Politechnika Wrocławska, Informatyka Stosowana

Asymetryczne algorytmy szyfrowania

Cyberbezpieczeństwo, Laboratorium nr.4 - raport

Autor: Aleksander Stepaniuk
Nr. Indeksu: 272644

Zad 1. Własności algorytmów asymetrycznych

Teksty których użyłem do analizy kolejnych algorytmów:

Tekst 1:

Litera „n” powtórzona 2000 razy.

Tekst 2:

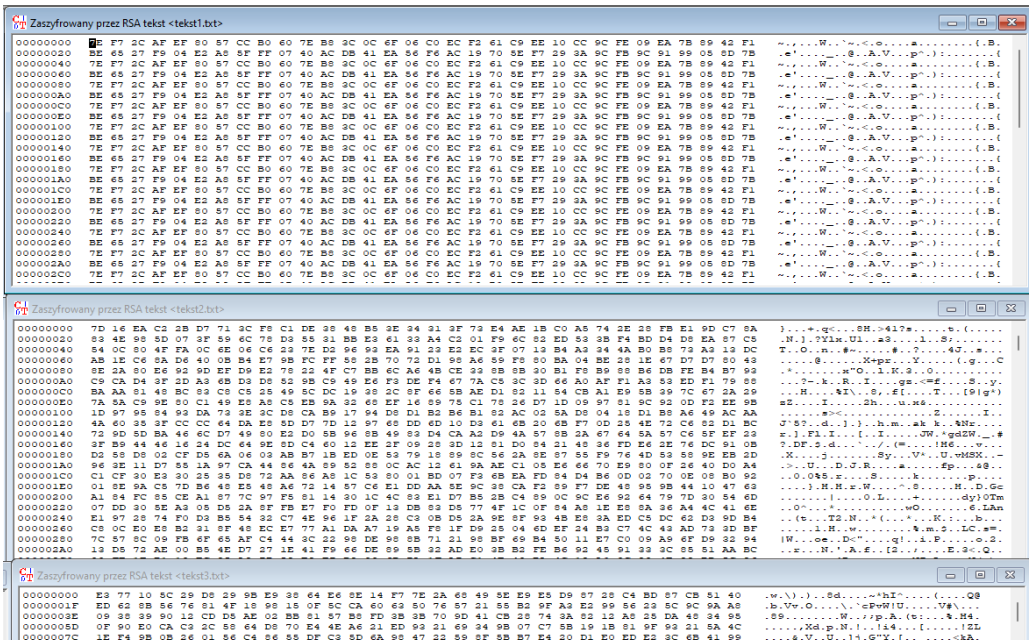
Tekst „i potem jeszcze do biedronki na lody” powtórzony 1000 razy.

Tekst 3:

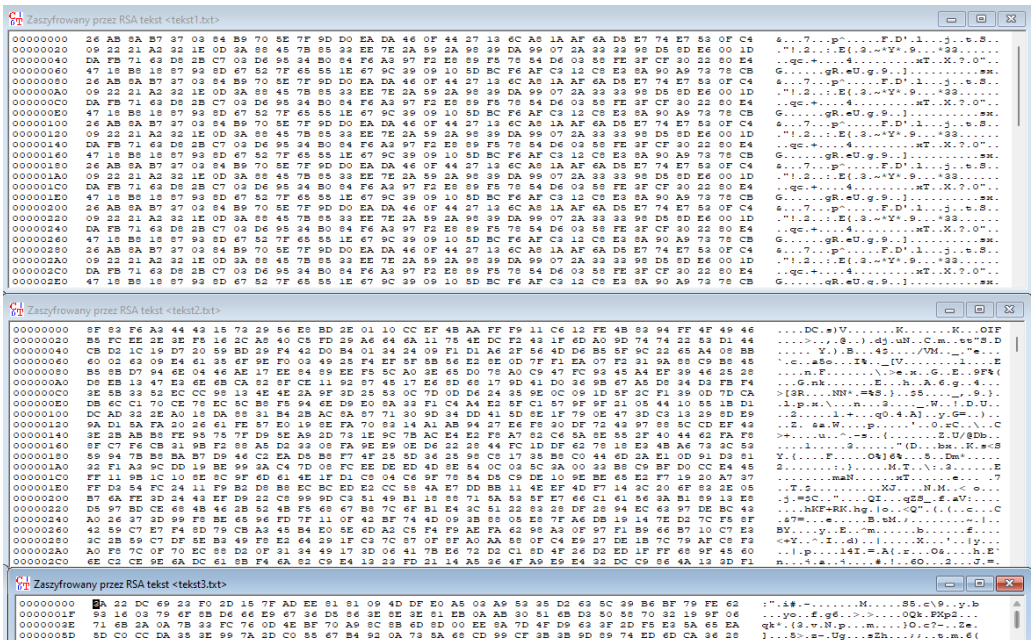
5000 pierwszych znaków pana Tadeusza („litwo ojczyzno moja ty jestes jak zdrowie ile cie trzeba cenic ten tylko sie dowie kto cie stracil…”)

Zadanie 1.1-1.3;

Dla RSA, klucz 512 bitowy:



Dla RSA, klucz 1024 bitowy:



Dla RSA, klucz 2048 bitowy:

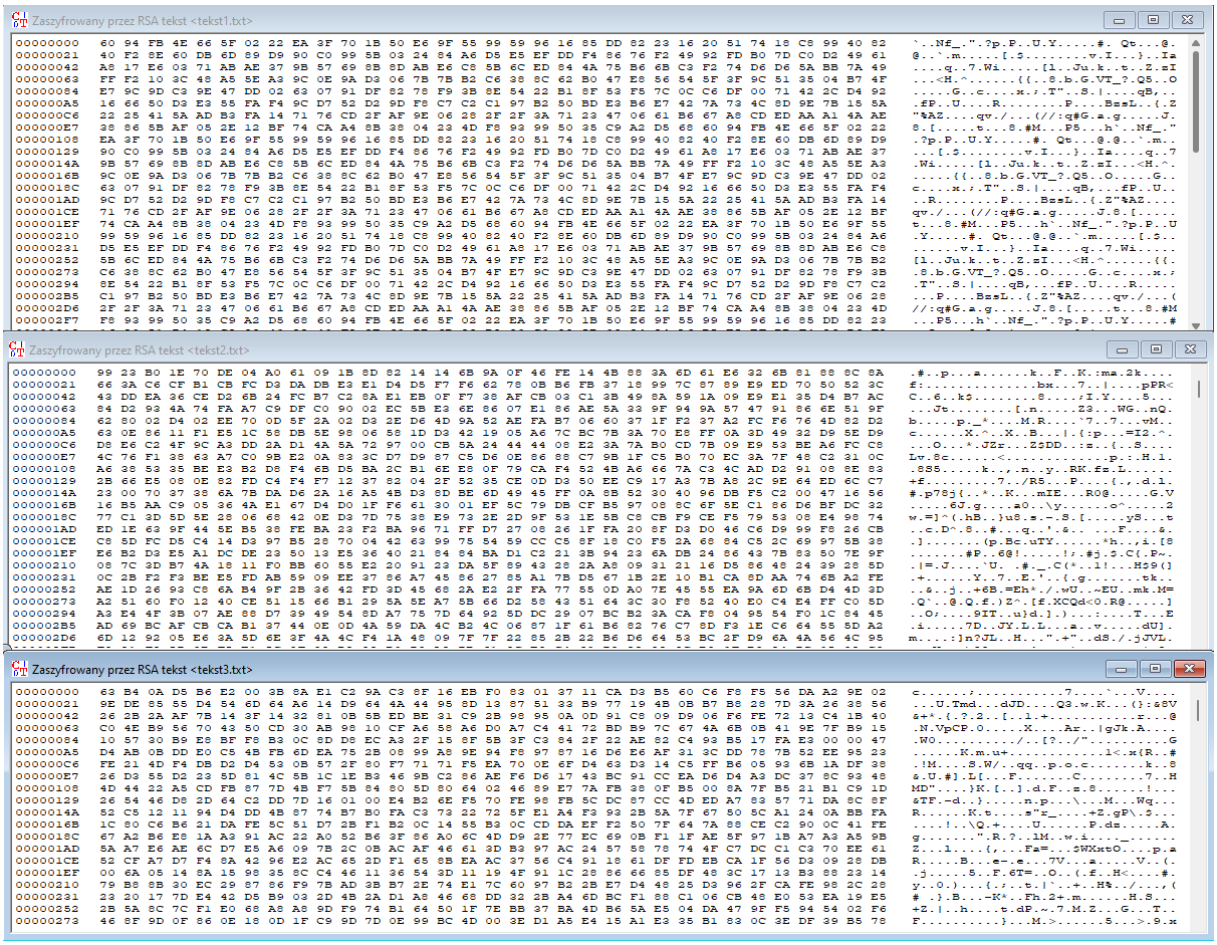
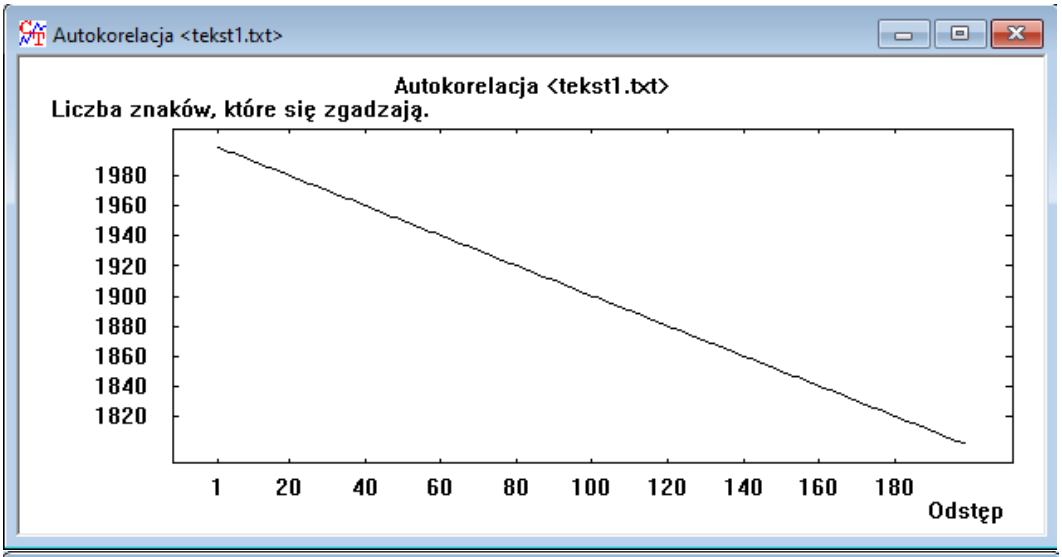


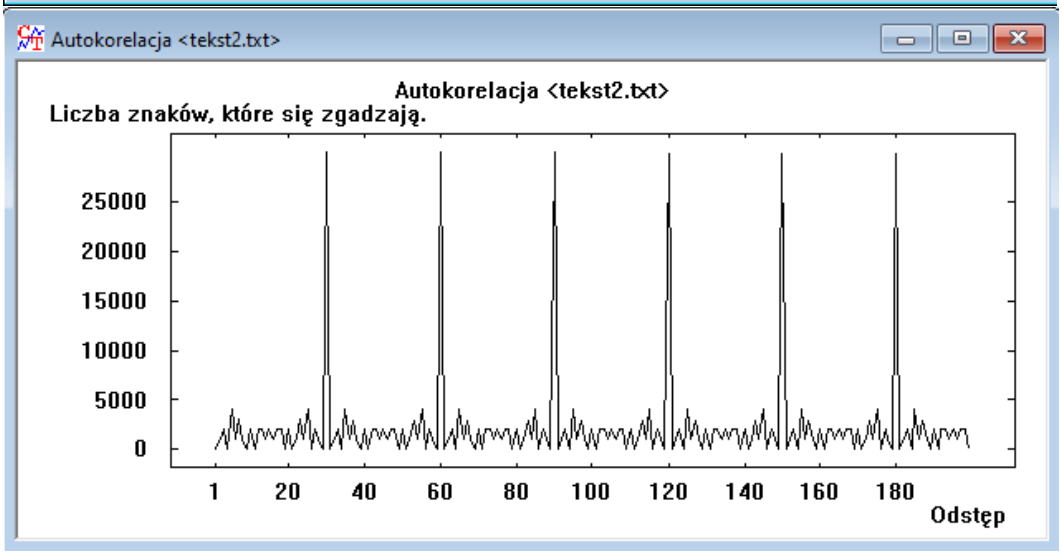
Tabela entropii: (TJ przeskalowany do wspólnego mianownika 8,00)

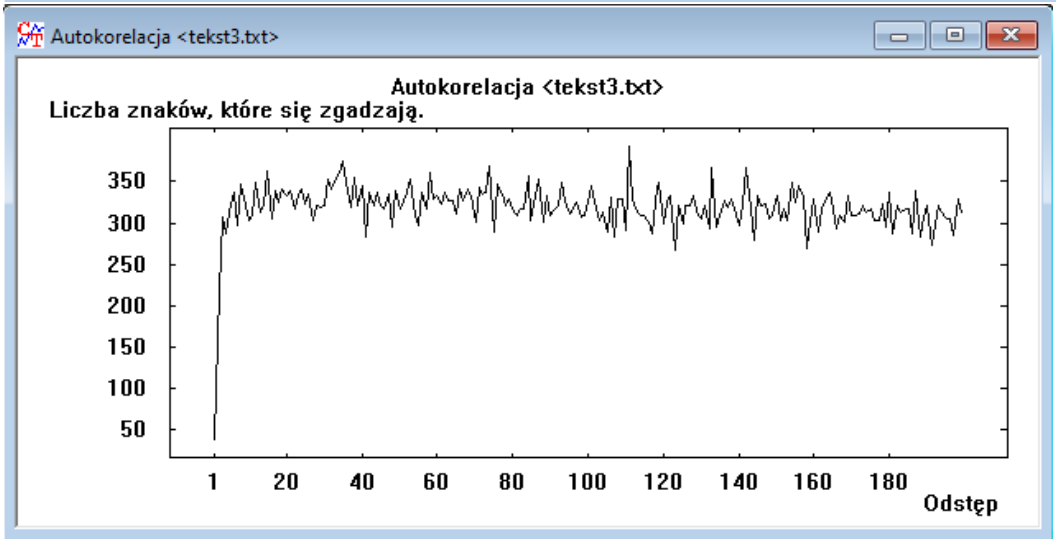
Entropia tekstu jawnego i zaszyfrowanego w zależności od długości klucza				
	TJ	TT 512bit	TT 1024bit	TT 2048bit
Tekst 1	0,00/4,70 (0,00/8,00)	5,90/8,00	6,77/8,00	7,37/8,00
Tekst 2	3,92/4,70 (6,67/8,00)	7,93/8,00	7,96/8,00	7,98/8,00
Tekst 3	4,25/4,70 (7,23/8,00)	7,97/8,00	7,97/8,00	7,97/8,00

Wykresy autokorelacji:

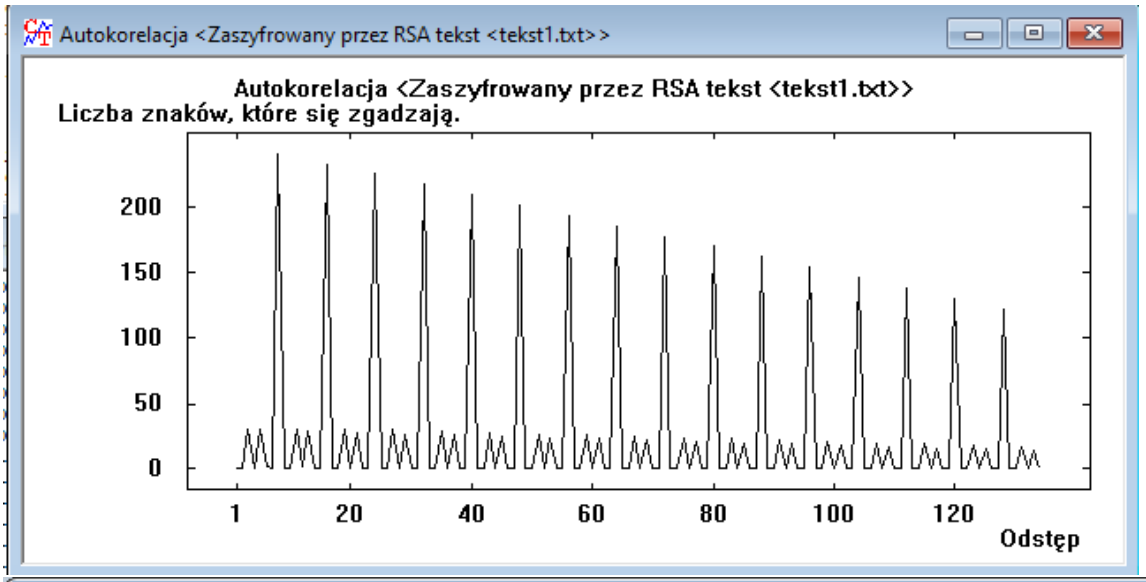
a) Tekst jawny:

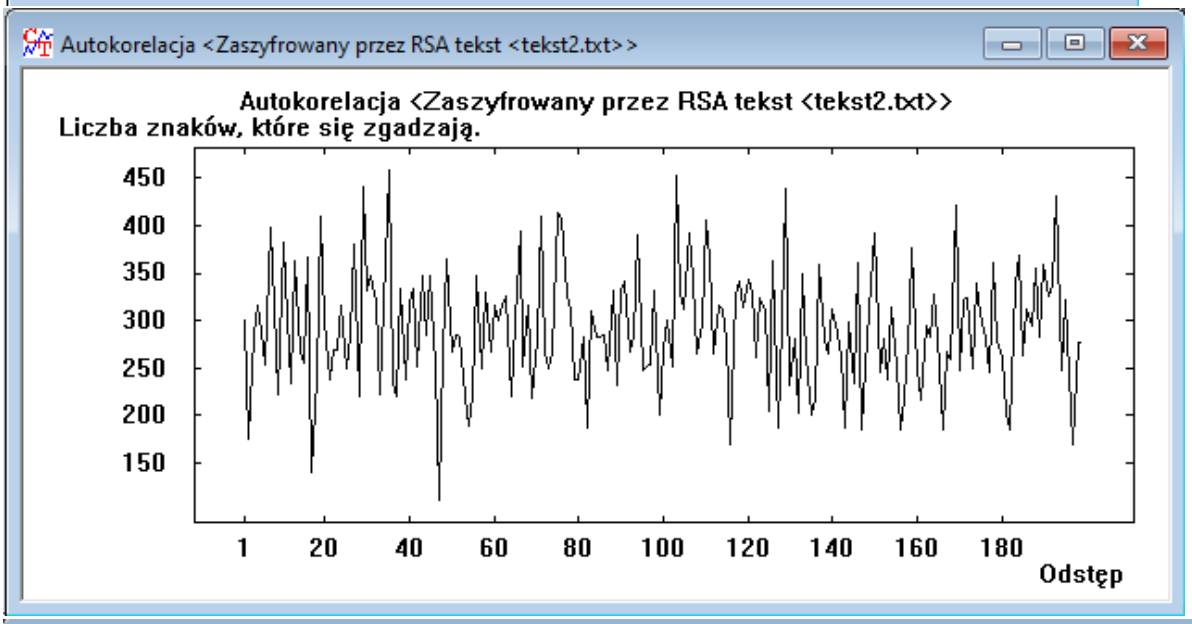


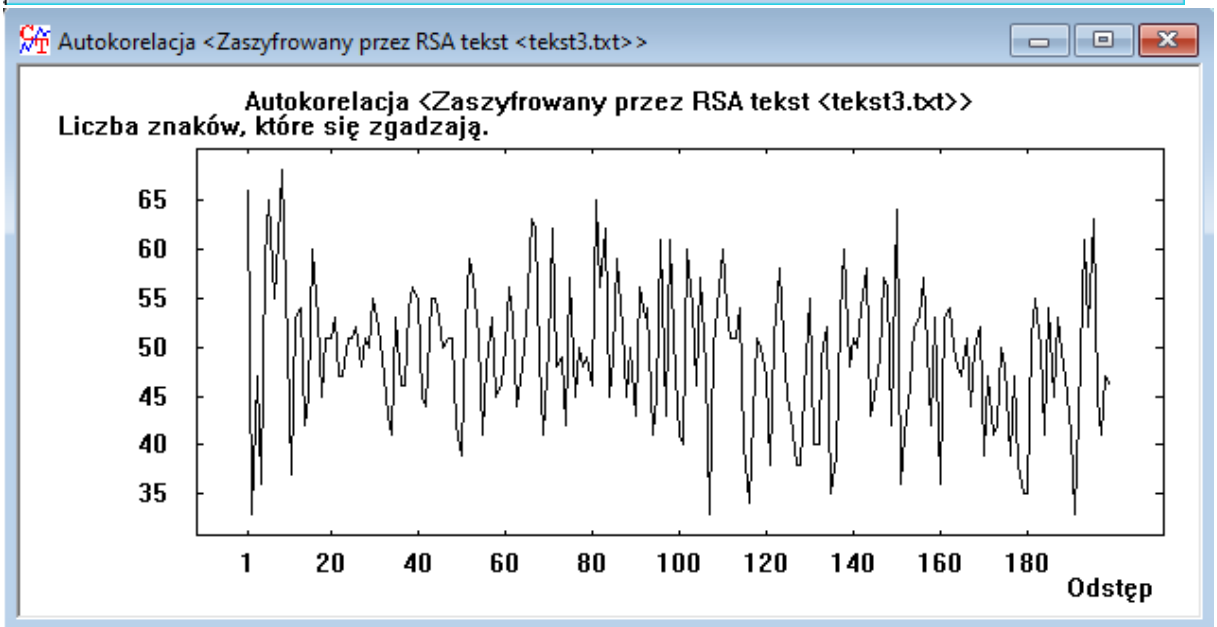




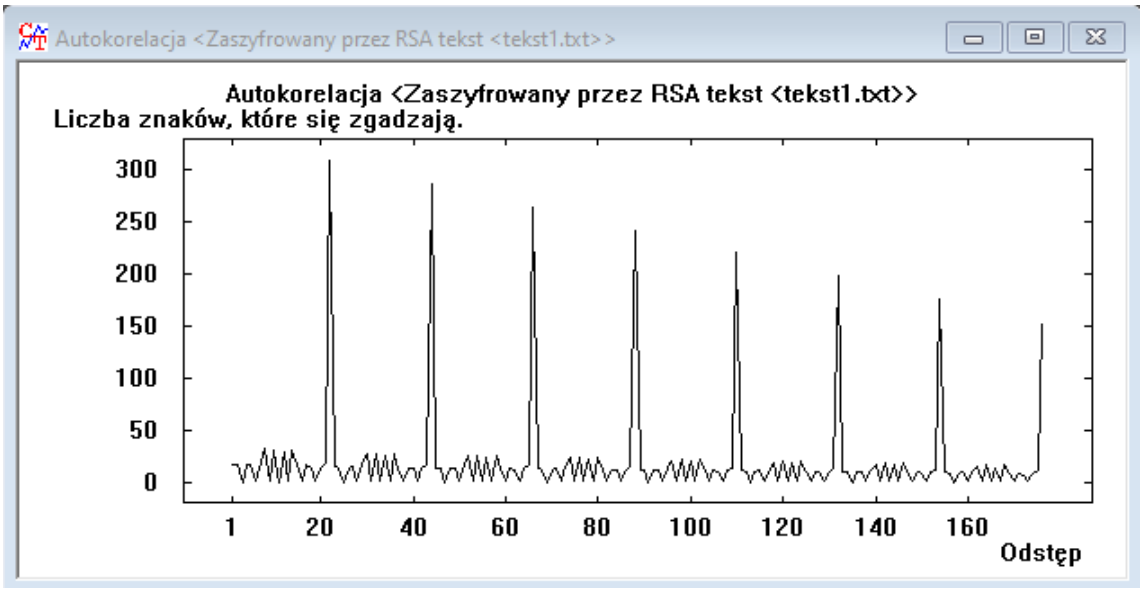
b) Tekst tajny 512 bit:

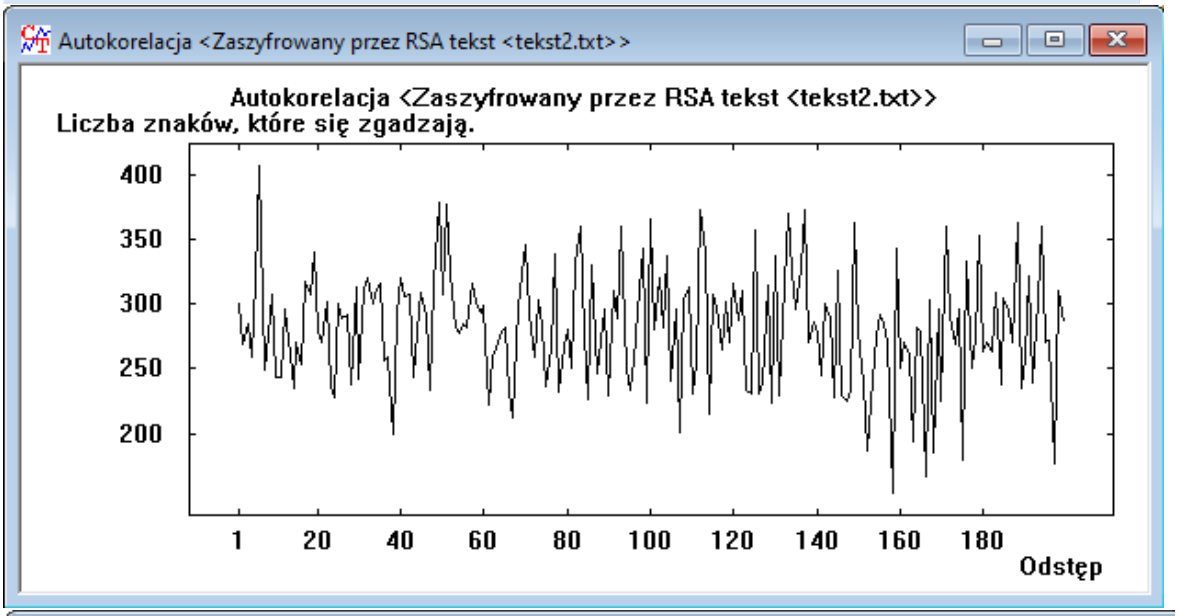


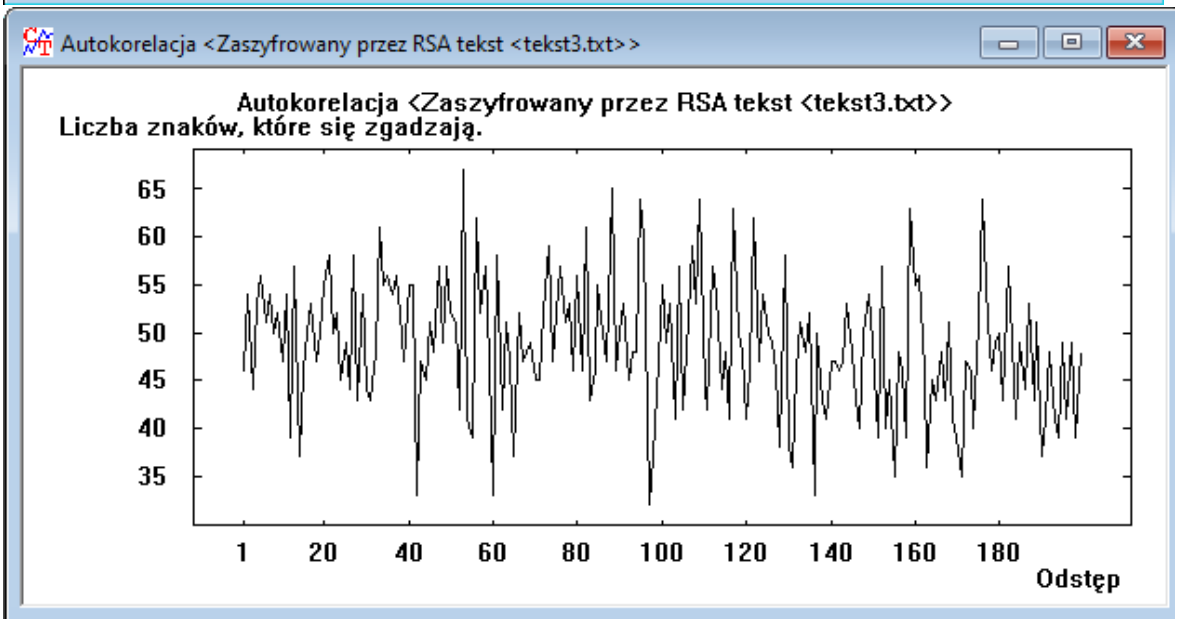




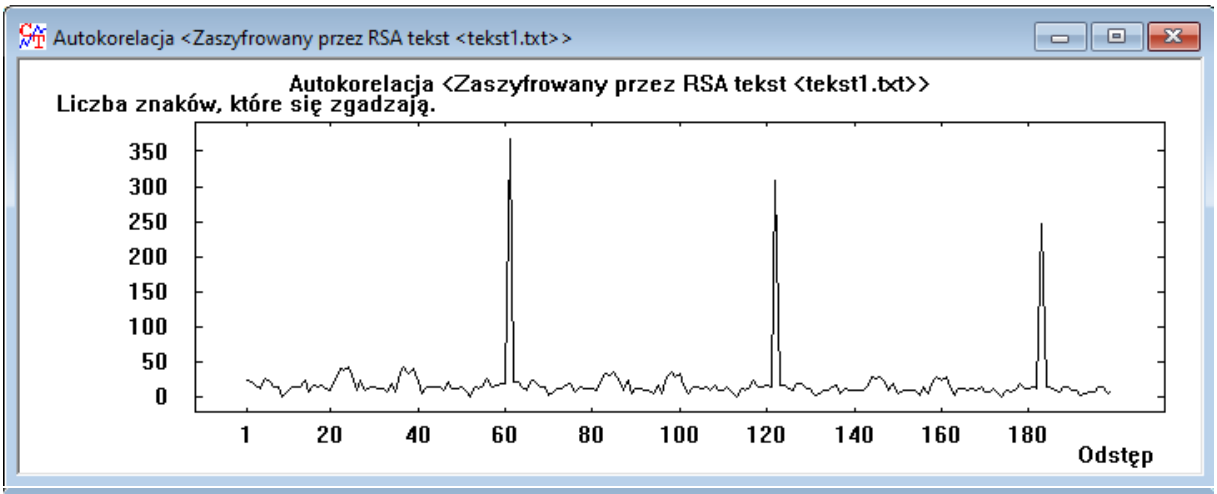
c) Tekst tajny 1024 bit:

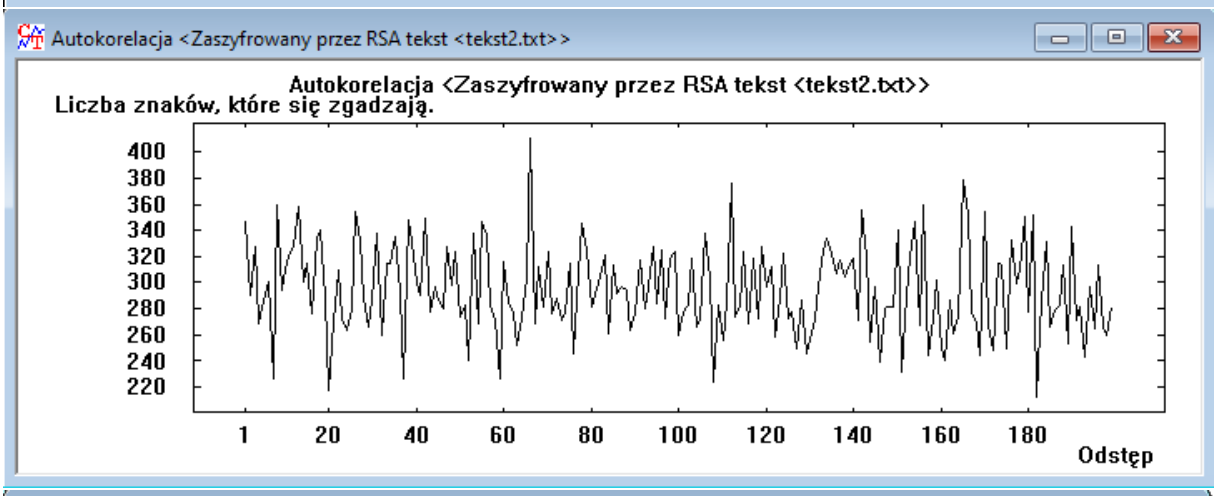


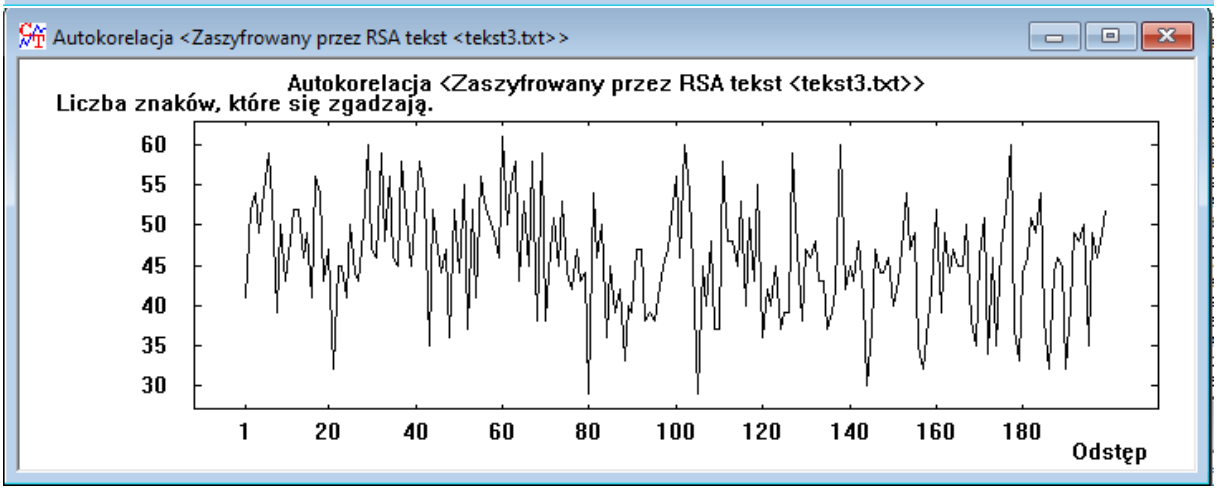




d) Tekst tajny 2048 bit:

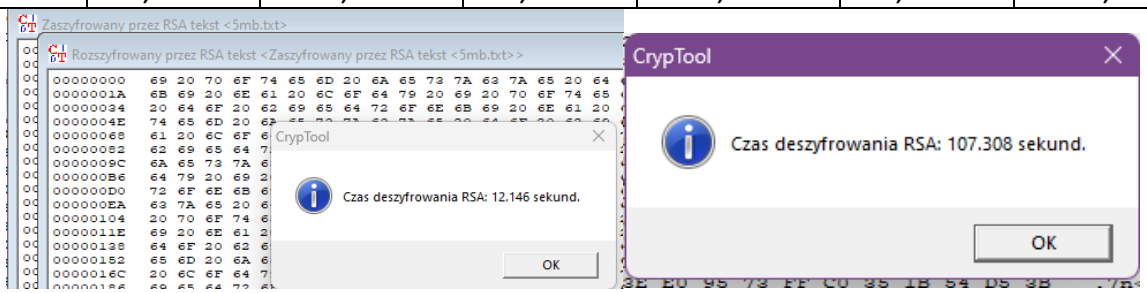






Zadanie 1.4;

Czas szyfrowania i deszyfrowania RSA zależnie od rozmiaru pliku i długości klucza						
	512 bitów		1024 bitów		2048 bitów	
	Szyfrowanie	Deszyfrowanie	Szyfrowanie	Deszyfrowanie	Szyfrowanie	Deszyfrowanie
1mb	0,184s	2,371s	0,270s	6,780s	0,542s	21,234s
2mb	0,363s	4,746s	0,547s	13,471s	1,083s	42,216s
5mb	0,928s	12,146s	1,382s	34,445s	2,809s	107,308s



Zadanie 1.5;

Czas szyfrowania i deszyfrowania AES zależnie od rozmiaru pliku i długości klucza						
	128 bitów		192 bitów		256 bitów	
	Szyfrowanie	Deszyfrowanie	Szyfrowanie	Deszyfrowanie	Szyfrowanie	Deszyfrowanie
1mb	<0,1s	<0,2s	<0,1s	<0,3s	<0,1s	<0,4s
2mb	<0,1s	<0,3s	<0,1s	<0,4s	<0,1s	<0,7s
5mb	<0,1s	<0,8s	<0,1s	<0,9s	<0,1s	<1,0s

Zadanie 1.6;

Eksperymenty przeprowadzone dla Tekstu nr. 3 oraz długości klucza kolejno 512, 1024, 2048 bitów:

a) Zmiana wartości 1 bajtu

512 bitów

```
...l....A.Rx..B.}.5.&...w
.......jqF9C.....@O.Loi....
.i/.Pl.7.Y.T.ten tylko si
e dowie kto cie stracil d
zis pieknosc twa w calej
ozdobie widze i opisuje b
o tesknie po tobie panno
swieta co jasnej broniss
czestochowy i w ostrej sw
iecisz bramie ty co grod
zamkowy nowogrodzki ochra
niasz z jego wiernym lude
m jak mnie dziecko do zdr
owia powrocilas cudem gdy
 od placzacej matki pod t
woje opieke ofiarowany ma
rtwa podnioslem powieke i
 zaraz moglem pieszo do t
wych swiatyn progu isc za
 wrocone zycie podziekowa
c bogu tak nas powrociss
cudem na ojczyzny lono ty
```

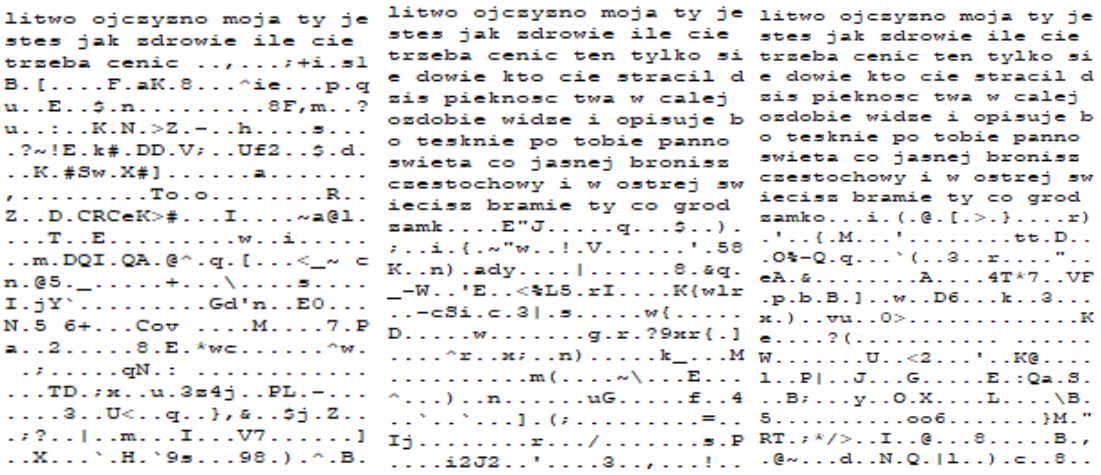
1024 bitów

```
litwo ojczyzno moja ty je
stes jak zdrowie ile cie
trzeba cenic ten tylko si
e dowie kto cie stracil d
zis pieknosc twa w calej
ozdobie widze i opisuje b
o tesknie po tobie panno
swieta co jasnej broniss
czestochowy i w ostrej sw
iecisz bramie ty co grod
zamk....`*..P.....CZI...?
.2....e\...M@...;o:(e..j.
.h....y..h..I&.l.@..s.:..
Z}.e5..A.3Z-bw.P..4{..v6.
U..8....OL..}.I...LJ.;...
n7+../pieke ofiarowany ma
rtwa podnioslem powieke i
 zaraz moglem pieszo do t
wych swiatyn progu isc za
 wrocone zycie podziekowa
c bogu tak nas powrociss
cudem na ojczyzny lono ty
```

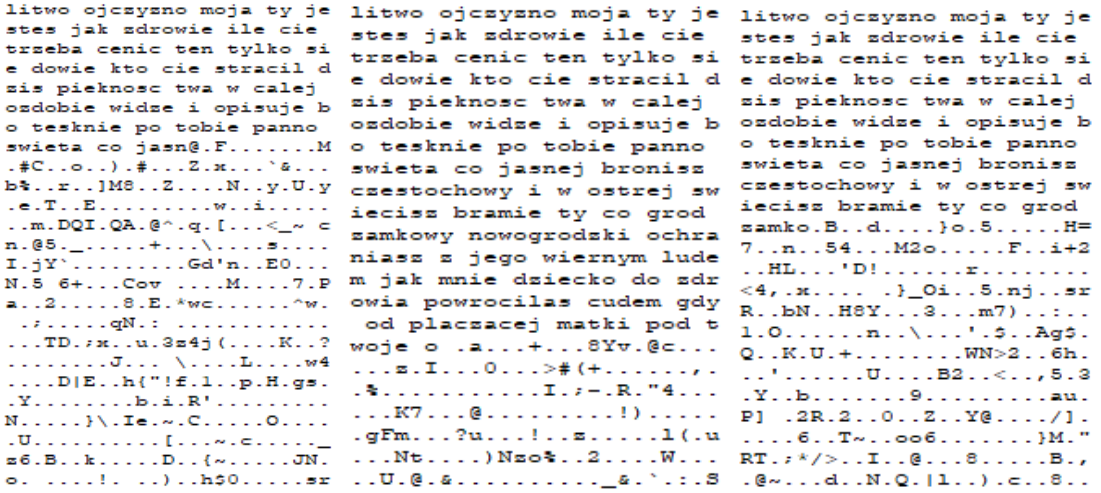
2048 bitów

```
.S.~.#.#...7Qs..gB..\.%]q
s....C.,.:!.......=.0Nog[
.......N.....Y..W....m".=
a.......ZA..W+!.TJ..I..?..
.......P.U.h.......!.Q>n..
4..E.(..$.....<....-3..y.
yHM...y......{..'ZD./...a
.M.Y.i....?.`..D..f......
.. .....l......B..G.o@..m
4...>x59.g......K0\.6.J.D
..N#.wy nowogrodzki ochra
niasz z jego wiernym lude
m jak mnie dziecko do zdr
owia powrocilas cudem gdy
 od placzacej matki pod t
woje opieke ofiarowany ma
rtwa podnioslem powieke i
 zaraz moglem pieszo do t
wych swiatyn progu isc za
 wrocone zycie podziekowa
c bogu tak nas powrociss
cudem na ojczyzny lono ty
```

b) usunięcie 1 bajtu

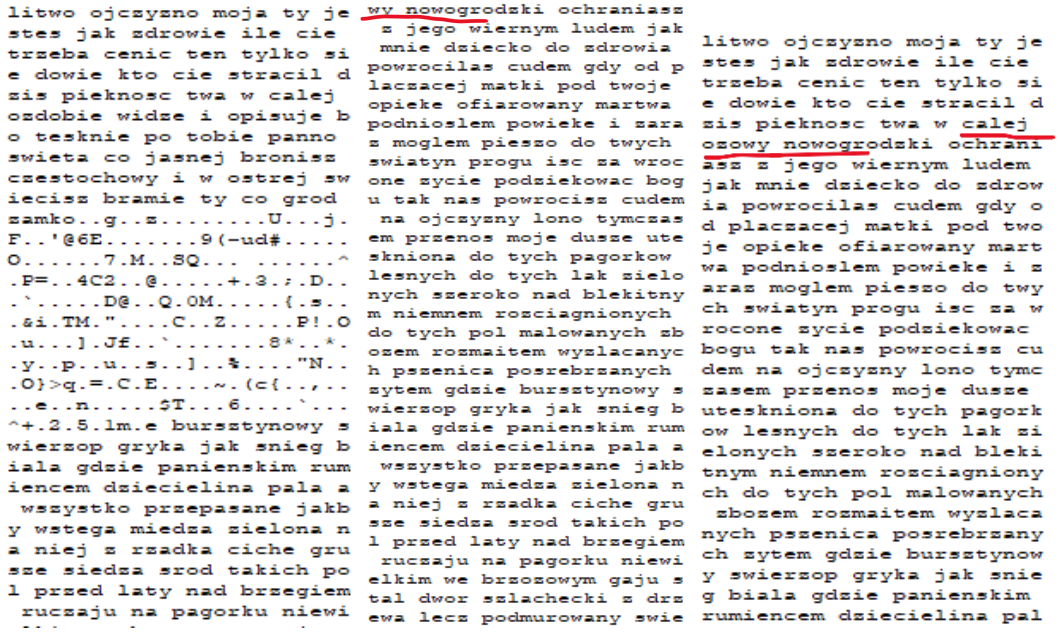


c) usunięcie kilku bajtów



d) usunięcie fragmentu równego długości modułu algorytmu

(pomiędzy dwoma blokami) (usunięcie początkowego bloku) (usunięcie drugiego bloku)



Pytanie 1.7;

Eksperymenty wykazały, że długość klucza ma wpływ na entropię tekstu zaszyfrowanego, ale głównie w przypadku tekstów mniej zróżnicowanych. Największa różnica jest widoczna dla tekstu jednorodnego (tekst nr. 1). Dla tekstu średnio zróżnicowanego (tekst nr 2) wzrost entropii jest mniej widoczny natomiast dla tekstu zróżnicowanego (tekst nr3) różnica jest niezauważalna i długość klucza nie ma wtedy widocznego wpływu na entropię tekstu zaszyfrowanego.

Pytanie 1.8;

Eksperymenty wykazały że dla tekstu jednorodnego i średnio zróżnicowanego (teksty 1, 2) widać było wpływ dłuższego klucza na dłuższe odstępy pomiędzy pikami na wykresach autokorelacji. Dla tekstu zróżnicowanego (tekst nr 3) nie było widać żadnego wpływu o długości klucza, ponieważ wykresy autokorelacji wyglądały bardzo podobnie i nie dało się zauważyć żadnej cykliczności.

Pytanie 1.9;

Eksperymenty wykazały, że entropia tekstu zaszyfrowanego jest zawsze większa od entropii tekstu jawnego niezależnie od długości klucza czy rodzaju tekstu. Dla tekstów mniej zróżnicowanych entropia tekstu zaszyfrowanego wzrośnie zdecydowanie szybciej niż dla tekstów zróżnicowanych, ale dla tekstów bardziej zróżnicowanych końcowa entropia będzie bliżej entropii maksymalnej niż w przypadku szyfrowania tekstów mniej zróżnicowanych

Pytanie 1.10;

Eksperymenty wykazały, że czas szyfrowania rośnie wraz ze wzrostem rozmiaru pliku i długości klucza. Czas szyfrowania/deszyfrowania rośnie mniej więcej liniowo dla rozmiaru pliku, gdzie pliki n razy większe są szyfrowane mniej więcej n razy dłużej. Mniej więcej liniowo rośnie również czas szyfrowania względem długości klucza. Sytuacja wygląda jednak inaczej dla operacji deszyfrowania, gdzie wzrost jest zdecydowanie bliższy do wykładniczego (jest znacznie szybszy niż liniowy). Czas deszyfrowania był zawsze zdecydowanie dłuższy od czasu szyfrowania.

Pytanie 1.11;

Eksperymenty wykazały, że czas szyfrowania algorytmów symetrycznych w głównej mierze wydawał się być zbyt krótki żeby go zmierzyć stoperem, więc wartości musiały być bardzo uśrednione. Większość operacji szyfrowania/deszyfrowania wydawała się być niemal natychmiastowa, a jej czas trwania zależny bardziej od czynników losowych niż długości klucza czy pliku. Nie można wyciągnąć jednoznacznych wniosków co do zależności czasu trwania szyfrowania/deszyfrowania względem rozmiaru pliku czy długości klucza. Deszyfrowanie natomiast zawsze trwało dłużej niż szyfrowanie.

Dla algorytmów asymetrycznych zarówno czas szyfrowania jak i deszyfrowania jest znacząco dłuższy. Widoczne są też zależności wzrostu długości czasu szyfrowania/deszyfrowania proporcjonalnie względem wzrostu długości pliku i wzrostu długości klucza.

Pytanie 1.12;

Eksperymenty wykazały że:

a) Zmiana 1 bajtu spowoduje, że zmianie po deszyfrowaniu ulegnie jedynie fragment tekstu (blok) ze zmienionym bajtem. Reszta tekstu pozostanie bez zmian.
b) Usunięcie 1 bajtu spowoduje, że zmianie po deszyfrowanie ulegnie zarówno blok ze zmnionym bajtem, jak i wszystkie kolejne bloki, które zostały przesunięte
c) Usunięcie kilku bajtów spowoduje wynik podobny do punktu b), szum tekstu pojawia się dopiero od momentu wystąpienia bloku z pierwszym usuniętym bajtem i trwa aż do końca tekstu, niezależnie od tego gdzie zostały zmienione inne bajty.
d) Usunięcie fragmentu równego długości bloku powodowało, że dla fragmentów spójnych okresowo z pozostałymi blokami (czyli w przypadkach których usuwamy cały blok, a nie fragmenty dwóch sąsiadujących bloków) otrzymujemy po deszyfrowaniu spójny tekst bez szumu, natomiast z utraconym fragmentem tekstu o tej samej długości co usunięty fragment. Dla fragmentów niespójnych okresowo (czyli w przypadkach usunięcia fragmentów dwóch sąsiadujących bloków) otrzymujemy normalne dane aż do pierwszego zmienionego bloku, później szum na dwa bloki długości, a później znowu normalny tekst.

Pytanie 1.13;

Eksperymenty wykazały, że tak, można tak zrobić, ale usuwany fragment musi mieć długość równą wielokrotności długości bloku algorytmu, a także fragment ten musi okresowo pasować do innych bloków (nie możemy usuwać fragmentów z dwóch sąsiednich bloków, a jedynie całe pojedyncze bloki). Rozszyfrowany tekst będzie wtedy czytelny, czyli nie będzie posiadać szumu, ale rzecz jasna nie będzie zawierał fragmentu, który był zawarty w bloku który został usunięty.

Pytanie 1.14;

Algorytmy asymetryczne używają pary kluczy, co ułatwia bezpieczne przesyłanie danych, bez konieczności dzielenia się kluczem, który może „wyciec”. Są jednak znacznie wolniejsze i bardziej zasobożerne niż algorytmy symetryczne, które operują na jednym kluczu zarówno do szyfrowania jak i deszyfrowania. Algorytmy symetryczne są szybsze, ale wymagają bezpiecznego przekazania klucza do odbiorcy zaszyfrowanego tekstu, co jest trudniejsze do bezpiecznego wykonania w praktyce.

Pytanie 1.15;

Algorytmy asymetryczne są przeznaczone do wymiany kluczy i autoryzacji, stosuje się je w SSL/TLS (bezpieczne połączenia w internecie) oraz w podpisach cyfrowych, gdzie klucz publiczny weryfikuje tożsamość nadawcy. Z kolei algorytmy symetryczne lepiej sprawdzają się w szyfrowaniu dużych ilości danych, takich jak pliki i bazy danych, ponieważ są znacznie szybsze i bardziej efektywne pod względem wydajności.