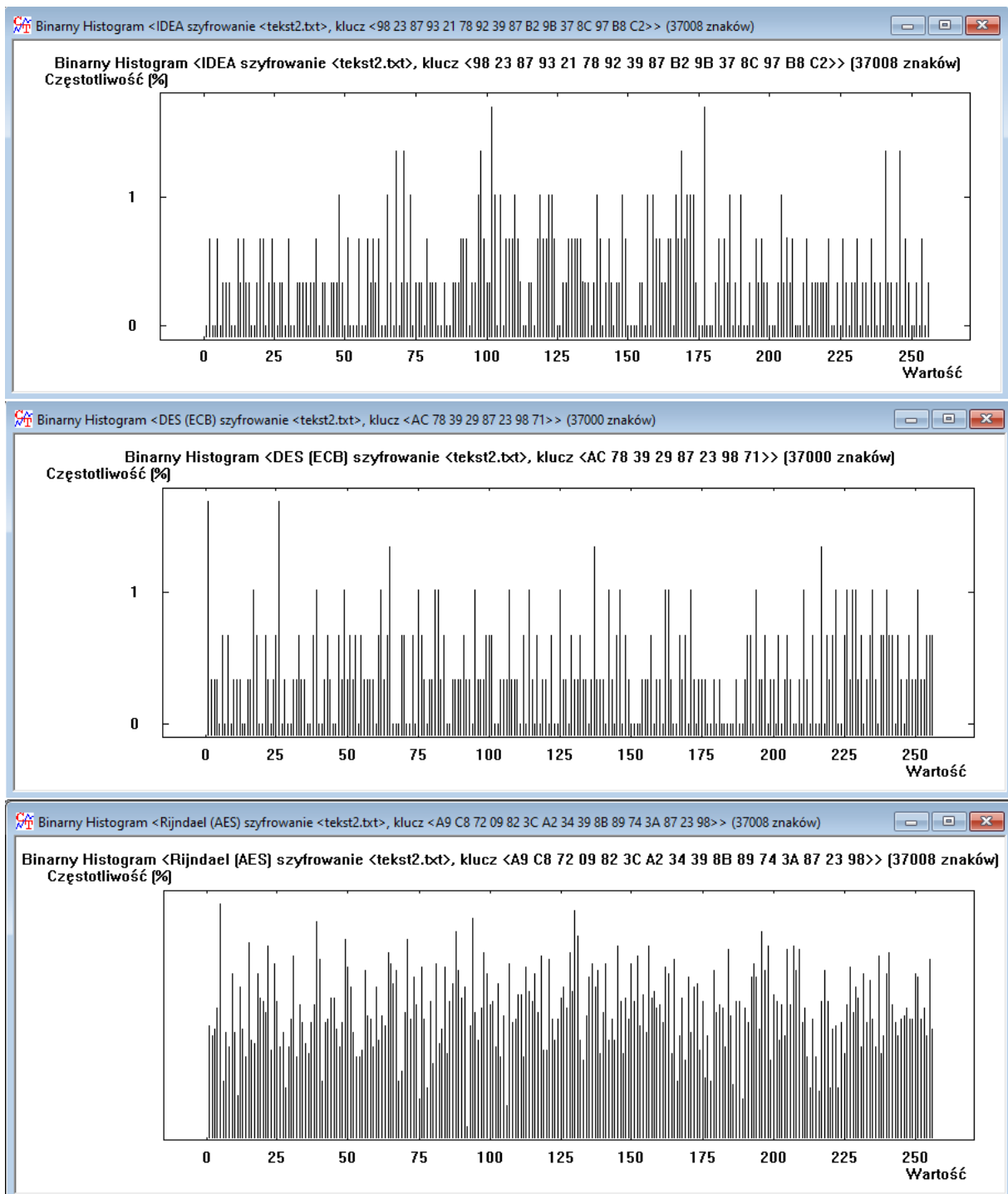
Histogramy dla tekstów jawnych:



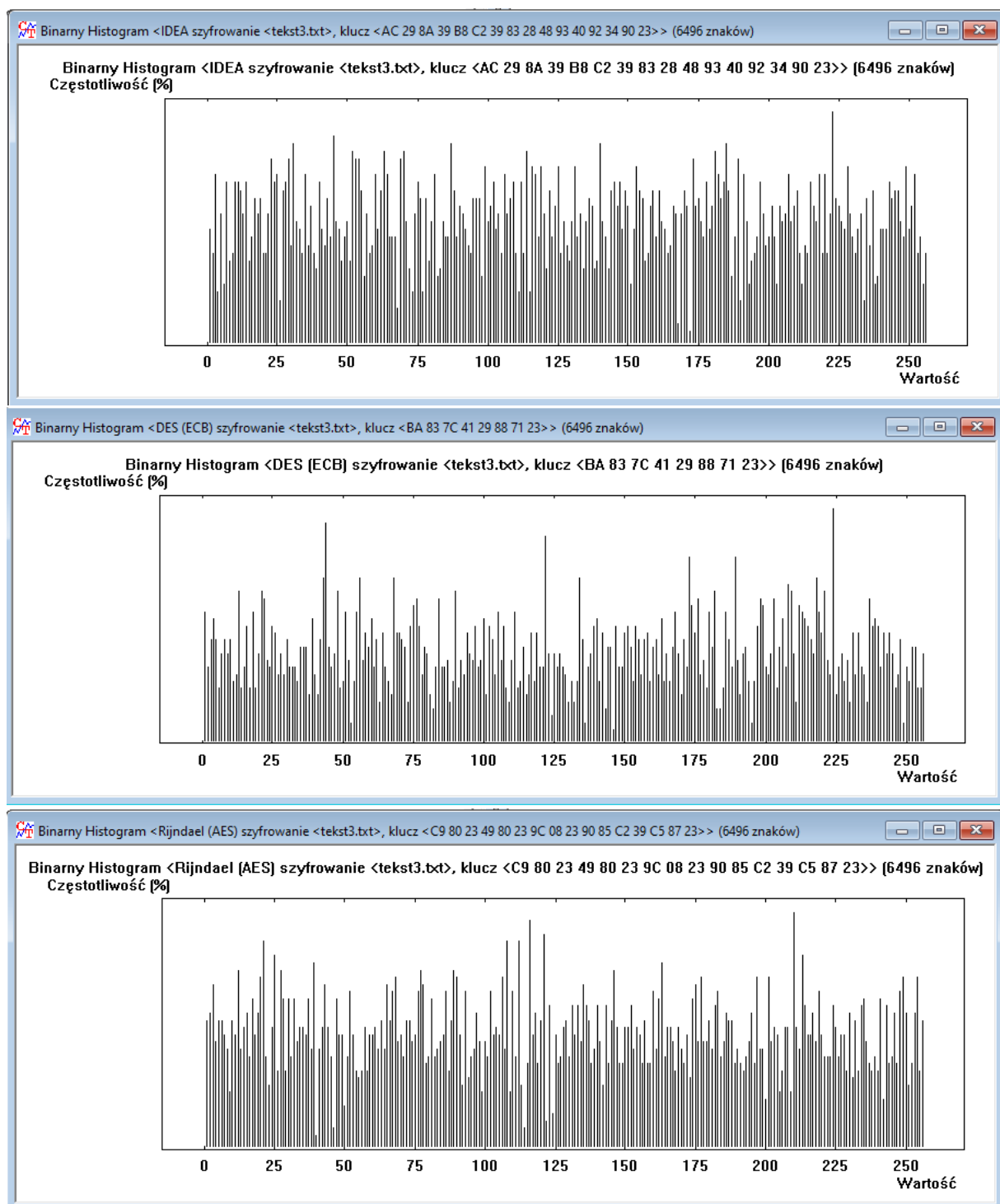
Histogramy dla tekstów tajnych z tekstu 1: (kolejno IDEA, DES (ECB), AES (CBC)):



Histogramy dla tekstów tajnych z tekstu 2: (kolejno IDEA, DES (ECB), AES (CBC)):



Histogramy dla tekstów tajnych z tekstu 3: (kolejno IDEA, DES (ECB), AES (CBC)):





## Zadanie 1.2;

Długości kluczy dla kolejnych algorytmów:

### IDEA:

K1 – 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

K2 – 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78

K3 – AC C9 49 08 3C 09 90 87 DF 49 80 7B F0 32 03 29

### DES (ECB):

K4 – 00 00 00 00 00 00 00 00

K5 – 12 34 12 34 12 34 12 34

K6 – AC C9 49 08 3C 09 90 87

### AES (CBC):

K7 – 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

K8 –

11 11

K9 –

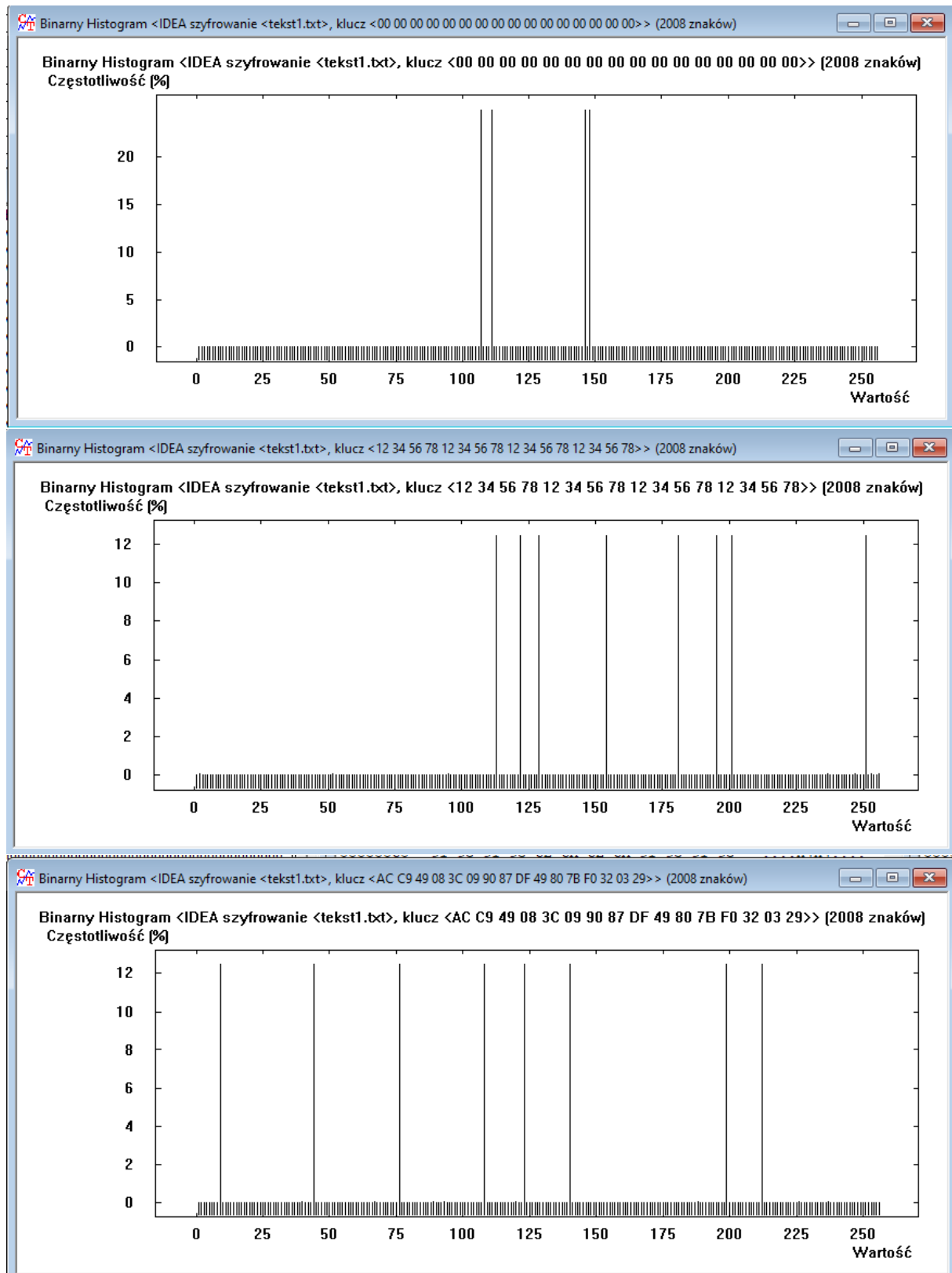
AC C9 49 08 3C 09 90 87 DF 39 A9 BC 10 02 05 8B CC 21 99 12 87 B9 F0 71 7A 4B C0 45 3B 89 6B A8

W tabeli poniżej znormalizowano entropię wszystkich tekstów w taki sposób, aby maksymalna możliwa wartość wynosiła 4,70.

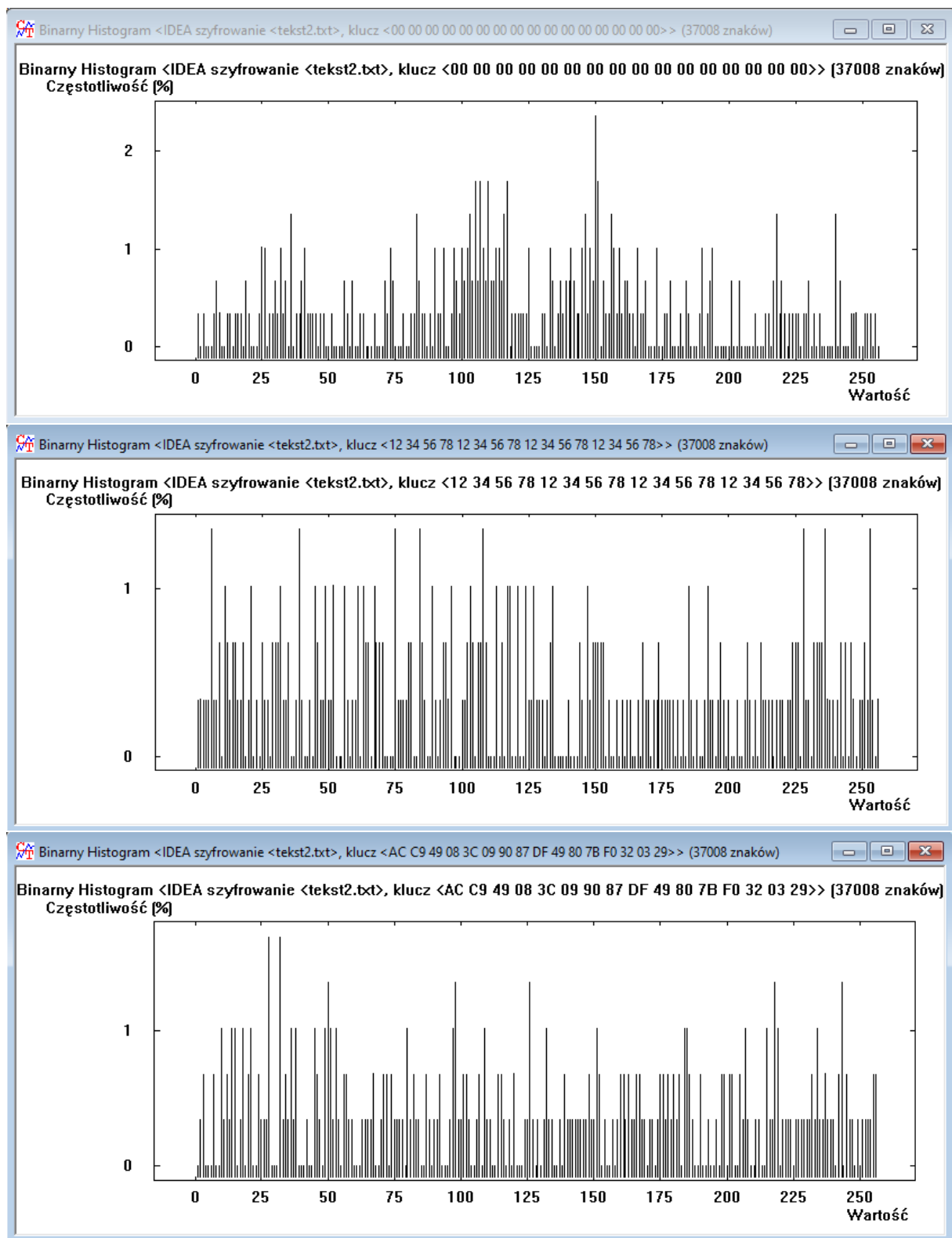
Entropia TT oraz TJ zależnie od szyfru, długości i wartości klucza										
Szyfr		IDEA			DES (ECB)			AES (CBC)		
Klucz		K1	K2	K3	K4	K5	K6	K7	K8	K9
tekst nr.	TJ	TT								
Tekst1	0,00	1.19	1,78	1,78	1.76	0,00	1.76	4,64	4,64	4,64
Tekst2	3,92	4.16	4,27	4,29	4.23	4.31	4.26	4,69	4,69	4,69
Tekst3	4,25	4,51	4,67	4,68	4.68	4.68	4.67	4,68	4,68	4,68

Histogramy:

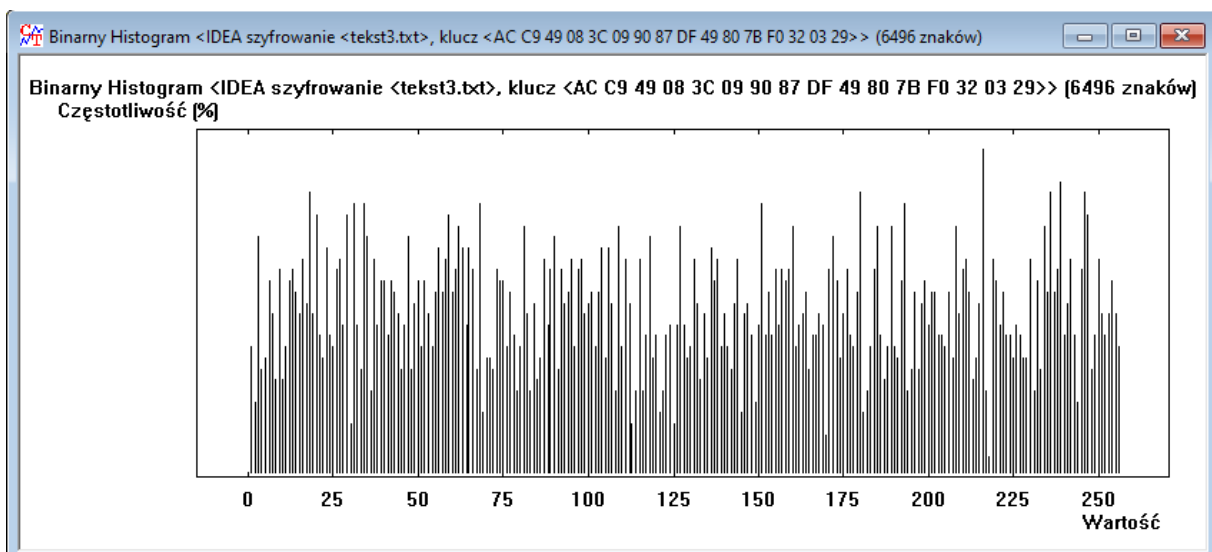
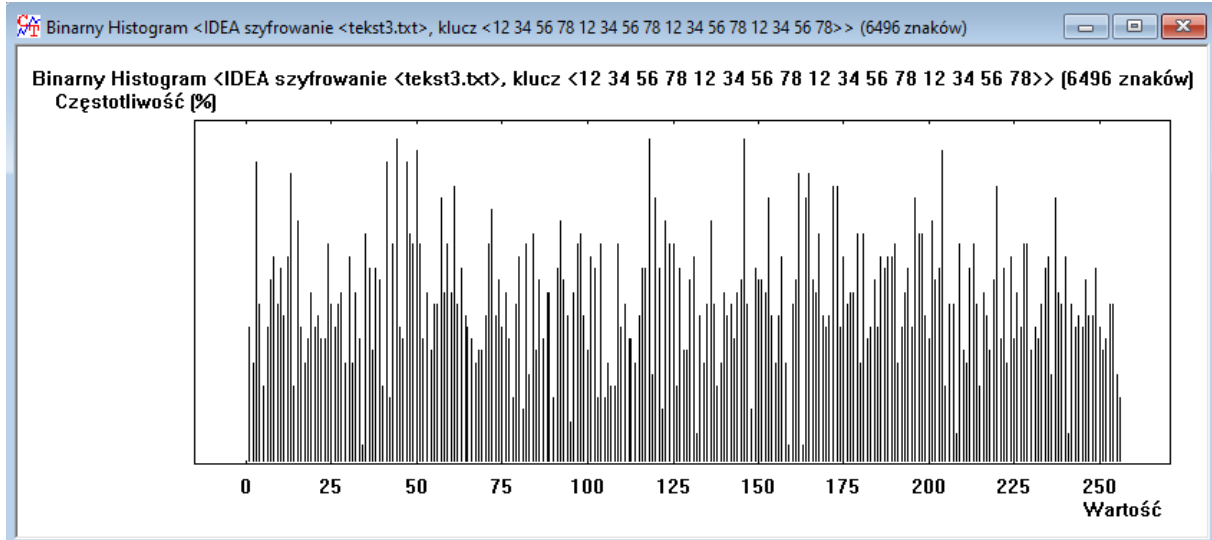
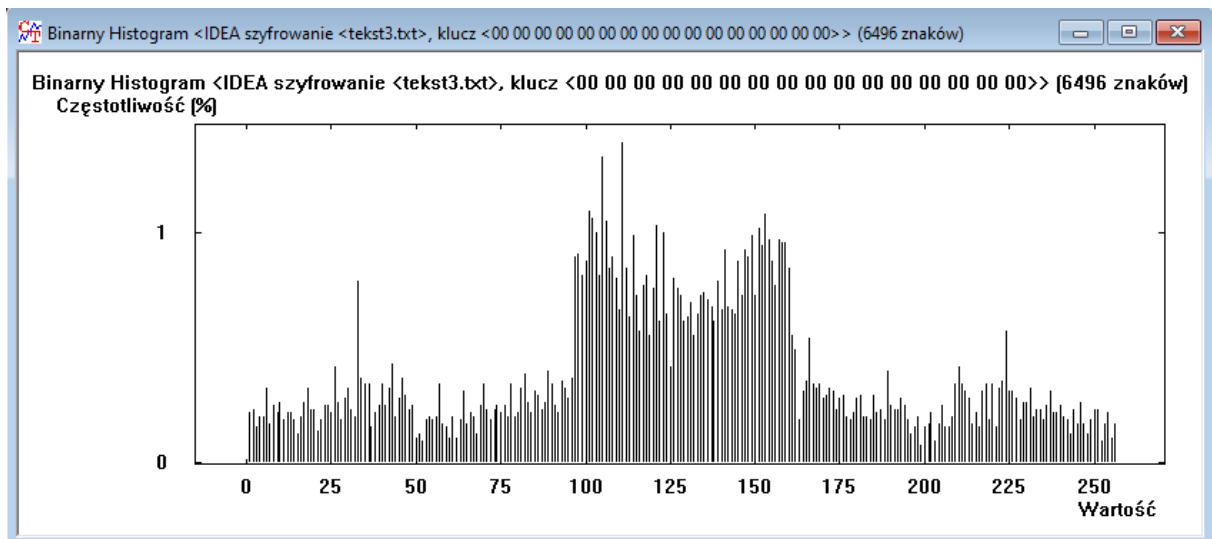
IDEA, tekst 1:



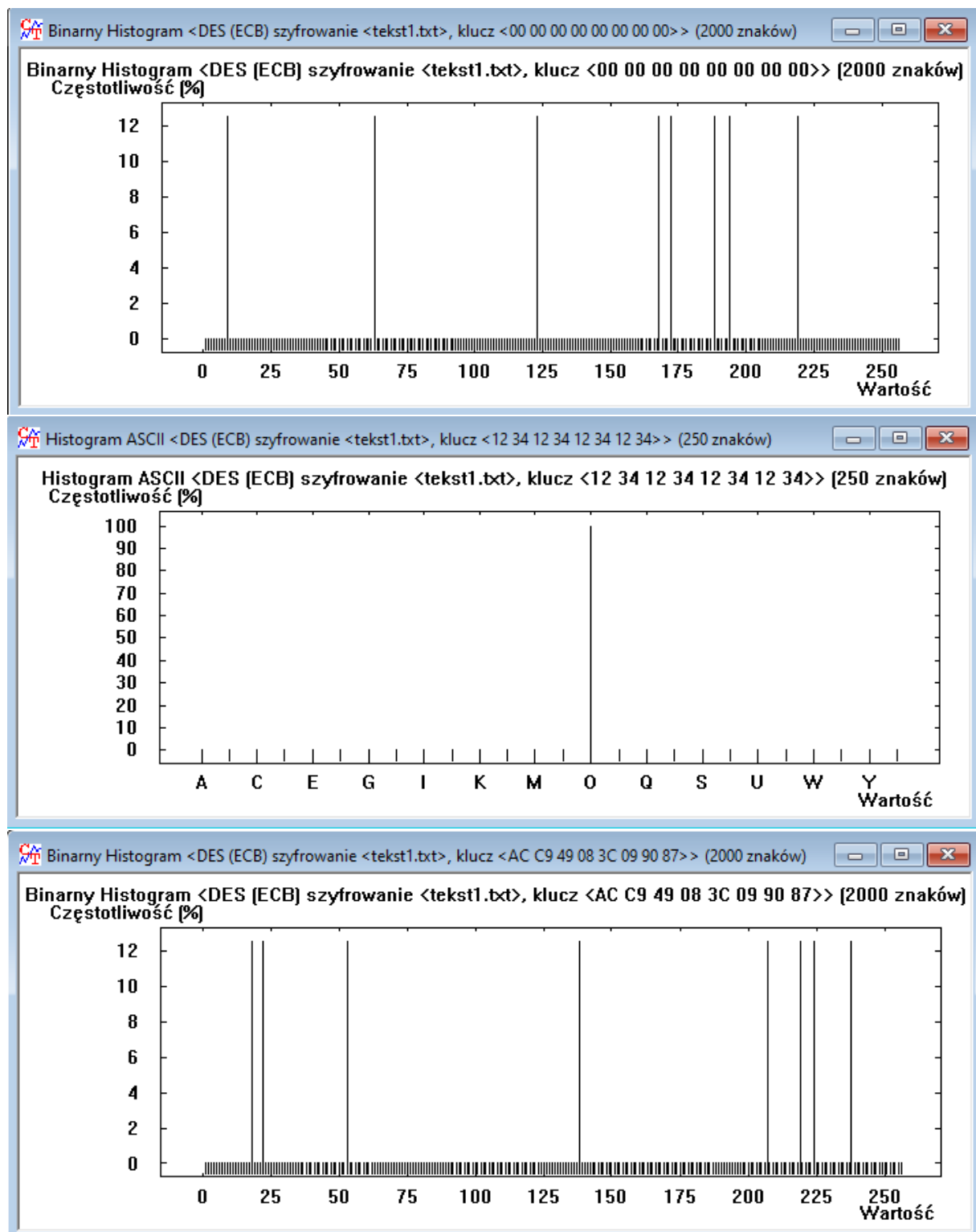
IDEA, tekst 2:



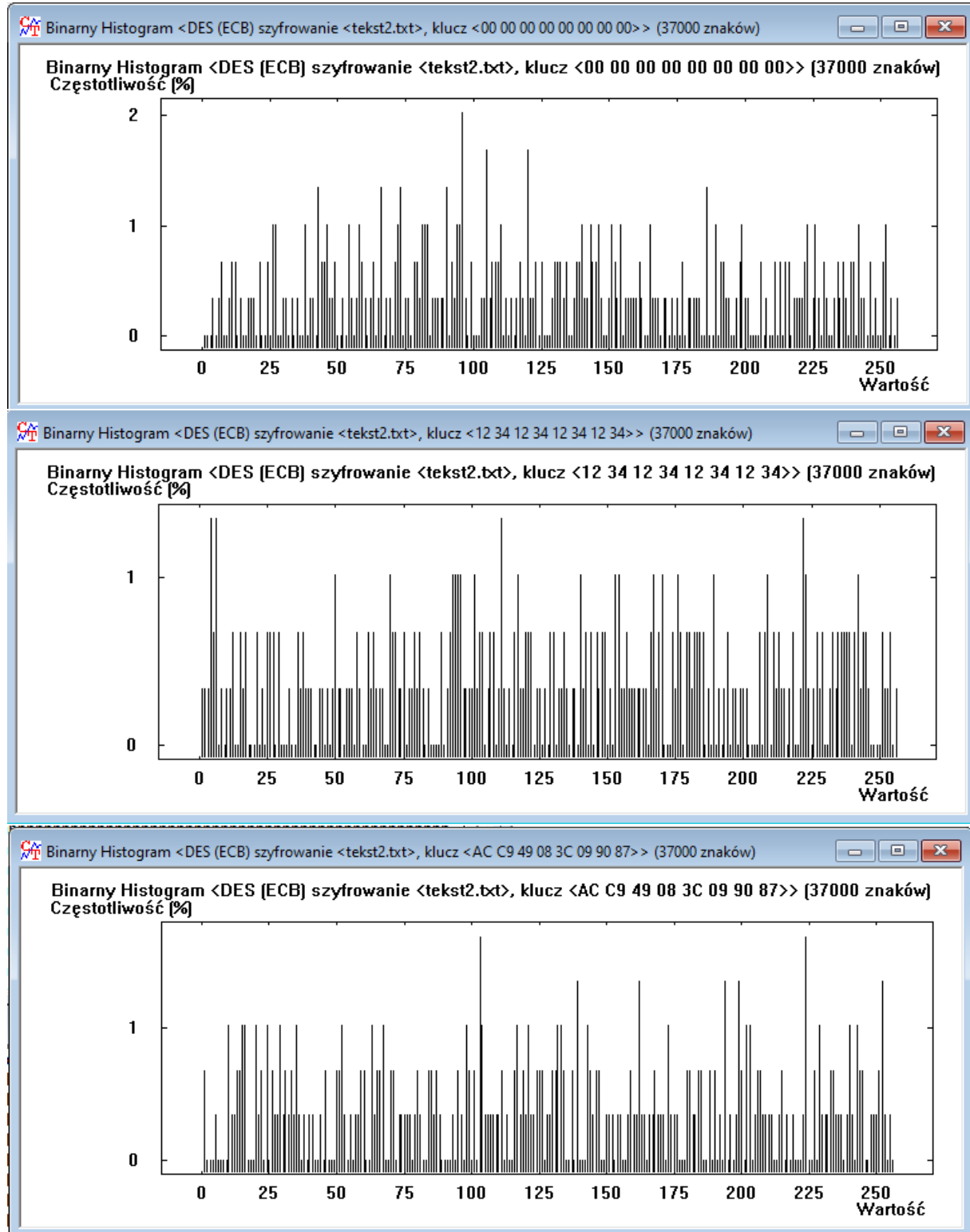
IDEA, tekst 3:



DES (ECB), tekst 1:

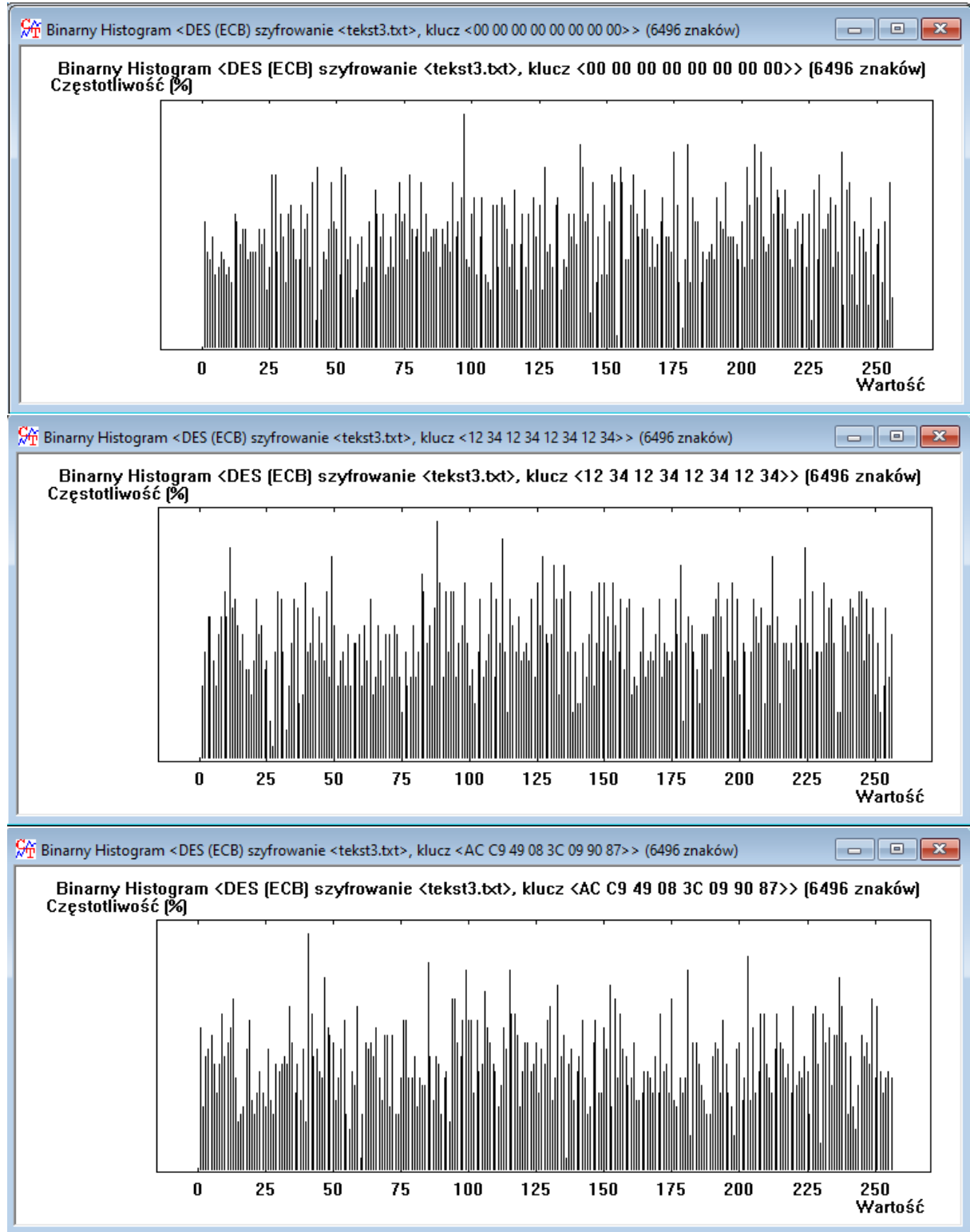


DES (ECB), tekst 2:

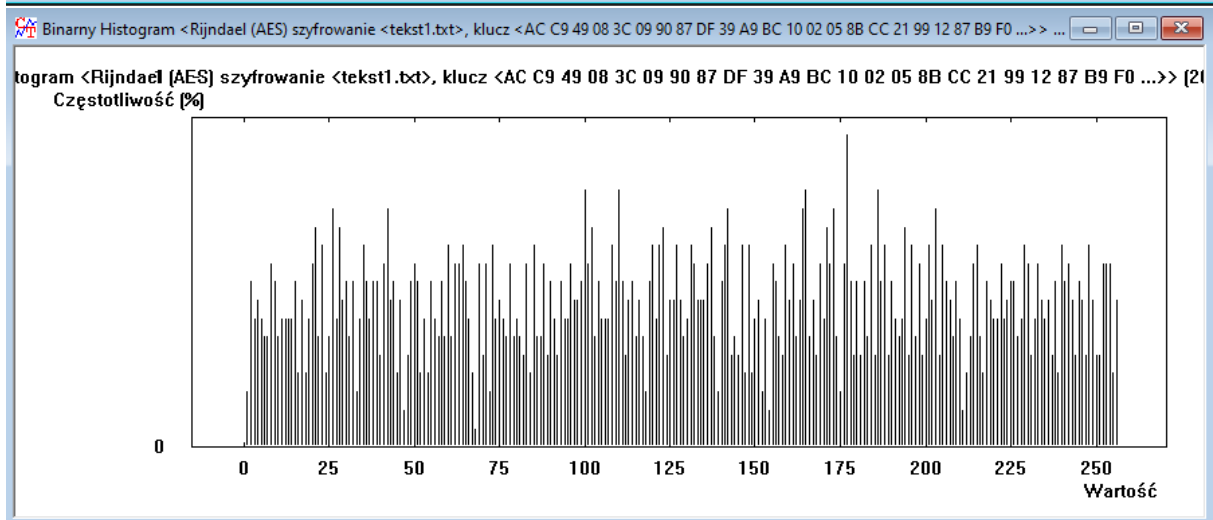
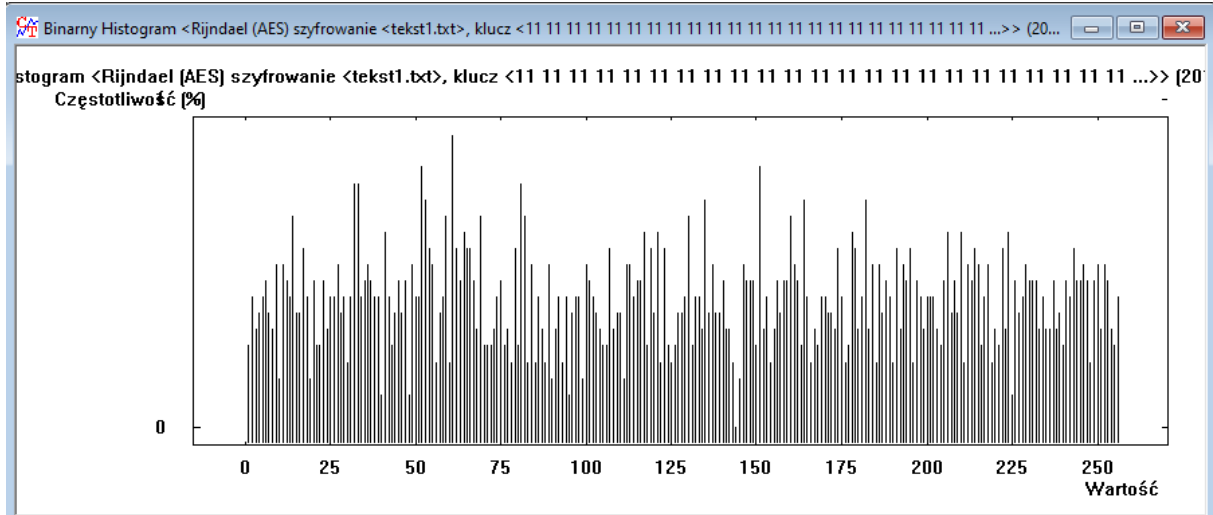
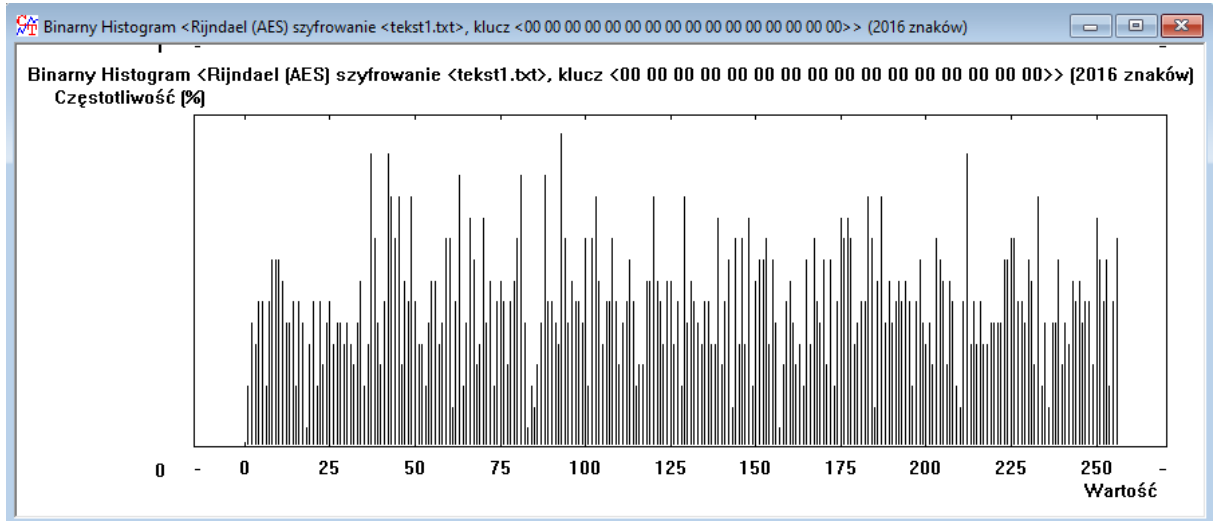




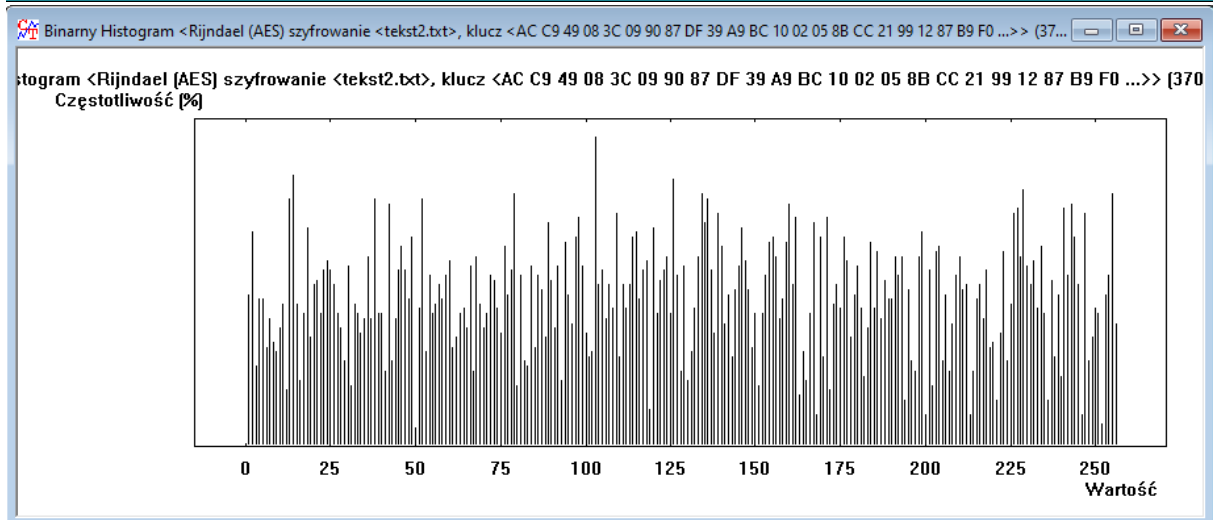
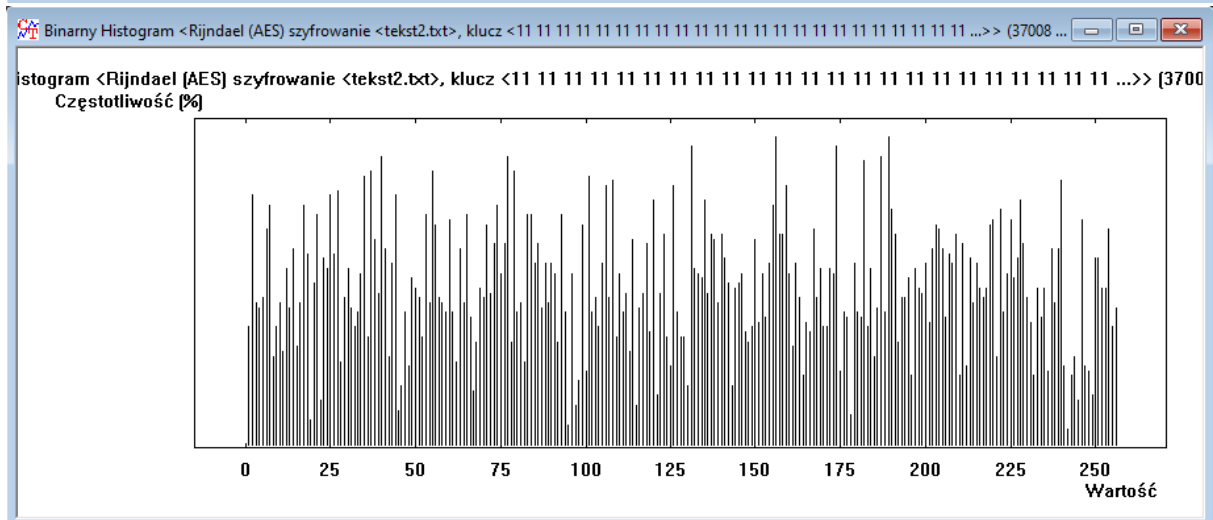
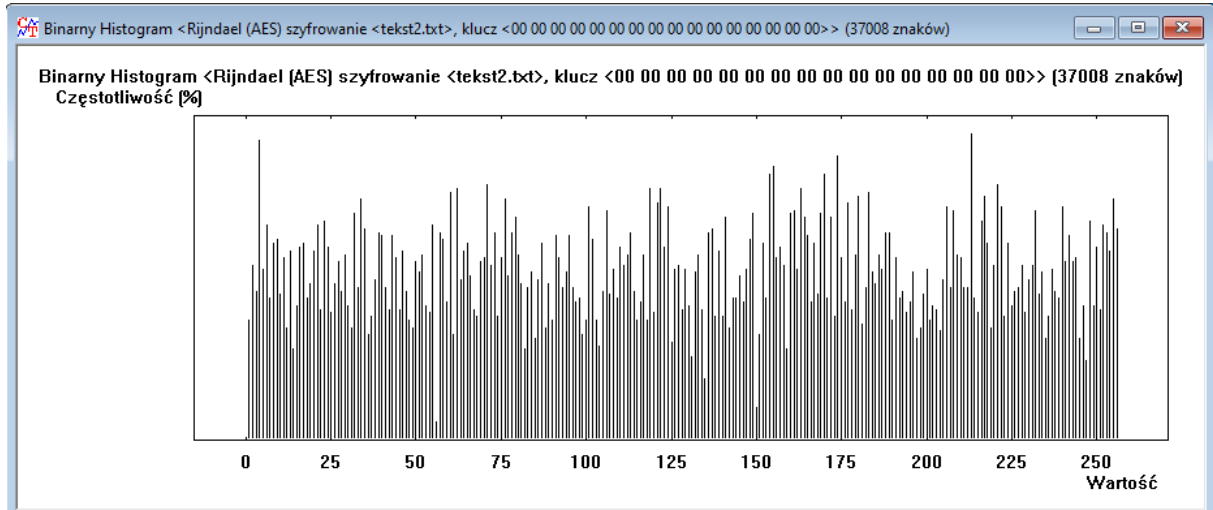
DES (ECB), tekst 3:



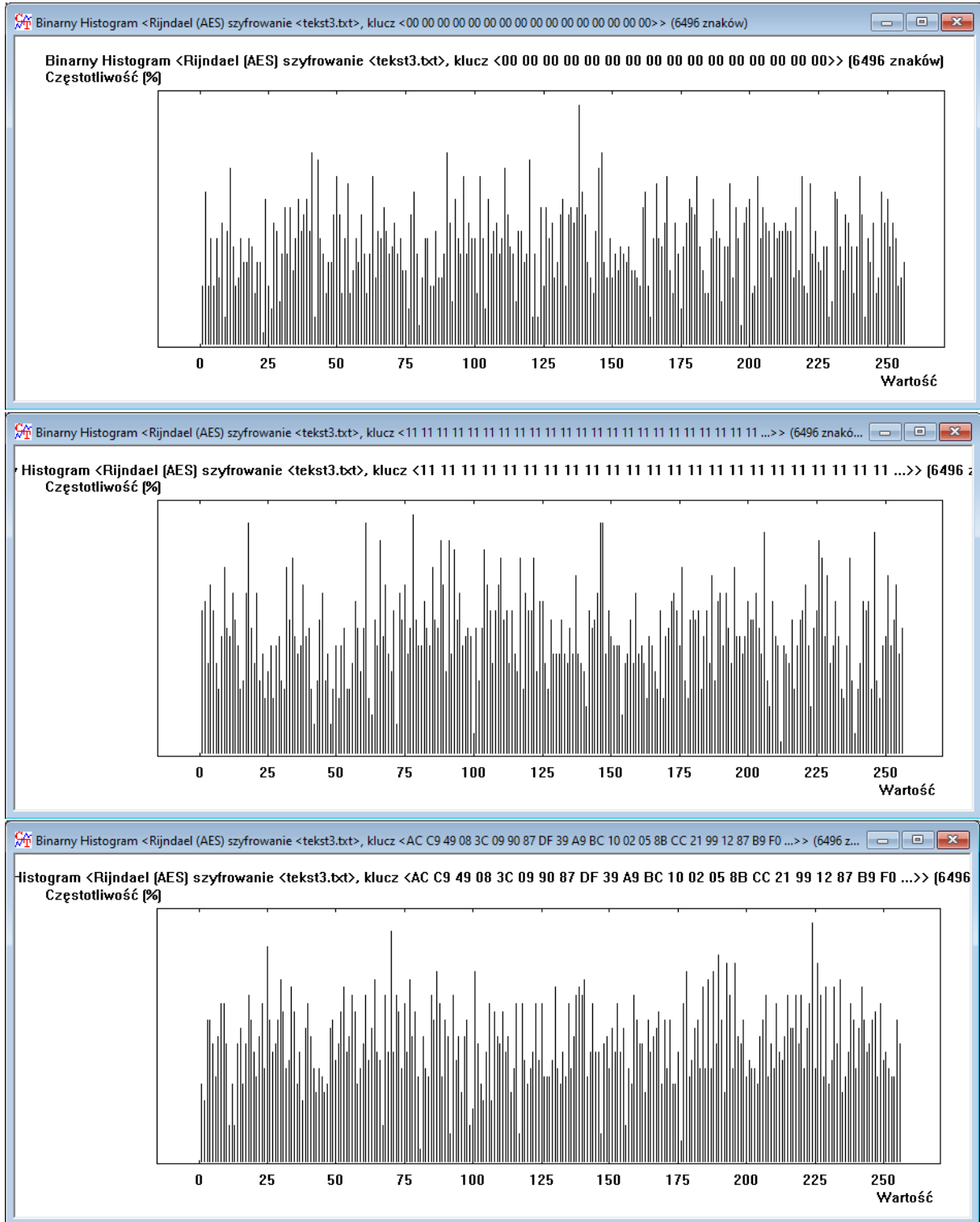
## AES (CBC), tekst 1:



## AES (CBC), tekst 2:



### AES (CBC), tekst 3:



### Pytanie 1.3;

Algorytmy blokowe są stosowane głównie w kryptografii do szyfrowania danych. Wykorzystuje się je w generatorach liczb pseudolosowych, adresach MAC oraz w mechanizmach do sprawdzania integralności danych. Podstawowe zastosowania algorytmów blokowych to:

- szyfrowanie danych (np. plików lub wiadomości)
- bezpieczeństwo transmisji w sieciach (np. VPN, HTTPS)
- szyfrowanie dysków (np. BitLocker)
- systemy bankowe i płatności online (szyfrowanie danych kart płatniczych)

Popularne algorytmy blokowe:

- AES (Advanced Encryption Standard) - najczęściej stosowany algorytm blokowy, uznawany za standard
- DES (Data Encryption Standard) - obecnie uznawany za przestarzały, zastąpiony przez AES
- 3DES (Triple DES) - wzmocniona wersja DES, ale powoli wychodzi z użycia
- Blowfish - stosowany w starszych systemach
- Twofish - następca Blowfish, również bezpieczny, ale mniej popularny niż AES

Standardowe i bezpieczne parametry:

- Długość bloku: 128 bitów (AES), choć niektóre algorytmy (np. Blowfish) używają 64-bitowych bloków, co jest uznawane za raczej mniej bezpieczne
- Długość klucza: AES obsługuje klucze o długości 128, 192 lub 256 bitów. Długość klucza 128 bitów jest uznawana za wystarczającą dla większości zastosowań, ale 256-bitowe klucze są stosowane w szczególnie wrażliwych aplikacjach.

### Pytanie 1.4;

Dla wszystkich algorytmów blokowych entropia wzrosła po zaszyfrowaniu tekstów jawnych. W przypadku tekstu nr. 1 algorytmy DES i IDEA spowodowały najmniejszy wzrost entropii, a zaszyfrowany tekst wciąż posiadał znaki powtarzających się znaków, co widać dokładnie na histogramach dla tego tekstu. Z kolei na AES entropia wzrosła znacząco, zbliżając się za każdym razem do maksymalnej ( $4,64/4,7$  – po przeskalowaniu), a słupki histogramów zdawały się być bardziej wyrównane w swoim rozkładzie, a znaki nie były tak powtarzalne jak dla dwóch poprzednich algorytmów.

Dla tekstu nr. 2 entropia także była wyższa w przypadku użycia AES w porównaniu do DES czy IDEA, ale różnica ta była o wiele mniejsza niż w przypadku tekstu nr. 1. Histogramy dla AES wydają się być bardziej wyrównane, gdzie w przypadku DES i IDEA niektóre znaki występowały znacznie częściej co było bardzo zauważalne na histogramach.

Dla tekstu nr. 3 entropia wzrosła w podobny sposób dla wszystkich algorytmów. Wszystkie histogramy wydają się być zbliżone, niezależnie od użytego algorytmu. IDEA najgorzej zmienia entropię dla klucza składającego się z samych zer (tworzyła widoczne wzniesienie na środku rozkładu). Dla AES zmiana entropii była znacząca, niezależnie od rodzaju klucza czy natury tekstu, a histogramy były wypłaszczone w każdym jego szyfrowaniu.

### Pytanie 1.5;

Dla niektórych algorytmów klasycznych entropia albo nie ulegała zmianie wcale, albo dla takiego adfgvx nawet malała. Wzrost entropii zależał od długości i poziomu skomplikowania klucza. Histogramy często zachowywały swój kształt, ale były przesunięte, co mogło ujawnić użyty algorytm (np. Cezar).

Dla algorytmów blokowych z kolei entropie rośnie znacząco, co utrudnia łamanie szyfrów prostymi metodami, jak analiza histogramów, które są w tym przypadku dużo bardziej wyrównane w poziomie.

### Pytanie 1.6;

Dla algorytmu z możliwością zmiany długości klucza AES wydłużenie klucza nie dało żadnej zauważalnej zmiany w entropii. Różnica ta jest prawdopodobnie minimalna i niezauważalna zarówno przy badaniu samego tekstu jak i jego histogramu.

### Pytanie 1.7;

Dla zaszyfrowanego tekstu entropia w dużej mierze zależy od entropii tekstu jawnego. Jednak dla lepszych algorytmów takich jak AES nie ma to większego znaczenia – entropia pozostaje wysoka, niezależnie od tego czy tekst jawny jest jednorodny czy bardziej złożony.

### Pytanie 1.8;

Dla szyfrowania IDEA wartość klucza zdecydowanie wpływa na entropię tekstu zaszyfrowanego. Szczególnie jest to widoczne dla klucza K1, gdzie entropia jest nieco niższa od pozostałych kluczy dla tego algorytmu. Natomiast w pozostałych kluczach zmiana klucza nie miała znacznego wpływu na entropię tekstu.

### Pytanie 1.9;

Dla wszystkich algorytmów szyfrujących entropia tekstu zaszyfrowanego była zależna od użytego algorytmu. AES powodował zwiększenie entropii do wartości zbliżonej do tej maksymalnej (4,70 po przeskalowaniu) niezależnie długości klucza czy rodzaju szyfrowanego tekstu. Dla DES oraz IDEA widać było bardzo silną zależność między entropią tekstu jawnego a tajnego. Algorytmy te radziły sobie o wiele gorzej z szyfrowaniem tekstów nr. 1 i nr. 2 niż tekstu nr. 3 ze względu na ich większą jednorodność.

## Zad 2. Tryby pracy algorytmów blokowych.

### Zadanie 2.1-2.4;

Użyłem poniższych tekstów do kolejnych zadań:

#### Tekst 1:

Litera „n” powtórzona 2000 razy.

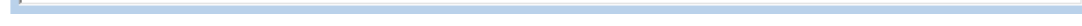
Do analizy wybrałem algorytm AES, w wariantcie 128-bitowym.

Klucz: B3 A4 09 34 3B C2 72 7B 34 C9 FF 87 96 05 3A F2

Wektor: 2B 17 A8 3B 91 C9 4F 37 E1 29 4D 79 12 A4 79 A1

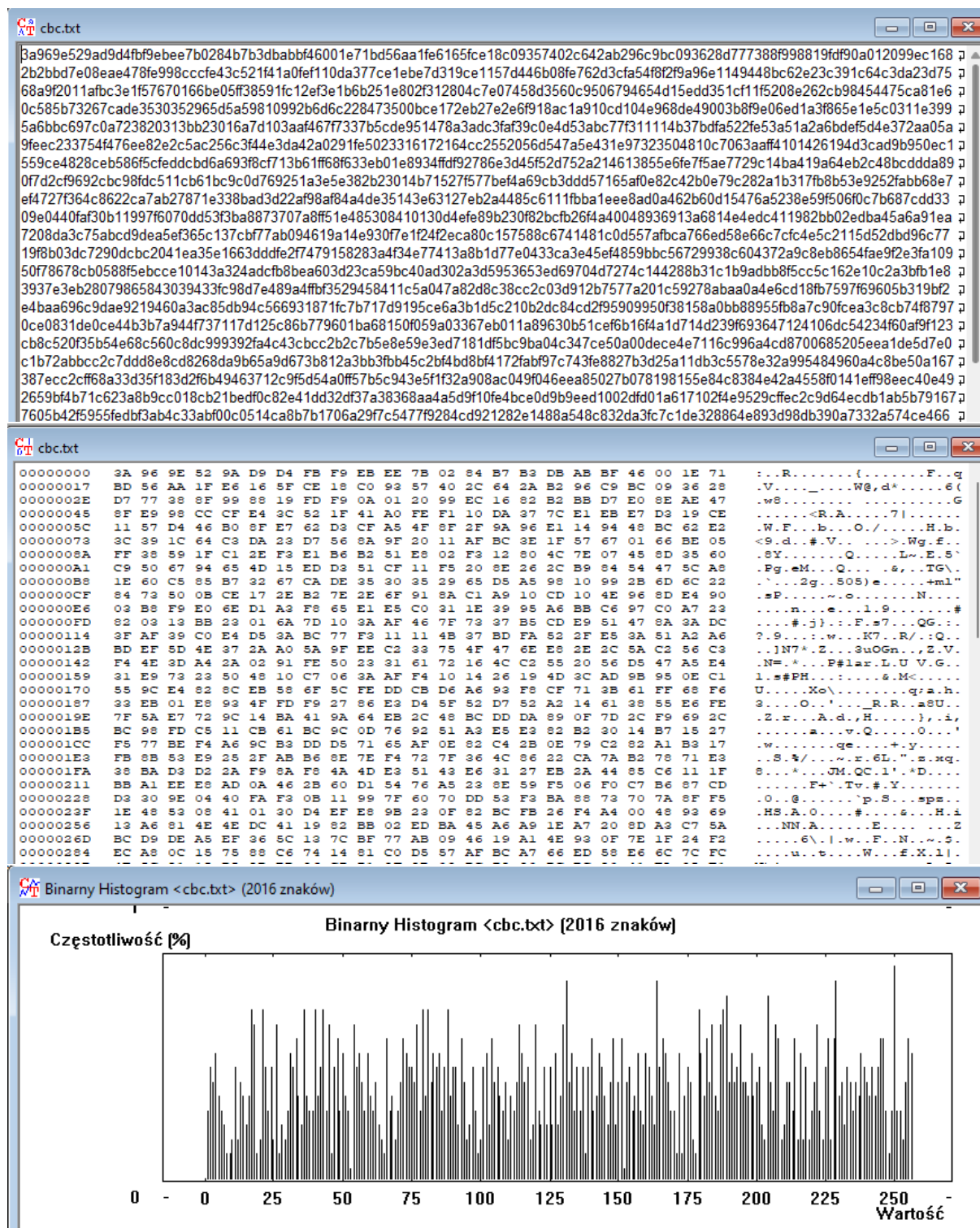


Entropia: 4,05/8 (niska), widoczne powtórzenia tekstu, histogram zdecydowanie nierównomierny.



CBC:

Entropia: 7,9/8 (wysoka), brak widocznych powtórzeń tekstu, histogram wyrównany



Entropia: 7,89/8 (wysoka), brak widocznych powtórzeń tekstu, histogram wyrównany

00000000 35 27 D4 44 51 5A 47 F7 69 32 97 47 CE D8 DD C2 9A 43 D0 F7 4C 2D 48 EC 7D 7A 28 S'.DQZG.12.G.....C.L.H.):=(  
00000001 37 52 83 13 E5 70 8D BC D8 1B 60 97 5B FD A0 E7 A0 04 A0 9B FD F7 4C 2E 50 BD 20 7R.....p.....F.  
00000002 81 F9 5D 89 AD 19 67 C4 E7 C2 72 2F 37 24 C4 E4 28 18 BE D8 0F 61 DA 87 C1 F7 E7 .....g.....7S.....  
00000003 B4 F2 85 B1 DE 78 2D 66 42 B3 9F 7D 13 17 E7 F9 71 4E 13 C1 DE 6E 94 CE 61 7D 0E .....x=EB.....qW.....a).  
00000004 41 53 85 FC 86 28 BD 1F 35 94 F1 AE AE 6F EB 43 67 01 42 28 BF AD CE 95 8F 0C .....(.5.....o.CW.B(.  
00000005 D4 1D 17 ED BA 00 70 FB 7B 2B 5B F3 AB AF E3 9E 27 54 5B FC 50 0D D1 31 C1 C7 F2 .....p.p.(+.....f.....P.....  
00000006 96 95 5F 93 F8 D1 CE 40 45 E2 83 0C F1 9B 36 59 E4 A5 14 1C F9 9B 14 86 BD 2D 2E .....@E.....6S.....  
00000007 11 0C 2B 95 CB 70 5C 63 91 9D 84 8A F1 07 C4 B0 E2 A2 47 6C D6 82 D2 E8 24 1A 17 .....+..p.c.....>G1.....S.  
00000008 C1 38 2E 0E 10 3B D3 27 77 70 93 0C 2F 48 3A 5B D0 61 E4 90 9C F9 16 E1 2E 2A 43 .....&w/p...../H.....:.....\*C  
00000009 C4 3E 6B DE CA FB 03 D2 F3 9E EE 06 D4 CD DE BB FE 15 8A 63 30 48 34 B4 CA BC 9C 0D .....>k.....2cOH4.....  
00000010 93 9F A1 E3 72 AC B8 44 70 61 DB 06 A4 A5 87 B2 36 EC A7 BD 5D 9C F6 A1 0F E2 FB .....x.....Dpa.....6.....)\n00000011 SD C5 54 1E DD 05 7C 07 90 9D 6B 1E 7C D7 38 0E 20 F2 A4 0E 3F 95 28 6D 9B E8 28 J.T.....h.....S.....(m.....  
00000012 62 54 4E 2E 2B D9 6C FE E7 80 A4 50 6B 9E 50 12 79 60 48 7F CA 67 84 36 BC B8 BIn.....h.....Fh.P.....g.....  
00000013 04 72 74 B2 BD 5F 90 24 F5 96 9E CE AE 96 61 D1 31 A7 52 50 93 CF 16 DB 6E FF 2F .....t.....a.....a1.RP.....n...../  
00000014 A5 AB D0 21 AE 69 D6 4F 30 99 09 0B BC CA A3 CF 69 12 5F BC BD AB 6E 89 4D 3A AC .....K.....1.....0O.....(.....M.....  
00000015 E7 B8 04 31 A0 F9 64 C4 00 C3 4D C9 40 EF 46 FD 17 78 0C D3 1C E5 0F 50 B8 E8 40 .....1.....d.....M.....G.....F.....P.....@  
00000016 08 0D 35 33 3F F4 37 F8 E9 C5 D6 49 5A 8D 8B 21 77 45 16 2F CB 12 9D 06 0E AD .....53?.....7.....I2.....wE...../  
00000017 F9 FD D3 22 8C 6A BE BF E8 C3 E8 07 29 5A 36 A1 F5 02 14 AD 3D AE 62 51 AF 0E B1 .....j.....j.....Z6.....=bQ.....  
00000018 37 88 2C D2 CC 87 BC 10 7A 0B 6C 71 AE 7D 9D 0C 1B 8C 6B 2A 0B 38 2D 61 E1 D4 D3 7.....\.....s.....q.....\.....h.....\*S.....  
00000019 FC 1C 04 B9 EB 91 D4 E9 56 F1 76 CB 11 7E F9 D0 FB 18 1C D2 D2 D4 0A 5B 8F A2 EE .....0.....0.....v.....\.....\.....[.....  
00000020 FE 94 13 D4 1F 4E EE 48 CE 91 69 0D F9 BE EF F1 60 0B 98 15 52 66 9C 11 CD 52 EC .....4.....H.....C.....1.....R.....E.....[.....  
00000021 F9 D8 28 40 30 76 01 CE 87 FB AB 57 FC A7 73 54 E1 28 06 07 E7 9B 44 B4 9F .....@.....@.....w.....F.....\.....\.....O.....  
00000022 BD 3D 17 92 6A 6B 6D 3C 7B D7 63 E7 9D 0C 87 EF 9B 6B 64 99 13 8B 4F 09 09 A3 45 67 .....F.....j.....m.....c.....\.....k.....d.....O.....Hg  
00000023 90 50 BC 7C 2F A5 BA D2 ED 0B 46 3F 13 B0 94 D6 5B DB 04 F3 7A 6D EA 30 D0 30 28 .....=...../.....\.....d2.....\.....m.....O.....O.....  
00000024 8E 8B C1 E7 20 55 BB B8 47 E7 32 98 9A CE CE 20 0E 6C 5C 01 E0 FA A1 E4 C8 A5 0D .....U.....G.....2.....\.....\.....\.....  
00000025 22 F5 8A 4F 85 A0 4A 27 D4 32 C6 CA EC D4 B0 77 81 19 4D 61 76 50 7E 87 6A 5E BA .....O.....U.....J.....2.....w.....M.....P.....j.....  
00000026 B2 13 84 74 FF 7E 8C DF 6D 7B D3 22 F3 6A CE D8 09 39 7A 81 BB CE E9 43 28 4C .....t.....t.....m.....\.....j.....\.....e.....C.....L.....  
00000027 44 26 EE 54 04 4A 6C 45 B1 38 0E 8A E9 56 2E 27 28 25 67 E0 B2 D0 AC 32 A5 21 De.T.....H.....S.....V...../.....\.....\.....2.....!  
00000028 B0 90 9C 94 D0 0C 08 B3 14 51 AE 60 61 S7 22 1A D2 F1 C1 0C 56 DE 33 A0 0E 8F 60 .....Q.....aW.....\.....V.....\.....

Binary Histogram <0fb.txt> [2000 znaków]

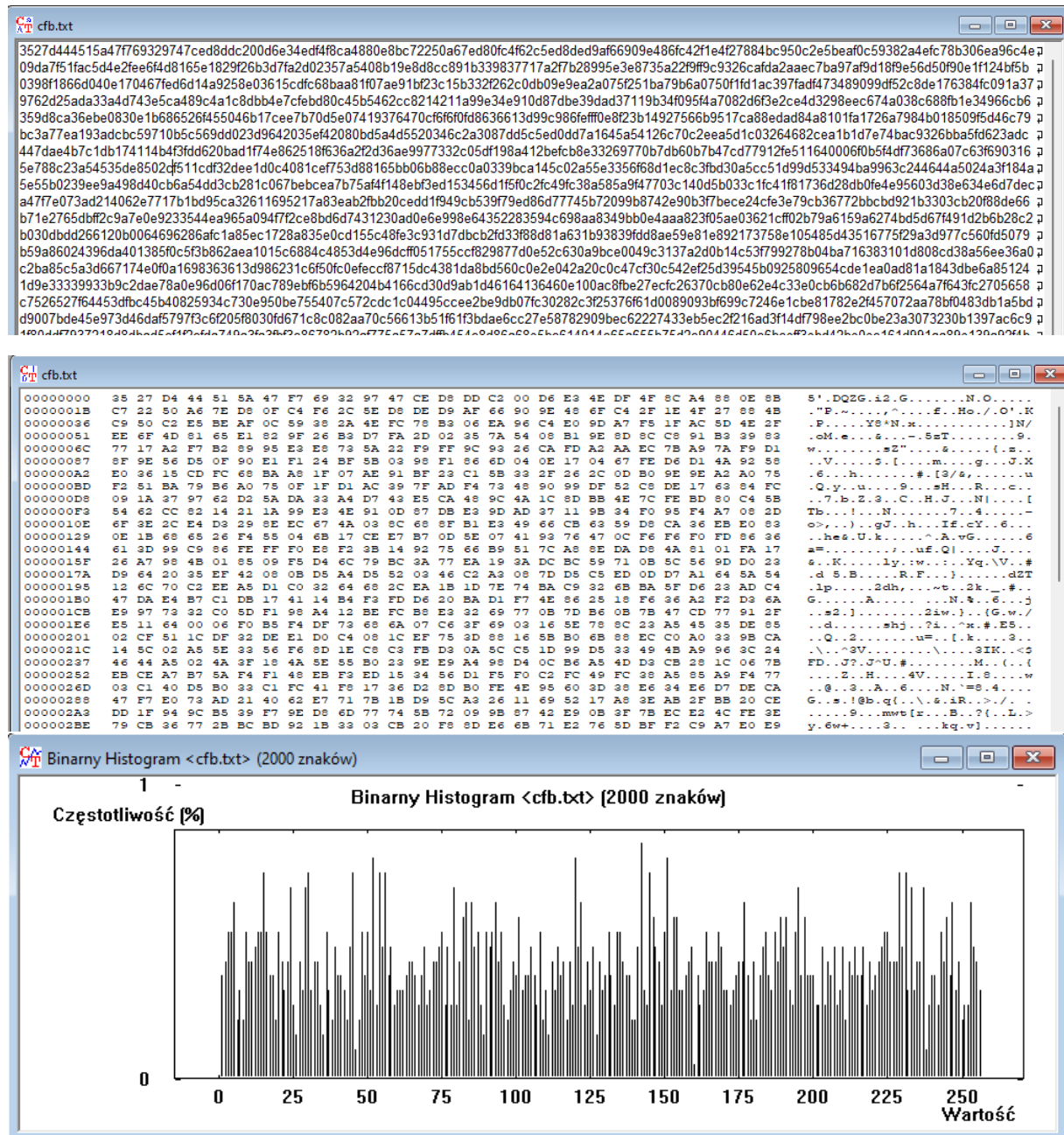
Częstotliwość [%]

0 25 50 75 100 125 150 175 200 225 250 Wartość



CFB:

Entropia: 7,91/8 (wysoka), brak widocznych powtórzeń tekstu, histogram wyrównany (początek tekstu taki sam jak OFB)



Wnioski:

Najgorzej z zaszyfrowaniem tekstu nr. 1 poradził sobie ECB, wyraźnie widać powtarzające się 128bitowe bloki tekstu. CBC, OFB i CFB są jednak dużo lepsze i nie da się zauważyć natury tekstu jednorodnego (wysoka entropia, wyrównane histogramy)

## Zadanie 2.5;

Użyłem poniższego tekstu:

### Tekst 4:

„litwo ojczyzno moja ty jestes jak zdrowie ile cie trzeba cenic ten tylko sie dowie kto cie stracil dzis pieknosc twa w calej ozdobie widze i opisuje bo tesknie po tobie panno swieta co jasnej bronisz czestochowy i w ostrej swiecisz bramie ty co grod zamkowy nowogrodzki ochraniasz z jego wiernym ludem jak mnie dziecko do zdrowia powrocilas cudem gdy od placzacej matki pod twoje opieke ofiarowany martwa podnioslem powieke i zaraz moglem pieszo do twych swiatyn progu isc za wrocone zycie podziekowac bogu”

Do analizy wybrałem algorytm AES (CBC), w wariancie 128-bitowym.

Klucz: B3 A4 09 34 3B C2 72 7B 34 C9 FF 87 96 05 3A F2

### a) zmień kilku bitów w różnych bajtach

zmiana jednego bitu w różnych bajtach blisko siebie:

```
!«•~•z%Guİ•STX•ñE•s•İja ty jestes jak zdrowie ile cie trzeba cenic ten tylko sie  
dowie kto cie stracil dzis pieknosc twa w calej ozdobie widze i opisuje bo  
tesknie po tobie panno swieta co jasnej bronisz czestochowy i w ostrej swiecisz  
bramie ty co grod zamkowy nowogrodzki ochraniasz z jego wiernym ludem jak mnie  
dziecko do zdrowia powrocilas cudem gdy od placzacej matki pod twoje opieke  
ofiarowany martwa podnioslem powieke i zaraz moglem pieszo do twych swiatyn  
progu isc za wrocone zycie podziekowac bogu
```

Zmiana jednego bitu w różnych bajtach daleko siebie:

```
[jF••];•u•Eç2k•ul•k &•u•s•ja ty jkstes jak zdrowie ile cie trzeba cenic ten tylko sie dowie kto cie stracil  
dzis pieknosc twa w calej ozdobie widze i opisuje bo tesknie po tobie panno swieta co jasnej bronisz  
czestochowy i w ostrej swiecisz bramie tİ•s•y•••²pw$•u•+*(•s•t•e•s•s•c•e•t•v••y nowogrodłkj achraniasz z jego wiernym ludem  
jak mnie dziecko do zdrowia powrocÿ{0#•s•s•K•••9&J%¹•nZóóİo°!U±••Nä•2•föi p•d twoje opiez°•c•a•n•}öjtóöä••s•t•İ•  
ÄEñxrtwa podnio~lem powieke i zaraz moglem pieszo do twych swiatyn progu isc za wrocone zycie  
podziekowac bogu
```

Dla zmienionego jednego bitu, po odszyfrowaniu zmieniony jest 16-znakowy fragment tekstu, pozostała część jest nienaruszona. Dla zmienionych kilku bitów w różnych bajtach, to gdy są one w jednej 128-bitowej grupie (blisko), to zmienia się tylko jeden 16-znakowy fragment, zaś gdy zmieniane bity są w odległości większej niż 128 bitów, to zmienianych jest kilka grup 16-znakowych (do których należą zmienione bity), pozostała część tekstu nie ulega zmianie.

## b) dodanie jednego bajtu

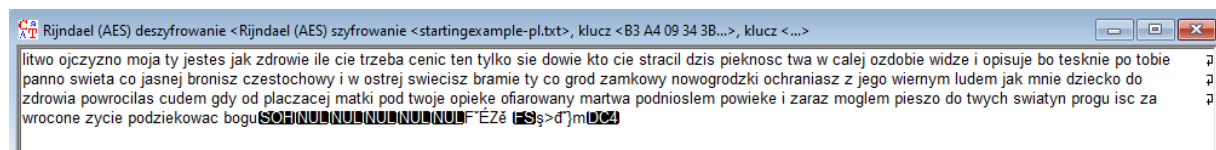
Dodanie bajtu zaburzy strukturę bloków szyfru, powodując przesunięcie reszty danych i błędne odszyfrowanie całości. Ponieważ AES wymaga, aby długość danych była wielokrotnością 16 bajtów, algorytm może zgłosić błąd związany z rozmiarem bloku (dla internetowych programów szyfrujących)

### Output

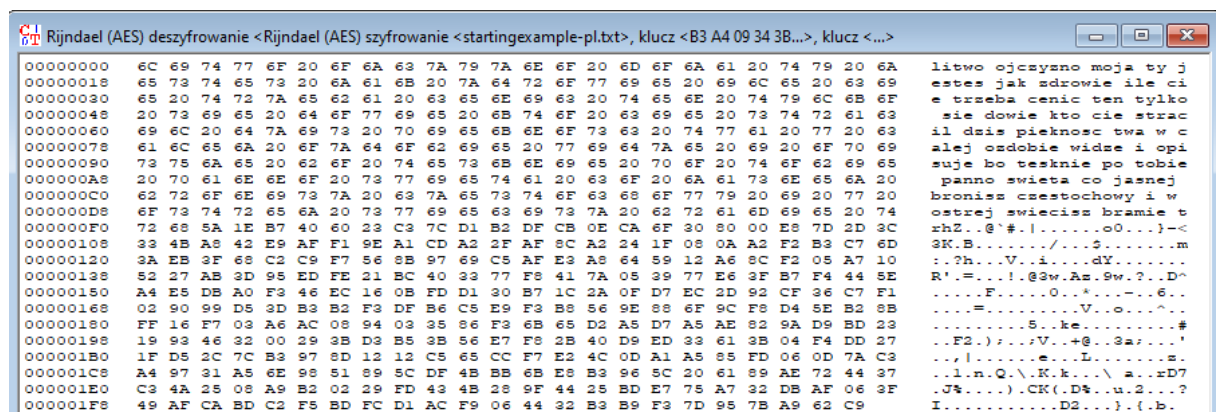
```
Unable to decrypt input with these parameters.
```

Jednak po wrzuceniu tego w Cryptool mamy kolejno:

-dla bajtu dodanego na końcu wiadomości, całość tekstu jest rozszyfrowana poprawnie, z dodanym szumem na końcu



-dla bajtu dodanego w środku wiadomości, fragment tekstu aż do zmiany jest rozszyfrowany poprawnie, a dalej jest szum ponieważ bajty w blokach są przesunięte i nie tworzą już wspólnej całości



## c) usunięcie jednego bajtu

Analogicznie do dodania bajtu z punktu b) nie działa to dla szyfrowania online.

### Output

```
Unable to decrypt input with these parameters.
```

W Cryptool również analogicznie widzimy, że tekst zachowuje się normalnie aż do momentu zmiany bajtu, po czym zamienia się w chaos z uwagi na przesunięte bajty w blokach.



```
Rijndael (AES) deszyfrowanie <Rijndael (AES) szyfrowanie <startingexample-pl.txt>, klucz <B3 A4 09 34 3B...>, klucz <...>

00000000 6C 69 74 77 6F 20 6F 6A 63 7A 79 7A 6E 6F 20 6D 6F 6A 61 20 74 79 20 6A 65 73 litwo ojczyszno moja ty jes
00000010 74 65 73 20 6A 61 6B 20 7A 64 72 6F 77 69 65 20 69 6C 65 20 63 69 65 20 74 72 tes jak zdrowie ile cie tr
00000020 7A 65 62 61 20 63 65 6E 69 63 20 74 65 6E 20 74 79 6C 6B 6F 20 73 69 65 20 64 seba cenic ten tylko sie d
00000030 6F 77 69 65 20 6B 74 6F 20 63 69 65 20 73 74 72 61 63 69 6C 20 64 7A 69 73 20 owie kto cie stracil dzis
00000040 70 69 65 6B 6E 6F 73 63 20 74 77 61 20 77 20 63 61 6C 65 6A 20 6F 7A 64 6F 62 pieknosc twa w calej osdob
00000050 69 65 20 77 69 64 7A 65 20 69 20 6F 70 69 73 75 6A 65 20 62 6F 20 74 65 73 6B ie widze i opisuje bo tesk
00000060 6E 69 65 20 70 6F 20 74 6F 62 69 65 20 70 61 6E 6E 6F 20 73 77 69 65 74 61 20 nie po tobie panno swieta
00000070 63 6F 20 6A 61 73 6E 65 6A 20 62 72 6F 6E 69 73 7A 20 63 7A 65 73 74 6F 63 6B co jasnej broniss czestoch
00000080 6F 77 79 20 69 20 77 20 6F 73 74 72 65 6A 20 73 77 69 65 63 69 73 7A 20 62 72 owy i w ostrej swiecisz br
00000090 61 6D 69 65 20 74 79 20 63 6F 20 67 72 6F 64 20 7A 61 6D 6B 6F 77 79 20 6E 6F amie ty co grod zamkowy no
00000100 77 6F 67 72 6F 64 7A 6B 69 20 6F 63 65 6D A1 49 66 88 92 A5 23 EE 8A 63 AF 94 wogrodzki oc..].If...#.c...
00000110 D6 0E E5 31 DF EC 30 EB 35 24 95 56 1F A8 AD 36 BC 07 EF F9 A7 DC 90 E2 8E 63 ...i..0.S2.V...6.....c
00000120 61 11 C6 62 FE 12 2F 83 B4 AE FA F8 8F 94 A7 20 D3 B8 FB 9D 75 0B 0D B3 2B 99 a..b../......u...+.
00000130 56 E6 19 BB 9C 62 9C FB DB BE E9 8B 01 25 78 6E 6A 64 2F BF 9C 8A 7D DF 2F F4 V...b.../.....%xnjd/...}.
00000140 61 B4 B7 9B 40 69 F3 6A E8 66 CF 56 1E D8 C2 00 3F 18 26 04 9C E3 74 74 15 F6 a...@i..j..f..V.../..$....tt...
00000150 9C 6E 9B FD C2 6A 94 3A C0 B7 4D 04 85 60 6D 53 42 72 7B 08 63 C9 F1 3E 0B AB .n...j...(.M...mSBz(.c...>..
00000160 6D 1A 7E 4A 0B C2 12 28 93 2A 63 B7 03 15 84 FC 72 C5 4B 0A 95 74 77 22 DC E4 m..wJ...(.M...mSBz(.c...>..
00000170 1B 45 E9 2F E5 8F 99 84 CC 45 14 73 2A 66 AC 8F 5E A4 B6 D2 89 C7 89 D5 29 DE .E../.....E..s*E..K..tw"..
00000180 0B 44 36 B1 1C 7F 45 17 E9 05 DF 11 51 6E A2 17 89 F7 6B 2D 26 B8 F1 3C 42 C5 ..6...E.....Qn...k-s...<B.
00000190 AB 2F 73 9E E3 74 51 02 26 8C DE 7F 19 7A 0F E8 7C ./s...tQ..6.....s...|
```

Oczywiście jeśli usuniemy jakikolwiek bajt z pierwszych 16 bajtów, wtedy cała wiadomość zostaje przesunięta i nie będziemy w stanie nic rozszyfrować:

```
Rijndael (AES) deszyfrowanie <Rijndael (AES) szyfrowanie <startingexample-pl.txt>, klucz <B3 A4 09 34 3B...>, klucz <...>

00000000 6F E7 7E E6 E7 9D A6 56 B8 D6 6A 64 2D 76 4C 3F 46 FF 52 D7 C1 B1 3F 75 B6 01 .....V..jd-wL?F.R...?u..
00000010 0D E2 FA 16 A6 20 AA 9D 1D 03 FC 31 12 39 72 53 70 CD DF 41 D4 7A CA D1 AA F6 .....l..9rSp...As...
00000020 41 5F 51 28 3B 95 C1 3B E2 76 AF 22 0C 12 81 11 F0 91 AC 95 01 EF BF E7 1F 09 A.Q(.v.../.....D.eF
00000030 FA 62 99 7C A1 B7 DB 2E 28 28 17 66 D2 FF 44 7B F1 C5 B7 F1 F3 CD 44 2E 6F 46 .b.(....((f..D{.....D.eF
00000040 C1 63 BA EB 9C 4B 89 F1 A2 8C 44 15 7B 12 7B BB F0 88 C4 89 F3 96 BC A0 EE 35 .c...K...D...{.....5
00000050 BB D4 73 31 61 F6 BF 85 5B AA 10 AB 3B 19 52 71 D3 DC 92 7E 15 7D DF FD 6C 8F .sla...{.../Rq...}.1..
00000060 7D B4 A3 60 AC 6B 38 26 9D 3E 50 17 B1 94 99 D7 C2 35 CA BE 0A D3 A2 42 BE DE }.k86>P...5.....B..
00000070 EC 6A 6E D4 68 68 61 69 74 FA 37 7B FE 29 EA 39 FE AA 1C 49 A9 5A FC 6D E1 49 .jn.hhaie.7(.).9...I.Z.m.I
00000080 27 06 3B 42 E5 36 E4 CC 05 3E E6 B5 44 F5 35 5B 70 9C F2 C8 AE E5 FE AF DE DC .rB.6...>..D.S(p...
00000090 4E 5F B9 2D D0 BF BA 59 8B C1 D0 70 FB 4B 4D 34 F5 2F A5 F4 6C 4B CA C0 6E AF 26 N...Y...p.M4/..lK..n..6
00000100 9A 8E 67 6D B3 12 B8 D9 B4 10 91 E3 43 AC 1E 8E E9 8E 47 36 06 4F A4 9A 88 1D .gm.....C.....G6.O...
00000110 69 A8 5A 6F 4B EC 30 E8 35 24 95 56 1F A8 AD 36 BC 07 EF F9 A7 DC 90 E2 8E 63 i.ZoH..0.S5.V...6.....c
00000120 61 11 C6 62 FE 12 2F 83 B4 AE FA F8 8F 94 A7 20 D3 B8 FB 9D 75 0B 0D B3 2B 99 a..b../......u...+.
00000130 56 E6 19 BB 9C 62 9C FB DB BE E9 8B 01 25 78 6E 6A 64 2F BF 9C 8A 7D DF 2F F4 V...b.../.....%xnjd/...}.
00000140 61 B4 B7 9B 40 69 F3 6A E8 66 CF 56 1E D8 C2 00 3F 18 26 04 9C E3 74 74 15 F6 a...@i..j..f..V.../..$....tt...
00000150 9C 6E 9B FD C2 6A 94 3A C0 B7 4D 04 85 60 6D 53 42 72 7B 08 63 C9 F1 3E 0B AB .n...j...(.M...mSBz(.c...>..
00000160 6D 1A 7E 4A 0B C2 12 28 93 2A 63 B7 03 15 84 FC 72 C5 4B 0A 95 74 77 22 DC E4 m..wJ...(.M...mSBz(.c...>..
00000170 1B 45 E9 2F E5 8F 99 84 CC 45 14 73 2A 66 AC 8F 5E A4 B6 D2 89 C7 89 D5 29 DE .E../.....E..s*E..K..tw"..
00000180 0B 44 36 B1 1C 7F 45 17 E9 05 DF 11 51 6E A2 17 89 F7 6B 2D 26 B8 F1 3C 42 C5 ..6...E.....Qn...k-s...<B.
00000190 AB 2F 73 9E E3 74 51 02 26 8C DE 7F 19 7A 0F E8 7C ./s...tQ..6.....s...|
```

Wniosek: Usunięcie i dodanie bajtów ma podobny wpływ i charakteryzuje się głównie normalnym działaniem aż do miejsca zmiany, po którym to ze względu na przesunięcie w lewo lub prawo o jeden wszystkich pozostałych bajtów, dalsze rozszyfrowanie bloków bajtów staje się bezskuteczne.

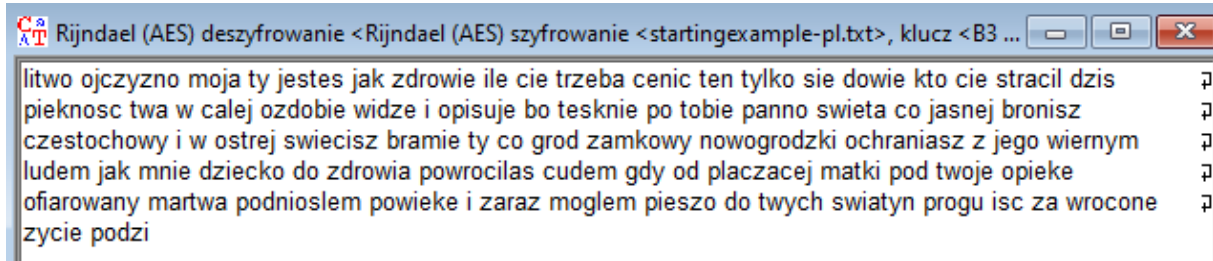
d) dodanie/usunięcie fragmentu tekstu równego długości bloku algorytmu

Dla usuniętego środkowego fragmentu o długości 16 bajtów (długość bloku tego algorytmu) możemy zauważyć, że zmianie uległy jedynie znaki, znajdujące się pomiędzy dwoma blokami algorytmu (oprócz zmienionych znaków sam tekst jest krótszy o 16 znaków). Reszta tekstu pozostała bez zmian.

```
litwo ojczyszno moja ty jestes jak zdrowie ile cie trzeba ce
nic ten tylko sie dowie kto cie stracil dzis pieknosc twa w
calej osdobie widze i opisuje bo tesknie po tobie panno sw
ieta co jasnej broniss czestochowy i w ostrej swiecisz bram
ie ty co grod zamkowy nowogrodzki ochraniass z jego wiernym
ludem jaEFX.)m.s.u..9...{.$...nudem gdy od placzacej matki
pod twoje opieke ofiarowany martwa podnioslem powieke i sar
az moglem piezzo do twych swiatyn progu isc za wroczone zyci
e podziekowac bogu
```

Dla usuniętego fragmentu o długości 16 bajtów od końca tekstu tajnego, możemy zauważyć, że tekst nie uległ zmianie, oprócz faktu że informacja o ostatnich 16 znakach została stracona. Analogicznie dla każdej innej konfiguracji, w której usuwamy przedział 16 bajtów, które definiują w całości pojedynczy

blok algorytmu.



Dla bajtów dodanych:

(na końcu tekstu tajnego)

```
litwo ojczyzno moja ty jes
tes jak zdrowie ile cie tr
seba cenic ten tylko sie d
owie kto cie stracil dzis
pieknosc twa w calej ozdob
ie widze i opisuje bo tesk
nie po tobie panno swieta
co jasnej bronisz czestoch
owy i w ostrej swiecisz br
amie ty co grod zamkowy no
wogrodzki ochraniaasz z jeg
o wiernym ludem jak mnie d
ziecko do zdrowia powrocil
as cudem gdy od placzacej
matki pod twoje opieke ofi
arowany martwa podnioslem
powieke i zaraz moglem pie
szo do twych swiatyn progu
isc za wroczone zycie podzi
ekowac bogu.....)....
..{e..0
```

(w środku tekstu tajnego)

```
litwo ojczyzno moja ty jestes jak
zdrowie ile cie trzeba cenic ten
tylko sie dowie kto cie stracil
dzis pieknosc twa w calej ozdobie
widze i opisuje bo tesknie po to
bie panno swieta co jasnej bronis
z czestochowy i w ostrej s....7.0
....Q.J..P...T.}...9...M....co gro
d zamkowy nowogrodzki ochraniaasz
z jego wiernym ludem jak mnie dzi
ecko do zdrowia powrocilas cudem
gdy od placzacej matki pod twoje
opieke ofiarowany martwa podniosl
em powieke i zaraz moglem pieszo
do twych swiatyn progu isc za wro
cone zycie podziekowac bogu■
```

Dla tych fragmentów 16 bajtowych, które zostały dodane w odpowiednich blokach (modulo 16 bajtów, tak aby nie zaburzać rozszyfrowania innych bloków) - tekst pozostaje niezmienny z dodanym fragmentem „szumu” na jeden dodatkowy blok. Jeśli jednak dodamy nasz 16 bajtowy blok, pomiędzy dwa inne bloki, „zaszumione” zostaną oba te bloki, co będzie skutkowało podwójną ilością znaków które się w nic nie układają.

## Podsumowanie:

**a)** Dla trybu ECB zmiana bitu wpływa jedynie na dany blok, ponieważ każdy blok jest szyfrowany niezależnie. Dla trybu CBC zmiana bitu wpływa zarówno na dany blok w którym znajduje się bit jak blok kolejny, ponieważ CBC stosuje XORowanie kolejnych bloków ze sobą. Dla OFB i CFB zmiana bitu wpływa tylko na odpowiednią część odszyfrowanego strumienia, ale wpływa na kolejne bloki.

**b) c)** Dla trybów ECB i CBC dodanie/usunięcie bajtów powoduje przesunięcie bloków, co uniemożliwia prawidłowe odszyfrowanie od miejsca zmiany. Oba tryby są podobnie wrażliwe na takie zmiany. Dla trybów OFB i CFB dodanie/usunięcie bajtów również prowadzi do zniekształceń w reszcie tekstu, ale z powodu zaburzenia synchroniczności między szyfrowanym tekstem a generowanym strumieniem.

**d)** Dla trybu ECB dodanie/usunięcie bloku danych nie wpływa na resztę szyfrogramu, ponieważ bloki są niezależne. Dla trybu CBC dodanie/usunięcie całego bloku wpływa na resztę szyfru (ale tylko te po, a nie przed momentem zmiany), ponieważ dalsze bloki nie są niezależne od bloków poprzednich. Dla trybów OFB i CFB dodanie/usunięcie bloku danych zaburza synchronizację i powodują błędy w dalszej części strumienia danych

## Zadanie 2.6;

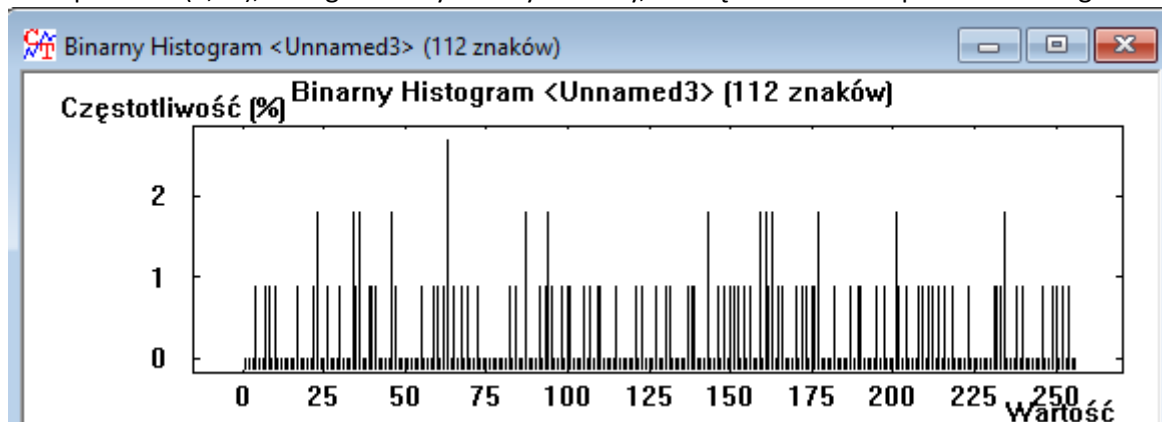
a) Gołym okiem widać, że długość bloku wynosi 16 bajtów, ponieważ ten sam fragment tekstu jest powtarzany co okres 16 bajtów. Wynika z tego, że bloki są o długości 8 znaków. Przeprowadzone wcześniej eksperymenty wskazują na to, że mamy tu do czynienia z przypadkiem szyfrowania tekstu jednorodnego trybem ECB.

b) W tym tekście także widać cykliczne powtórzenia ciągów bajtów, jednak tym razem są one co 32 bajty. Wynika z tego, że bloki są o długości 16 znaków. Przeprowadzone wcześniej eksperymenty wskazują na to, że ponownie mamy tu do czynienia z przypadkiem szyfrowania tekstu jednorodnego trybem ECB.

c) W tekście brakuje jednego bitu w ostatnim jego bloku (ostatnie kilka znaków jest napisane inną czcionką co sugeruje jakąś zmianę na ostatni moment). Z tego powodu do poniższej analizy została wstawiona na sam koniec wartość '0', tak aby ostatni bajt miał postać „50” w HEX.

```
6a239123a19647032e3e637ab0b02d56
26b593edf8445c07e9e79e408ec89e56
16d9e89bef28c92d61ab6d82d39921ac
8109e9d03dafd2fd6416a0ba97513ede
0615a2688a785319d75d95a4a9aed55e
e6c221cfa51d7e89f93a5dfbc4a2bc22
368ecba0bd42c8f56c2710723b883e5
```

Entropia niska (6,53), histogram dosyć niewyrównany, ale ciężko stwierdzić przez krótki fragment.



Może to sugerować użycie jakiegoś „słabszego” algorytmu np. IDEA, oraz użycie trybu ECB lub CBC. Gołym okiem nie da się zauważyć żadnych cykliczności, więc trudno wskazać długość bloku algorytmu. Na końcowy kształt tekstu tajnego mogło również wpłynąć zróżnicowanie tekstu jawnego.

## Pytanie 2.7;

Dla trybów CBC, OFB oraz CFB entropia jest na bardzo wysokim poziomie (7,89-7,91/8), natomiast tryb ECB charakteryzuje się bardzo niską entropią (4,05/8). Kryptogram trybu ECB posiada widoczne, cykliczne powtórzenia ciągów liter, natomiast kryptogramy CBC, OFB i CFB nie posiadają już takiej cechy.

### Pytanie 2.8;

Dla trybu ECB oraz CBC niepoprawnie rozszyfrowane są jedynie specyficzne bloki 16 lub 32 znaków, w których zostały przekłamane bity. Pozostała część tekstu nie uległa zmianie.

Dla trybu OFB niepoprawnie rozszyfrowywane są jedynie znaki, w których znajdują się przekłamane bity.

Dla trybu CFB niepoprawnie rozszyfrowane są znaki z przekłamanym bitem oraz blok tekstu w odległości 16 bajtów.

### Pytanie 2.9;

Dla trybu CBC wiadomość zostaje „zaszumiona” od momentu w którym nastąpiła strata (a konkretniej w bloku, w którym nastąpiło pierwsze przesunięcie pozostałych bajtów).

Dla trybu ECB utrata wiadomości zaczyna się od bloku, w którym nastąpiło usunięcie pierwszego bajta, do wystąpienia bloku gdzie został usunięty ostatni bajt.

Dla trybów OFB i CFB utrata wiadomości zaczyna się od pierwszego usuniętego bajta i trwa do końca wiadomości, ponieważ reszta bajtów zostaje przesunięta.

### Pytanie 2.10;

ECB jest prostszy i szybszy w deszyfrowaniu, ale mniej bezpieczny, ponieważ ujawnia wzorce w jednorodnych i długich tekstach. Jest odporny na zakłócenia, więc dobrze nadaje się do szybkich transmisji danych, choć lepiej sprawdza się z tekstami zróżnicowanymi.

CBC zapewnia większe bezpieczeństwo i ukrywa naturę tekstu, ale jest bardziej podatny na błędy, ponieważ zakłócenie jednego bitu może zepsuć cały szyfrogram. Nadaje się do transmisji, które wymagają bezpieczeństwa (np. VPN) albo szyfrowania poufnych danych (np. wiadomości email) oraz do szyfrowania plików, gdzie bezpieczeństwo jest kluczowe, a zakłócenia transmisji są mało prawdopodobne.

### Pytanie 2.11;

Dla ECB jest możliwe, ponieważ każdy blok danych jest szyfrowany oraz deszyfrowany niezależnie od innych. Bloki więc, można podzielić na kilka części, które będą przetwarzane równolegle na różnych komputerach, a wynik końcowy będzie taki sam, jak w przypadku przetwarzania na jednym komputerze.

Dla CBC równoległe szyfrowanie i deszyfrowanie nie jest możliwe. Szyfrowanie nie jest możliwe, ponieważ każdy blok jest ściśle zależny od poprzedniego (z powodu XORa z poprzednim szyfrogramem). Deszyfrowanie również nie jest możliwe, ponieważ konieczne jest uprzednie odszyfrowanie poprzedniego bloku, więc również wymaga sekwencyjności.

Dla OFB i CFB nie jest możliwe, ponieważ każdy blok zależy od poprzedniego (przy OFB generowany ciąg musi być sekwencyjny, a w CFB sprzężenie zwrotne wymaga przetwarzania poprzednich bloków).

### Wniosek:

ECB to jedyny tryb, w którym szyfrowanie i deszyfrowanie można prowadzić równolegle.