Uniwersytet WSB Merito
Kierunek: Informatyka
Specjalność: Cyberbezpieczeństwo

Rok akademicki: 2024/2025
semestr letni

Analiza szyfrów symetrycznych z wykorzystaniem narzędzia CrypTool

Wykonał: Maciej Niemiec

Numer albumu: 107162

1. Wprowadzenie

Bezpieczeństwo informacji jest jednym z fundamentalnych elementów współczesnych systemów teleinformatycznych, a szyfrowanie danych odgrywa kluczową rolę w zapewnianiu poufności, integralności i dostępności informacji. W ramach drugiego laboratorium z przedmiotu ***Cyberbezpieczeństwo*** analizowano działanie wybranych szyfrów symetrycznych, takich jak DES, AES czy IDEA, ze szczególnym uwzględnieniem ich właściwości kryptograficznych oraz wpływu struktury danych wejściowych na jakość szyfrowania.

Zajęcia miały na celu zrozumienie zasad funkcjonowania szyfrów blokowych oraz oceny ich efektywności z punktu widzenia entropii – czyli miary nieprzewidywalności danych. W tym celu przygotowano trzy różne teksty jawne o zróżnicowanej entropii (jednorodny, średnio zróżnicowany i wysoko zróżnicowany), które następnie szyfrowano przy użyciu różnych algorytmów i konfiguracji. Porównanie histogramów oraz wartości entropii przed i po szyfrowaniu pozwoliło na lepsze zrozumienie mechanizmów działania szyfrów symetrycznych i ich odporności na analizę statystyczną.

W dalszej części laboratorium analizowano wpływ trybu pracy szyfratora (ECB i CBC) na strukturę kryptogramu, a także przeprowadzono eksperymenty związane z modyfikacjami szyfrogramów – w tym zmianą pojedynczych bitów oraz manipulacją długością danych. Szczególną uwagę poświęcono także wpływowi długości klucza oraz rodzaju algorytmu na wynik szyfrowania.

Zajęcia wpisują się w szerszy kontekst zagadnień omawianych na wykładach, takich jak ochrona informacji, identyfikacja zagrożeń w cyberprzestrzeni oraz praktyczne aspekty stosowania kryptografii w systemach informatycznych. Laboratorium to miało również na celu rozwijanie umiejętności praktycznej oceny bezpieczeństwa różnych metod kryptograficznych, zgodnie z aktualnymi standardami i wymaganiami bezpieczeństwa teleinformatycznego

2. Część Laboratoryjna

Zadanie 1.

1. **Tekst jednorodny**
2. **Tekst średnio zróżnicowany**
3. **Tekst wysoce zróżnicowany**
4. **Dla kilku ustalonych tekstów jawnych o różnych entropiach (np. tekst jednorodny, tekst średnio zróżnicowany, tekst bardzo zróżnicowany) porównać entropię tekstu jawnegoz entropią po zaszyfrowaniu (entropią tekstu tajnego)**

Tekst jednolity:

Algorytm	Entropia tekstu jawnego	Entropia testu zaszyfrowanego
IDEA (128bit)	0.00	3.06
DES z ECB (64bit)	0.00	3.00
AES (CBC) (128bit)	0.00	7.80

Tekst średnio zróżnicowany:

Algorytm	Entropia tekstu jawnego	Entropia testu zaszyfrowanego
IDEA (128bit)	3.80	3.80
DES z ECB (64bit)	3.80	6.55
AES (CBC) (128bit)	3.80	7.84

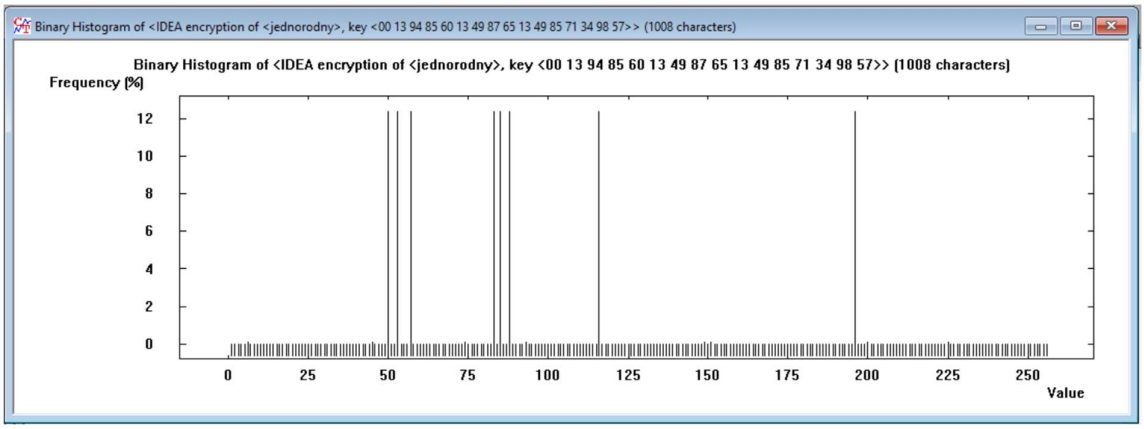
Tekst bardzo zróżnicowany:

Algorytm	Entropia tekstu jawnego	Entropia testu zaszyfrowanego
IDEA (128bit)	4.27	4.25
DES z ECB (64bit)	4.27	7.87
AES (CBC) (128bit)	4.27	7.88

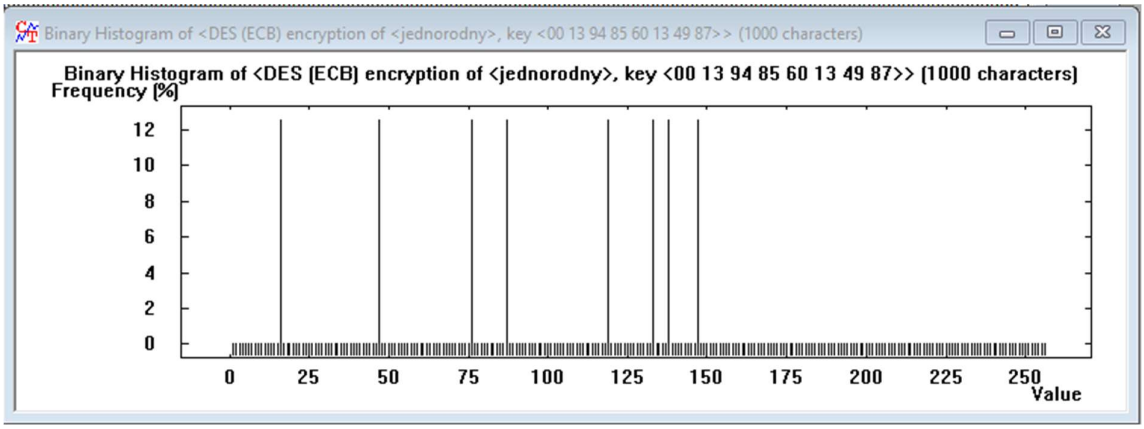
Histogramy:

Tekst jednolity:

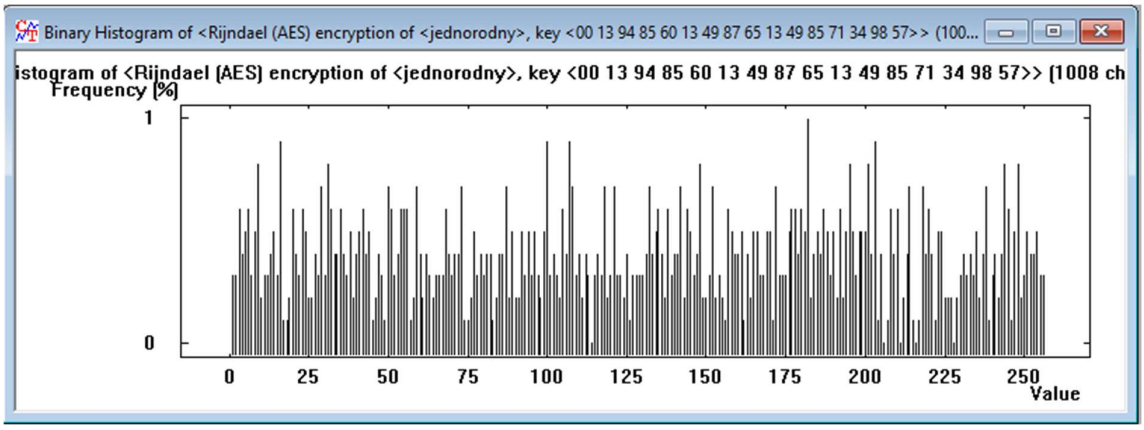
IDEA (128bit) (wkleić zrzut histogramu)



DES z ECB (64bit) (wkleić zrzut histogramu)

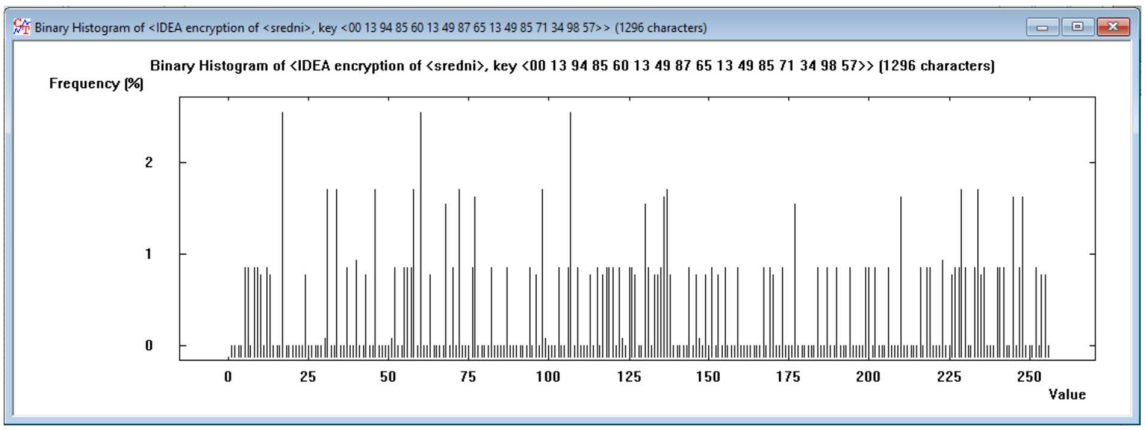


AES (CBC) (128bit) (wkleić zrzut histogramu)

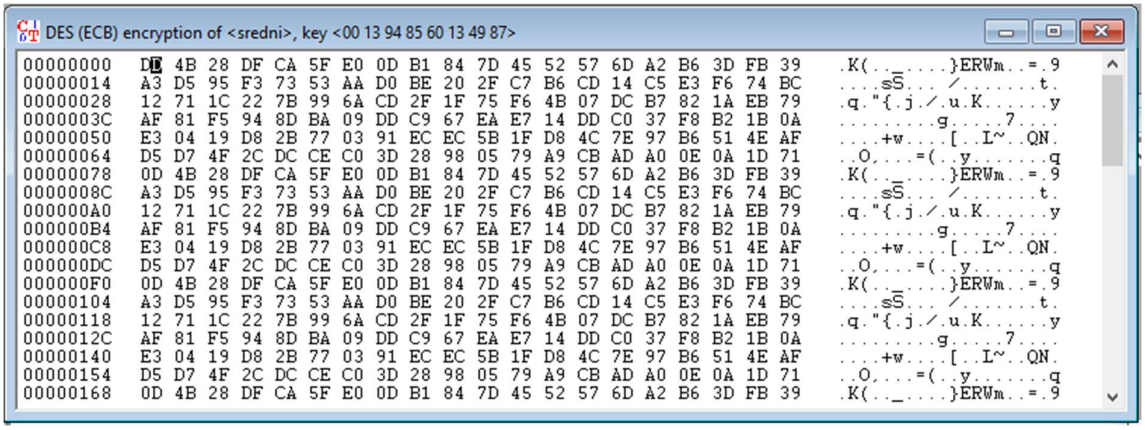


Tekst średnio zróżnicowany:

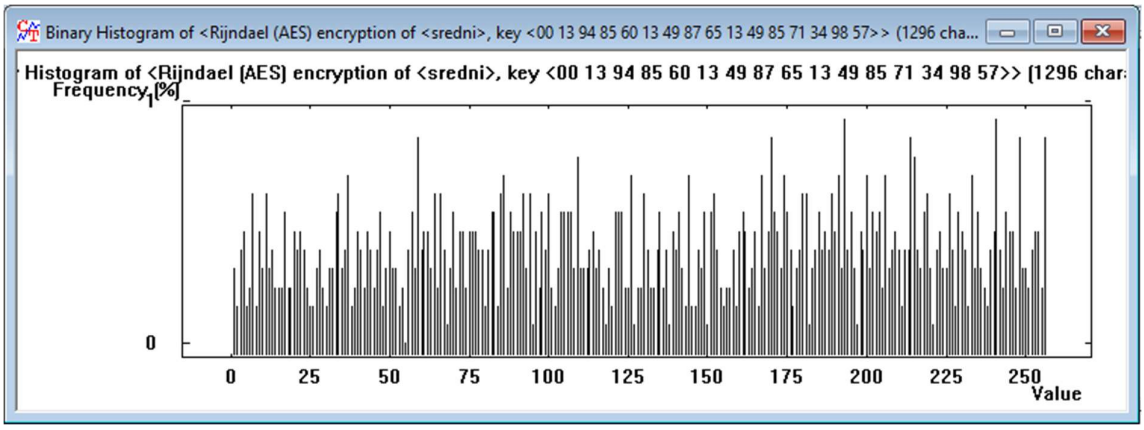
IDEA (128bit) (wkleić zrzut histogramu)



DES z ECB (64bit) (wkleić zrzut histogramu)

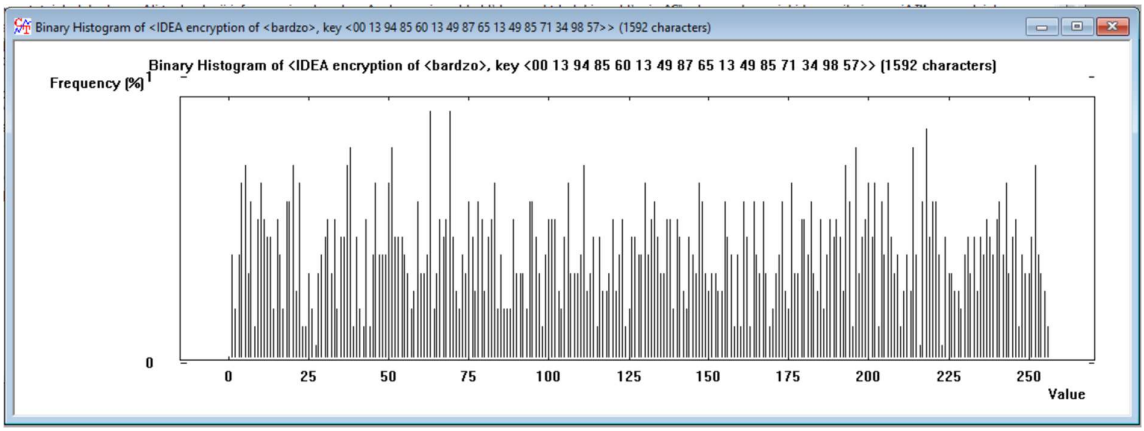


AES (CBC) (128bit) (wkleić zrzut histogramu)

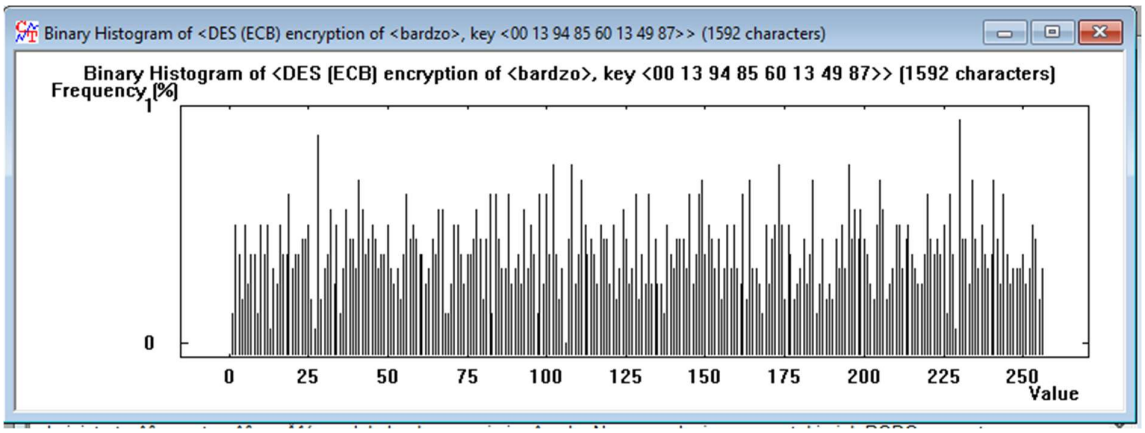


Tekst bardzo zróżnicowany:

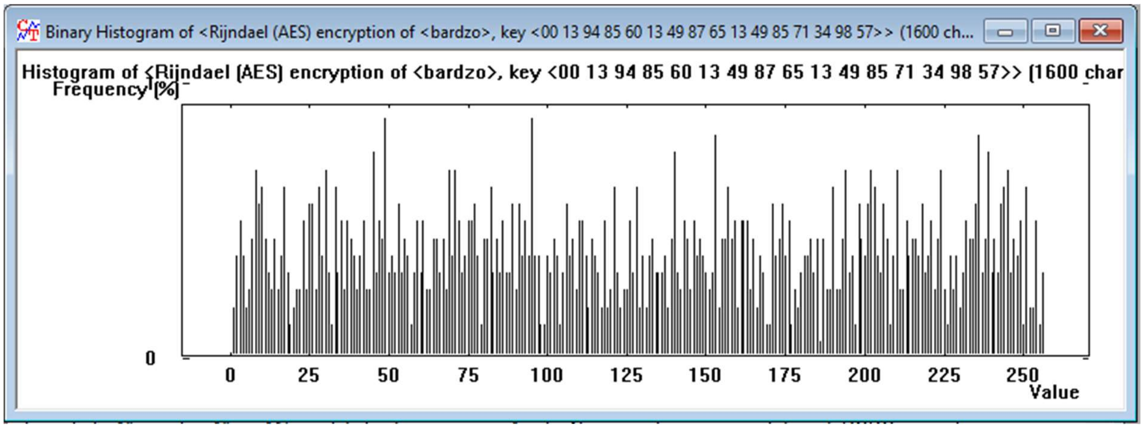
IDEA (128bit) (wkleić zrzut histogramu)



DES z ECB (64bit) (wkleić zrzut histogramu)



AES (CBC) (128bit) (wkleić zrzut histogramu)



5. **Porównać parametry szyfrowania (długość klucza) i zbadać ich wpływ na tekst tajny (np. na jego entropię). Należy w tym punkcie zaszyfrować dany tekst korzystając z różnych dostępnych długości i wartości klucza**

Zadanie 2.

1. **Proszę wygenerować pliki tekstowe jak w zadaniu nr 1.**

Tekst jednorodny:

„aaa”

Tekst średnio zróżnicowany:

„agree all use city with model agree all use city with

model agree all use city with model agree all use city with model agree all use city with model agree all use city with model agree all use city with model agree all use city with model agree all use city with model agree all use city with model”

Tekst wysoce zróżnicowany:

„W ciągu ostatnich dekad rozwój technologii informacyjnych wpłynął na niemal każdy aspekt ludzkiego życia — od sposobu, w jaki komunikujemy się na co dzień, po globalne systemy gospodarcze, transportowe i medyczne. Nowoczesne algorytmy sztucznej inteligencji są w stanie analizować ogromne zbiory danych w czasie rzeczywistym, identyfikując wzorce, które wcześniej były poza zasięgiem tradycyjnych metod statystycznych. Równocześnie jednak rośnie skala zagrożeń związanych z cyberbezpieczeństwem — ataki phishingowe, ransomware, wycieki danych czy manipulacje informacyjne stają się codziennością dla użytkowników i instytucji.

Zabezpieczanie danych nie polega dziś jedynie na ustawieniu hasła do konta, lecz obejmuje cały zestaw technik i protokołów: szyfrowanie, autoryzację wieloskładnikową, kontrolę dostępu, monitorowanie aktywności sieciowej i szybkie reagowanie na incydenty. Każda luka może zostać wykorzystana, dlatego podejście warstwowe (tzw. „defense in depth”) jest standardem w projektowaniu bezpiecznych systemów. Kluczowe znaczenie ma również edukacja użytkowników końcowych, którzy często są najsłabszym ogniwem całego łańcucha bezpieczeństwa.

Wraz z rosnącym znaczeniem chmury obliczeniowej oraz urządzeń IoT, odpowiedzialność za ochronę prywatnych i wrażliwych danych rozszerza się na coraz więcej podmiotów — od indywidualnych użytkowników, przez administratorów systemów, aż po globalne korporacje i rządy. Nowe regulacje prawne, takie jak RODO czy ustawa o krajowym system”

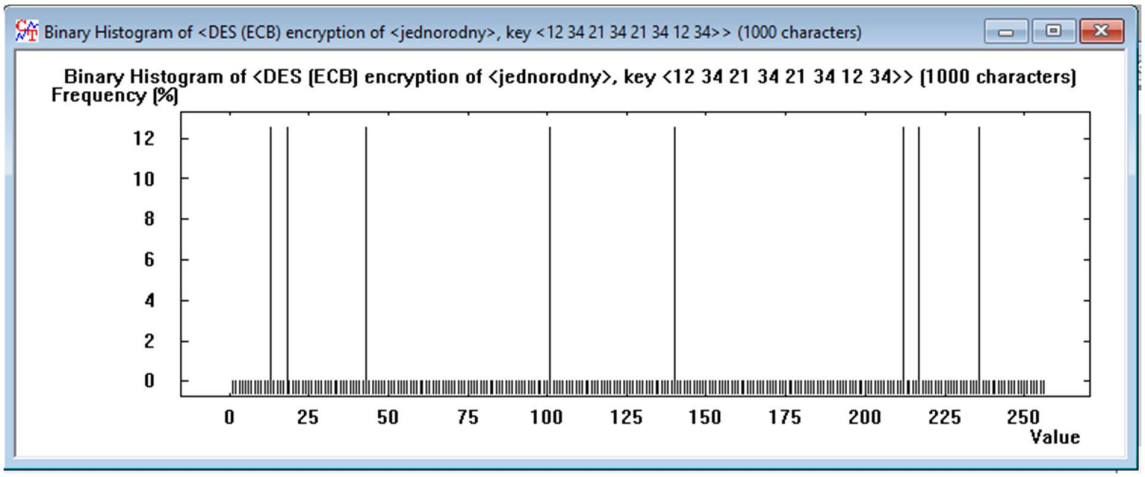
2. **Proszę zaznaczyć przy szyfrowaniu opcję PlainText a przy deszyfrowaniu Hex.**
3. **Proszę wybrać jeden z dostępnych algorytmów szyfrowania i tekst zaszyfrować korzystając z różnych trybów pracy szyfratora, tzn. utworzyć kryptogram dla trybów: ECB i CBC –** Wybrano algorytm DES
4. **Proszę obejrzeć histogramy i obliczyć entropię dla tak utworzonych TT.**

Algorytm DES ECB

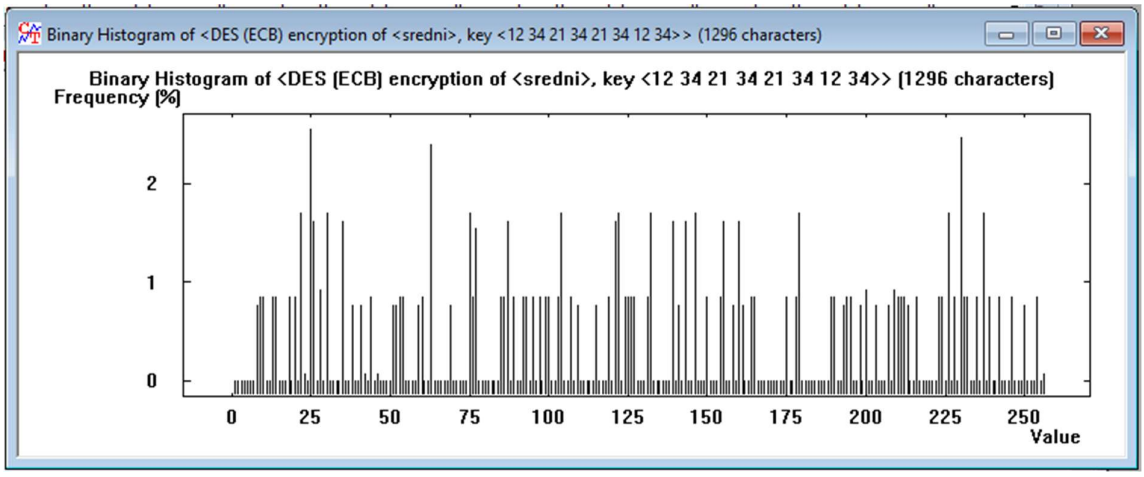
Zróżnicowanie tekstu	Entropia tekstu jawnego	Entropia testu zaszyfrowanego
Tekst jednorodny	0.00	3.00
Tekst średnio zróżnicowany	3.80	6.48
Tekst bardzo zróżnicowany	4.27	7.87

Histogramy:

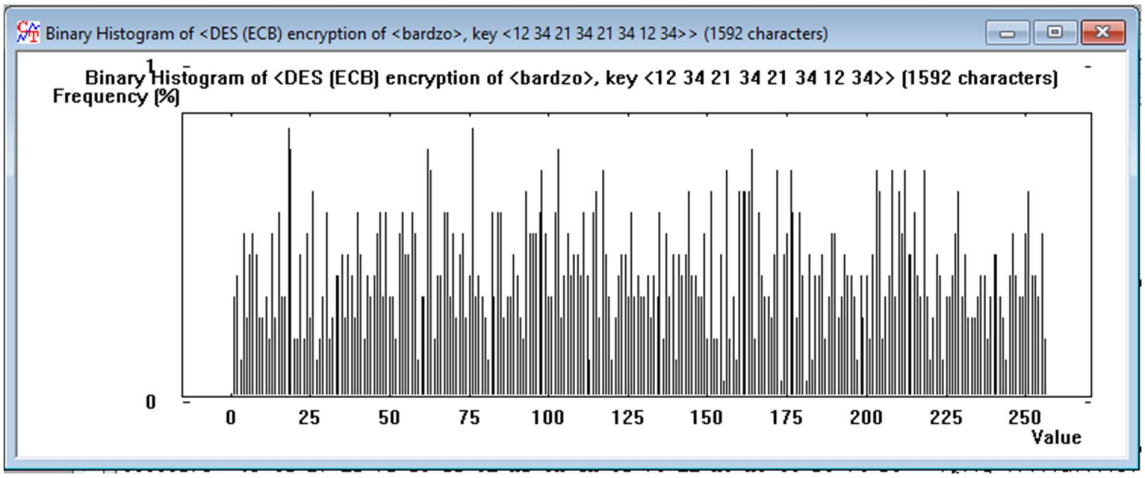
Tekst jednolity:



Tekst średnio zróżnicowany:



Tekst bardzo zróżnicowany:

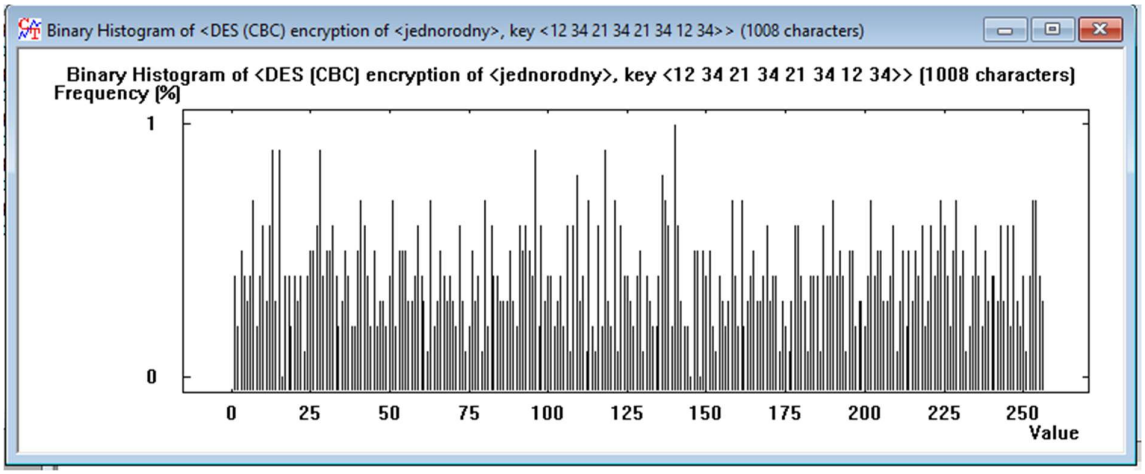


Algorytm DES CBC

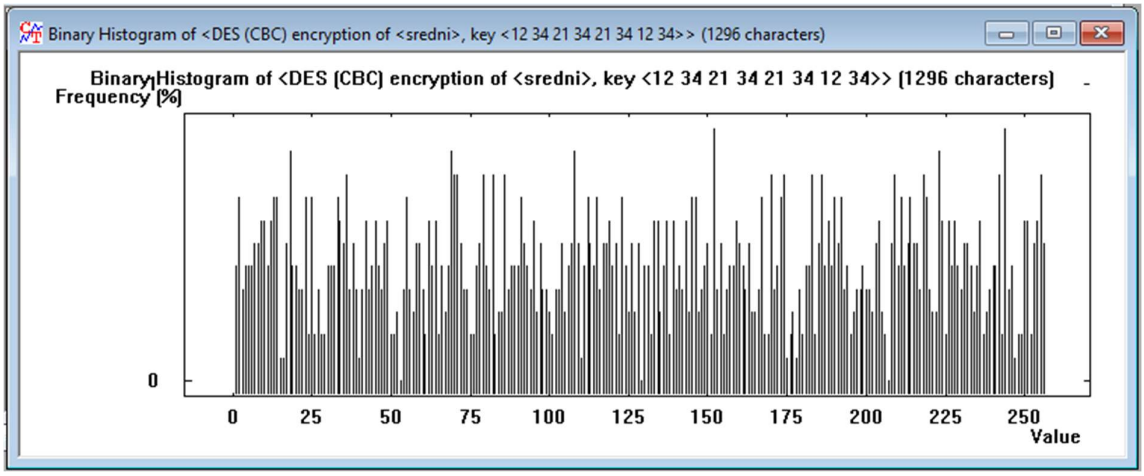
Zróżnicowanie tekstu	Entropia tekstu jawnego	Entropia testu zaszyfrowanego
Tekst jednorodny	0.00	7.82
Tekst średnio zróżnicowany	3.80	7.84
Tekst bardzo zróżnicowany	4.27	7.88

Histogramy:

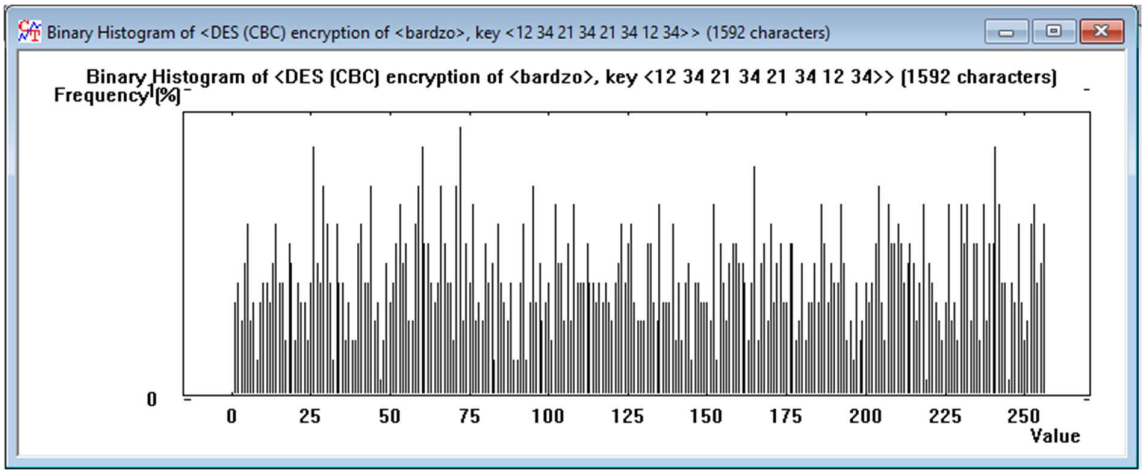
Tekst jednolity:



Tekst średnio zróżnicowany:



Tekst bardzo zróżnicowany:



5. **Dla pliku z tekstem jednorodnym proszę obejrzeć zawartość kryptogramu i ocenić przydatność każdego trybu pracy szyfratora do szyfrowania plików składających się z powtarzających się bloków bajtów.**

Tryb ECB

W trybie ECB (Electronic Codebook), identyczne bloki danych wejściowych są szyfrowane w identyczny sposób. W zaszyfrowanym pliku zaobserwowano regularnie powtarzające się bloki szyfrogramu. Oznacza to, że struktura danych jawnych nie została ukryta. Tryb ten **nie zapewnia poufności** w przypadku danych o powtarzalnej strukturze.

Tryb CBC

W trybie CBC (Cipher Block Chaining), każdy blok szyfrogramu zależy nie tylko od bieżącego bloku danych, ale także od poprzedniego bloku szyfrogramu. Dzięki temu nawet identyczne bloki wejściowe generują różne bloki wyjściowe. W wyniku tego szyfrogram wygląda jak losowy i nie ujawnia żadnych powtórzeń – **skutecznie maskując strukturę danych**.

Wniosek

Tryb **ECB jest niezalecany** do szyfrowania plików zawierających powtarzające się dane, ponieważ nie ukrywa ich struktury. Z kolei tryb **CBC znacznie lepiej zabezpiecza dane**, zapewniając brak powtarzalności w szyfrogramie. **CBC jest zdecydowanie bardziej przydatny do szyfrowania danych jednorodnych**.

6. **Dla pliku z tekstem jednorodnym proszę zaszyfrować za pomocą jednego wybranego algorytmu działającego we wszystkich trybach pracy. Następnie dokonać następujących zmian w szyfrogramie:**

a) **zmienić jeden bit,**

Zmiana pojedynczego bitu w szyfrogramie zakłóciła odszyfrowanie bieżącego bloku oraz kolejnego bloku – efekt łańcuchowy wynikający z natury trybu CBC, w którym każdy blok szyfrowany zależy od poprzedniego.

b) zmienić po jednym lub kilka bitów w różnych bajtach (blisko lub daleko od siebie).

Zmiany w kilku miejscach spowodowały losowe zakłócenia odszyfrowanych bloków odpowiadających zmodyfikowanym fragmentom szyfrogramu oraz – w przypadku CBC – dodatkowe zakłócenia bloków następujących po zmodyfikowanych. Odszyfrowany tekst staje się częściowo lub całkowicie nieczytelny, w zależności od ilości i rozmieszczenia zmian.

7. Co spowoduje następujące zmiany dla szyfru AES:

a) dodanie jednego bajtu,

Zaburza strukturę blokową danych (AES oczekuje danych w wielokrotnościach 16 bajtów). Powoduje błąd odszyfrowania lub niepoprawny wynik – całe deszyfrowanie się „rozsypuje".

b) usunięcie jednego bajtu,

Analogicznie – brak jednego bajtu uniemożliwia poprawny podział danych na bloki, co uniemożliwia odszyfrowanie całości. Może skutkować błędem parsera lub nieczytelnym tekstem.

c) dodanie kilku bajtów,

Zmienia strukturę bloków, powodując przesunięcie danych i niepoprawne ich odszyfrowanie. CBC przestaje działać poprawnie, gdy naruszona zostanie integralność długości.

d) usunięcie kilku bajtów.

Podobnie jak wyżej – usunięcie kilku bajtów powoduje błąd interpretacji długości bloku, prowadząc do błędnego odszyfrowania lub błędu działania algorytmu.

Pytania / Wnioski

1. Gdzie są stosowane algorytmy blokowe? Które algorytmy są najbardziej popularne?

Algorytmy blokowe są stosowane wszędzie tam, gdzie wymagana jest poufność i integralność danych. W szczególności:

W przemyśle motoryzacyjnym (automotive):

- **Bezpieczna komunikacja ECU-ECU (Electronic Control Units)**: Przykład to szyfrowanie danych przesyłanych w sieci CAN lub FlexRay w pojazdach.
- **Bezpieczna aktualizacja oprogramowania (OTA)**: Dane przesyłane do ECU są szyfrowane np. AES-em, aby uniemożliwić przechwycenie lub modyfikację firmware'u.
- **Zabezpieczenie danych użytkownika w infotainment** – np. książka telefoniczna, dane logowania, dane nawigacyjne.
- **Zabezpieczenia komunikacji V2X (Vehicle-to-Everything)** – wymagają szyfrowania i uwierzytelniania informacji.

Najpopularniejsze i aktualnie zalecane algorytmy blokowe (industry standard):

AES (Advanced Encryption Standard)

Status: Globalny standard (wg NIST, ISO, AUTOSAR)

Parametry: Blok 128-bit, klucze 128/192/256-bit

Tryby: CBC, GCM, CCM – w zależności od zastosowania (np. GCM do autentykacji)

Zalety:

- Wspierany sprzętowo przez wiele automotive-grade mikrokontrolerów (np. Infineon, NXP, Renesas).
- Niskie zużycie energii, wysoka wydajność w systemach embedded.

Zastosowanie: OTA, MAC computation, szyfrowanie danych diagnostycznych.

Camellia

Status: Zatwierdzony przez ISO/IEC, uznany przez NESSIE, alternatywa dla AES

Parametry: Blok 128-bit, klucz 128/192/256-bit

Zalety:

- Bezpieczeństwo porównywalne z AES.
- Efektywność na różnych platformach, także embedded.

Zastosowanie: W projektach wymagających dywersyfikacji algorytmów lub dla organizacji, które preferują japońskie standardy (np. JASPAR).

SKINNY / ASCON (dla lekkich systemów i odporności kwantowej)

ASCON:

Status: Zwycięzca NIST LWC (Lightweight Cryptography) w 2023 r.

- **Odporność kwantowa:** częściowa – zaprojektowany z myślą o odporności na ataki przyszłości
- **Zastosowanie:** Małe ECU, czujniki, IoT w pojazdach.

SKINNY:

- **Zalecany przez:** ISO/IEC 29192, używany w lekkiej kryptografii.
- **Zastosowanie:** Czujniki, systemy z niskim zapotrzebowaniem na energię.

Z racji na posiadane doświadczenie w branży automotive, odnosiłem się przede wszystkim do rozwiązań wykorzystywanych w tym sektorze. W praktyce przemysłowej AES pozostaje dominującym algorytmem, jednak w kontekście rozwoju systemów rozproszonych i niskomocowych (np. czujniki ADAS, V2X), a także rosnącego ryzyka ataków kwantowych, coraz częściej analizowane są lekkie i bardziej odporne alternatywy, takie jak ASCON.

2. Jakie wartości parametrów (długość bloku, długość klucza) uznaje się współcześnie za standardowe (bezpieczne)?

Współczesne standardy bezpieczeństwa (wg NIST i branżowych rekomendacji):

Długość bloku:

- **128 bitów** – to standardowa długość bloku dla AES. Jest wystarczająco długa, by uniknąć ataków typu birthday attack dla typowych zastosowań.

Długość klucza:

- **128 bitów** – minimalna długość uznawana obecnie za bezpieczną dla większości zastosowań.
- **192 i 256 bitów** – stosowane w środowiskach o zwiększonych wymaganiach (np. dane wrażliwe, komunikacja między serwerami, infrastruktura bezpieczeństwa pojazdu).

Przykład automotive: W systemie **OTA update** producent OEM może stosować **AES-256** do szyfrowania firmware'u, który przesyłany jest przez internet do bramki centralnej pojazdu, a następnie do ECU.

3. Co możemy powiedzieć o obserwowanych zmianach w histogramach i wartościach entropii podczas realizacji operacji wymienionych w punktach 1 i 2?

Podczas realizacji operacji szyfrowania różnych plików (zarówno zróżnicowanych, jak i jednorodnych) zaobserwowano następujące zjawiska:

Histogramy szyfrogramów ECB (DES): wykazują powtarzalność, szczególnie w przypadku danych jednorodnych – co wskazuje na brak ukrycia wzorców.

Histogramy CBC i AES: pokazują równomierne rozproszenie wartości bajtów – co świadczy o dobrej jakości szyfrowania.

Entropia:

- Wzrasta przy większym zróżnicowaniu danych wejściowych.
- Algorytmy jak **AES (CBC)** generują **wysoką entropię niezależnie od struktury danych wejściowych**, co świadczy o ich wysokiej odporności na analizę statystyczną.

Wniosek:
AES w trybie CBC zapewnia najlepsze właściwości kryptograficzne – niezależnie od jakości danych wejściowych generuje szyfrogram statystycznie losowy i odporny na analizę. DES w trybie ECB nie spełnia tych wymagań, zwłaszcza w kontekście danych jednorodnych.

4. Co możemy powiedzieć o tych wartościach w kontekście podobnych ćwiczeń realizowanych dla algorytmów klasycznych?

Algorytmy klasyczne (np. szyfr Cezara, szyfr Vigenère'a):

- **Histogram szyfrogramu** nie ulega znacznemu spłaszczeniu – nadal można dostrzec wzorce znane z tekstu jawnego.

- **Entropia** szyfrogramu rośnie, ale znacznie mniej niż w przypadku współczesnych algorytmów – nadal można dokonywać analizy częstotliwości i łamać szyfr.

Porównanie:

- Algorytmy klasyczne zapewniają **niską odporność na analizę statystyczną**, co czyni je bezużytecznymi w zastosowaniach przemysłowych.
- W branży automotive ich użycie byłoby poważnym naruszeniem zasad **cybersecurity zgodnych z normą ISO/SAE 21434**.

5. Czy długość klucza wpływa na entropię tekstu tajnego?

Bezpośrednio - niekoniecznie. Pośrednio - tak.

- Sam **szyfrogram (tekst zaszyfrowany)** przy poprawnym algorytmie i poprawnej implementacji będzie wyglądał jak losowy ciąg, niezależnie od długości klucza.
- **Dłuższy klucz** zwiększa tzw. **przestrzeń możliwych kluczy**, czyli liczbę wszystkich unikalnych kombinacji, jakie można uzyskać dla danej długości klucza. Przestrzeń ta rośnie wykładniczo wraz z długością klucza – przykładowo:
    - dla klucza 128-bitowego możliwych kombinacji jest 2^{128}, czyli około 3.4×10^{38}
    - dla klucza 256-bitowego to już 2^{256}, czyli liczba tak ogromna, że nawet przy użyciu wszystkich komputerów na świecie nie dałoby się sprawdzić wszystkich możliwości w rozsądnym czasie.

  Im większa ta przestrzeń, tym **bardziej nierealny staje się atak typu brute-force**, polegający na systematycznym testowaniu wszystkich możliwych kluczy w celu odszyfrowania danych.

Entropia szyfrogramu będzie podobna dla AES-128 i AES-256, *jeśli klucz jest losowy i algorytm działa poprawnie*.

Ważna uwaga praktyczna (automotive): W niektórych mikrokontrolerach długość klucza wpływa na sposób przetwarzania (np. sprzętowe szyfrowanie z AES-256 może być wolniejsze), co ma znaczenie dla czasu działania funkcji bezpieczeństwa w pojazdach.

6. Od czego zależy obserwowana entropia tekstu tajnego (np. klucz, algorytm, tekst jawny)?

Entropia szyfrogramu zależy od kilku czynników:

- **Algorytm szyfrowania** – nowoczesne algorytmy jak AES zapewniają maksymalną entropię szyfrogramu. Proste algorytmy (np. XOR) nie.
- **Długość i jakość klucza** – klucz o niskiej entropii (np. "123456") może skutkować szyfrogramem o niższej losowości.
- **Tryb szyfrowania (ECB, CBC, CTR, GCM)** – np. **ECB** nie ukrywa wzorców w tekście jawnym, przez co histogram szyfrogramu może mieć widoczne struktury.
- **Tekst jawny** – jeśli tekst jawny ma bardzo niską entropię (np. "AAAAAAAAAAAA..."), to tylko dobry algorytm i tryb szyfrowania ukryje tę strukturę.