Politechnika Wrocławska

Sprawozdanie z zajęć laboratoryjnych
Cyberbezpieczeństwo

Lab 03
Blokowe algorytmy szyfrowania

Tomasz Mroczko, 266604

October 22, 2023

1 Dane testowe

Użyty tekst

Zgodnie z poleceniem, do eksperymentów użyto 3 tekstów o różnych poziomach entropii:

1) **Tekst jednorodny:** Jako tekst jednorodny użyto litery 'a' powtórzonej 2000 razy.
2) **Tekst częściowo zróżnicowany:** Ciąg słów "this test is very repetetive" powtórzony 100 razy.
3) **Tekst zróżnicowany:** Fragment początku książki "1984", autorstwa George'a Orwella, w języku angielskim. Fragment został ograniczony do długości około 1700 znaków.

2 Zadania

Zadanie 1.1

Dla kilku ustalonych tekstów jawnych (TJ) o różnych entropiach (np. tekst jednorodny, tekst średnio zróżnicowany, tekst bardzo zróżnicowany) porównać entropię TJ z entropią po zaszyfrowaniu (entropią tekstu tajnego TT).

	Rodzaj tekstu		
Algorytm	Jednorodny	Średnio zróżnicowany	Normalny
Brak	0.00	3.00	4.21
AES (CBC)	7.92	7.94	7.89
DES (CBC)	7.90	7.93	7.90
IDEA	2.03	5.51	7.68

Table 1: Entropia tekstów po zaszyfrowaniu różnymi algorytmami

Przyjrzano się również histogramom tekstów tajnych. Zgodnie z tym co sugeruje entropia, histogramy AES oraz DES w każdym wypadku okazały się dość płaskie, co wskazuje na niską schematyczność powtórzeń znaków w tekście, *niezależnie* od tekstu jawnego.

W związku z brakiem widocznego gołym okiem schematu w histogramach, nie załączono zrzutów ekranu histogramów DES ani AES

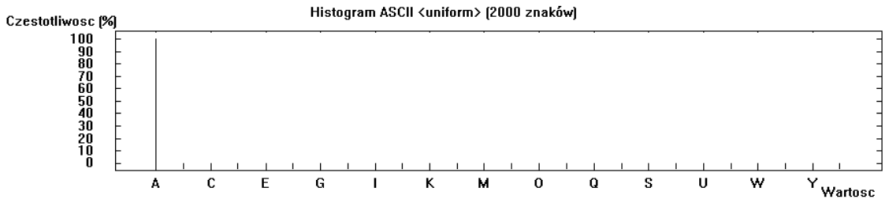


Figure 1: Histogram tekstu jawnego jednorodnego

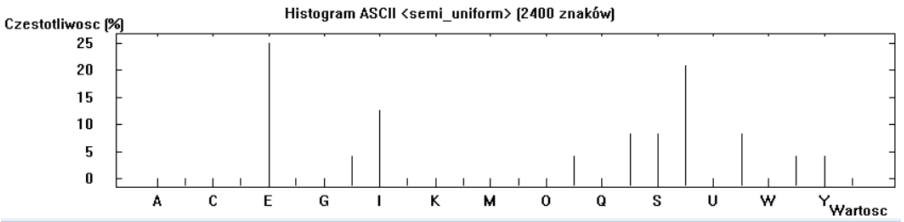


Figure 2: Histogram tekstu jawnego średnio zróżnicowanego

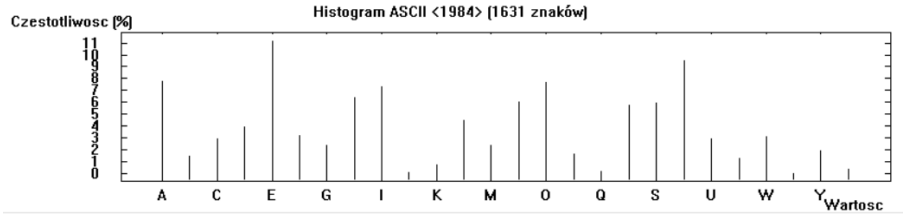


Figure 3: Histogram tekstu jawnego zróżnicowanego

Szyfrowane IDEA

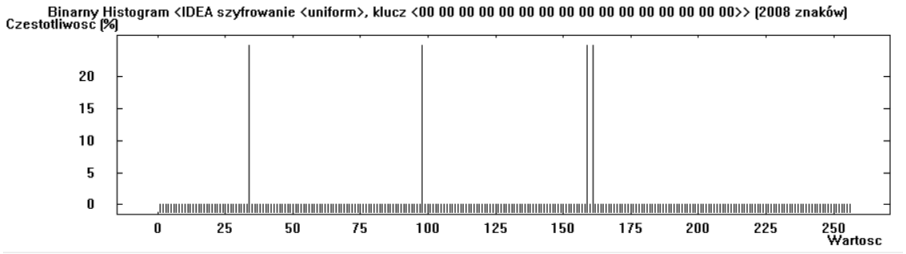


Figure 4: Histogram tekstu jednorodnego po zaszyfrowaniu IDEA

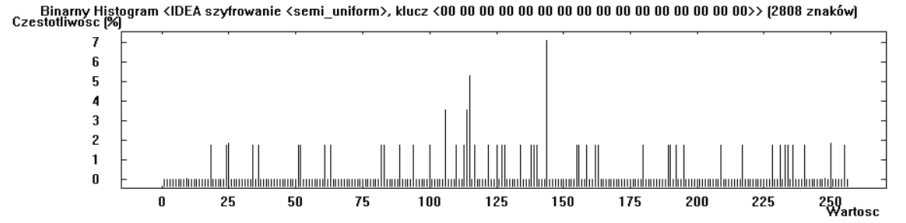


Figure 5: Histogram tekstu średnio zróżnicowanego po zaszyfrowaniu IDEA

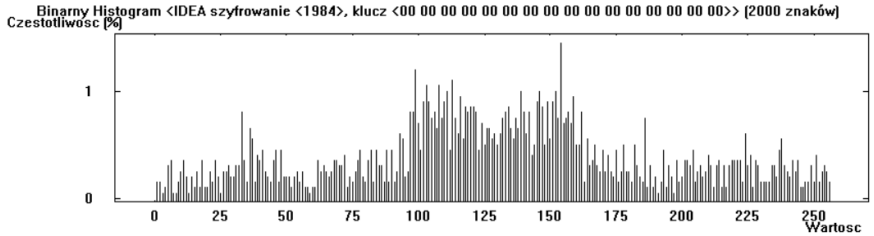


Figure 6: Histogram tekstu zróżnicowanego po zaszyfrowaniu IDEA

Algorytm IDEA okazał się podatny na zmianę zróżnicowania tekstu jawnego. Dla tekstu jednorodnego, jego histogram binarny wykazał tylko 4 różne wartości, powtarzające się równomiernie (25% każda). Wraz ze zróżnicowaniem tekstu, rosła entropia tekstu zaszyfrowanego, a histogram stawał się coraz bardziej płaski. Po zaszyfrowaniu tekstu częściowo zróżniocowanego, histogram dalej wykazuje ewidentną wybiórczość użytych wartości, podczas gdy większość nieużywana jest wcale. Dopiero dla tekstu zróżnicowanego, histogram jest dość wyrównany a tendencje do powtarzania pojedynczych wartości znikają. Jednak nawet tutaj widać wyraźne powtarzanie wartości na środku zakresu (mnej więcej od 100 do 150).

Zadanie 1.2

> *Zaszyfruj dany tekst jawny korzystając z różnych dostępnych długości i wartości klucza, sprawdź, które algorytmy oferują zmianę długości klucza. Porównaj otrzymane wyniki jak zmiana parametrów wpływa na entropie TT.*

Zaszyfrowano tylko tekst naturalny (wysoko zróżnicowany). Wykorzystano algorytm AES(CBC) oraz następujące wartości klucza:

Klucz 1 Same zera

Klucz 2 12327A3212901939929193919191O323 (tylko dla 128 bit)

Klucz 3 ABA32184382193412BBA9890910234812340980912348901 (tylko dla 192 bit)

Klucz 4 012312312938479879 8B7A98798798A798798798798798712983791827319827 (tylko dla 256 bit)

Algorytm	Entropia
Brak	4.21
AES (CBC-128), Klucz 1	7.89
AES (CBC-192), Klucz 1	7.89
AES (CBC-256), Klucz 1	7.89
AES (CBC-128), Klucz 2	7.91
AES (CBC-192), Klucz 3	7.91
AES (CBC-256), Klucz 4	7.92

Jak widać, dla każdej wartości i długości klucza, entropia *znacząco* się zwiększyła. Przy kluczu złożonym z samych zer, entropia nie rosła wraz z długością klucza. Dla klucza zróżnicowanego, entropia zwiększyła się nieco względem samych zer, a dłuższy klucz zwiększył jeszcze ten efekt, jednak różnica jest bardzo mała. Podsumowując, algorytm AES znacząco zwiększa entropię tekstu niezależnie od klucza, klucz jednak *może* nasilić ten efekt.

Zadanie 1.3

> *Gdzie obecnie stosowane są algorytmy blokowe? Które algorytmy są najbardziej popularne? Jakie wartości parametrów (długość bloku, długość klucza) uznaje się współcześnie za standardowe (bezpieczne)?*

Algorytmy blokowe są powszechnie stosowane w kryptografii. Znajdują zastosowanie w takich dziedzinach jak protokoły bezpieczeństwa sieci, szyfrowanie dysków, blockchain, szyfrowanie baz danych. Wiele popularnych komunikatorów używa ich do bezpiecznego przesyłania wiadomości. Programy do konwersji plików, takie jak 7zip i WinRAR także je wykorzystują.

Najpopularniejsze algorytmy blokowe to między innymi:

- **AES (Advanced Encryption Standard)** to nowoczesny szyfr blokowy, jeden z najbardziej popularnych na świecie. Obsługuje różne długości klucza i jest powszechnie stosowany w wielu zastosowaniach.
- **DES (Data Encryption Standard)** starszy algorytm, kiedyś bardzo popularny, na początku 21 wieku jego bezpieczeństwo przestało być wystarczające. Nie używany w nowoczesnych systemach

- **3DES(Triple DES)** jest bezpieczniejszym szyfrem bazującym na *DES*, jednak również ma swoje ograniczenia i nie jest zalecany do współczesnych zastosowań.

Obecnie za standardowe parametry zwykle uważa się długość bloku 128 bit. Klucz 256 bit dla maksymalnego poziomu bezpieczeństwa, jednak 128 oraz 192 również są szeroko używane.

Zadanie 1.4

> *Co możemy powiedzieć o obserwowanych zmianach w histogramach i wartościach entropii podczas realizacji powyższych zadań?*

Dla wszystkich testowanych algorytmów AES(CBC), DES(CBC), IDEA, entropia *znacząco* zwiększyła się po zaszyfrowaniu. Algorytmy AES oraz DES, niezależnie od zróżnicowania tekstu jawnego, uzyskały bardzo płaski histogram oraz niemal maksymalny poziom entropii. Algorytm AES wykazał *bardzo niewielką* tendencję do zwiększenia entropii wraz z długością klucza.

Jeśli chodzi o algorytm IDEA, wyniki zarówno entropii jak i histogramu były bardzo zależne zarówno od użytego klucza, jak od zróżnicowania tekstu. Dla tekstu jednorodnego dał niezadawalające się efekty (histogramy ewidentnie wskazują powtarzalność). Wraz ze wzrostem zróżnicowania tekstu jawnego rośnie zróżnicowanie tekstu zaszyfrowanego. Warto jednak zaznaczyć, że nawet dla tekstu mocno zróżnicowanego, histogram wykazał widoczne powtórzenia.

Zadanie 1.5

> *Co możemy powiedzieć o tych wartościach w kontekście podobnych ćwiczeń realizowanych dla algorytmów historycznych (klasycznych)?*

Na tle algorytmów historycznych, algorytmy blokowe wydaja się *znacznie* silniejsze. W przypadku algorytmów historycznych, wzrost entropii oraz zmiana histogramu były *bardzo* zależne od długości oraz charakteru użytego klucza, a także zróżnicowania tekstu jawnego. Algorytmy blokowe zwiększają entropię na *bardzo* wysoki poziom, spłaszczają histogramy, a parametry klucza oraz tekstu jawnego, nie mają tak istotnego wpływu na jakośc tekstu zaszyfrowanego. Nawet dla tekstu jednorodnego oraz klucza złożonego z samych zer, entropia była wysoka a histogram płaski (poza algorytmem IDEA, on nie radził sobie tak dobrze).

Zadanie 1.6

> *Czy długość klucza wpływa na entropię TT?*

W przypadku algorytmu AES(CBC), eksperymenty wykazały *bardzo nieznaczną* poprawę poziomu entropii dla dłuższych kluczy (w przypadku klucza innego niż same zera). Jendak trudno stwierdzić czy jest to przypadek, czy wpływ długości hasła.

Zadanie 1.7

> *Czy obserwowana entropia TT zależy od entropii TJ?*

Algorytmy AES(CBC) oraz DES(CBC) nie wykazały zależności entropii TT od TJ. Entropia TT dla algorytmu IDEA była z kolei *ewidentnie* zależna od TJ.

Zadanie 1.8

Czy obserwowana entropia TT zależy od wartość klucza?

Zmiana wartości klucza nie wpłynęła znacząco na zmianę entropii TT.

Zadanie 1.9

Czy obserwowana entropia TT zależy od użytego algorytmu?

Zależność entropii TT od algorytmu jest znacznie mniejsza niż w przypadku algorytmów historycznych. AES(CBC) oraz DES(CBC) wykazały podobne, bardzo wysokie poziomy entropii. Jendak algorytm IDEA osiągnął dużo niższe wyniki.

Zadanie 2.1

Proszę wygenerować plik tekstowy z cyklicznie powtarzającą się zawartością (np. może to być plik zawierający jedynie litery – „A", ciągi – „abcd", itp.)

Użyto tych samych tekstów co w zadaniu 1.

Zadanie 2.2

.Proszę wybrać jeden z dostępnych algorytmów szyfrowania oraz tekst zróżnicowany. Zaszyfruj TJ korzystając z różnych trybów pracy szyfratora, tzn. utworzyć kryptogram dla trybów: ECB, CBC,OFB,CFB

Wybrano algorytm AES.
W związku z kodowaniem Base64 przez stronę *encode-decode.com*, w celu ujednolicenia wyników z eksperymentami przeprowadzonymi w *Cryptool*, zakodowany tekst przekonwertowano na kodowanie HEX z pomocą narzędzie webowego *cryptii.com/pipes/base64-to-hex*. Po dokonaniu konwersji tekst przeniesiono do *Cryptoola*, gdzie wcześniej ustawiono tryb tekstu "Heksadecymalnie".

Zadanie 2.3

Proszę obejrzeć histogramy i obliczyć entropię dla tak utworzonych TT

Algorytm	Entropia
Brak	4.21
AES(128-ecb)	7.91
AES(128-cbc)	7.89
AES(128-ofb)	7.91
AES(128-cfb)	7.89

Po obliczeniu entropii przeanalizowano histogramy. Dla każdeo trybu pracy, histogram był płaski, bez wyraźnych wzniesień ani schematów. W związku z podobieństwem i płaskim charakterem histogramów, nie załącono zrzutów ekranu.

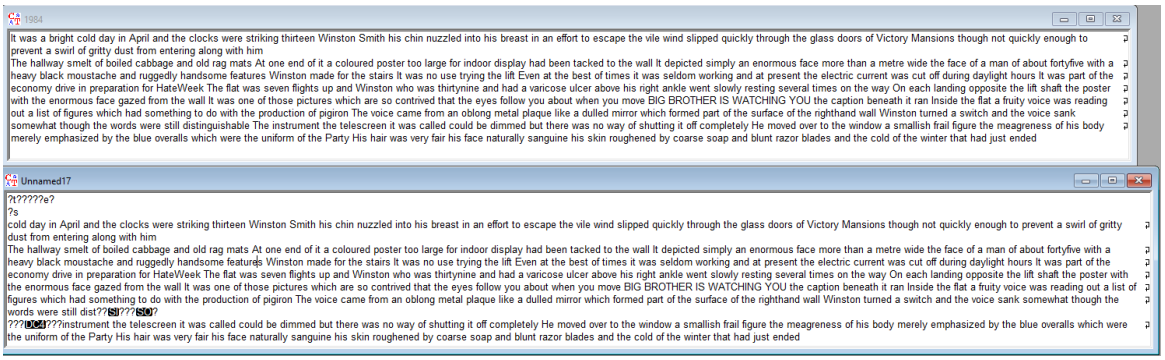
Zadanie 2.5

Zaszyfruj za pomocą algorytmu AES lub DES zaszyfruj tekst jawny (tekst zróżnicowany).
- *zmień po jednym lub kilka bitów w różnych bajtach (blisko lub daleko od siebie),*
- *dodanie jednego bajtu,*
- *usunięcie jednego bajtu,*
- *dodaj/ usuń fragmentu tekstu równego długości bloku algorytmu,*

Odszyfruj szyfrogramy i sprawdzić ich zawartość

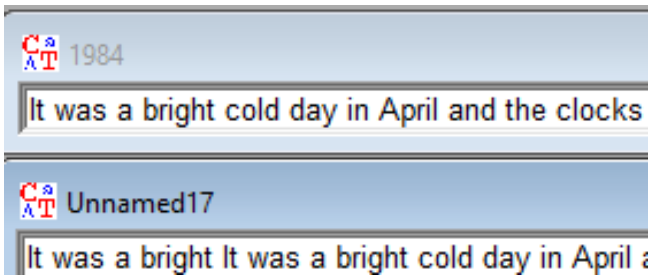
Wybrano algorytm AES.

- **Tryb ECB**
  - **Zmiana jednego lub kilku bitów**



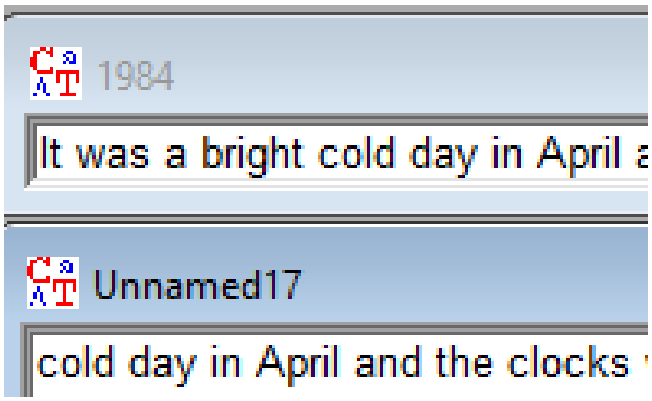
Po zmianie jednego lub kilku bitów, fragmenty tekstu o długości bloku(16 znakowe) stały sie nieczytelne. Nie zaobserwowano propagacji na następne bloki.

- **Dodanie/usunięcie bajtu** Po dodaniu lub usunięciu bajtu nie udało się odszyfrować krytptogramu. Dzieje się tak, ponieważ kryptogram powinien mieć długość równą wielokrotności długości bloku.
- **Dopisanie długości bloku (skopiowano pierwsze 32 znaki kodu hex)**



Skopiowanie pierwszego bloku, spowodowało podwojenie pierwszych 16 znaków. Pozostała część tekstu pozostała nienaruszona.

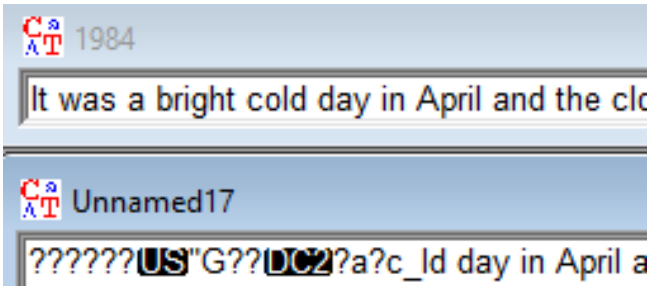
- **Usunięcie długości bloku (usunięto pierwsze 32 znaki kodu hex)**



Analogicznie do dopisania, usunięcię długości bloku, spowodowało wycięcie pierwszych 16 znaków, bez wpływu na resztę tekstu.

- **Tryb CBC**

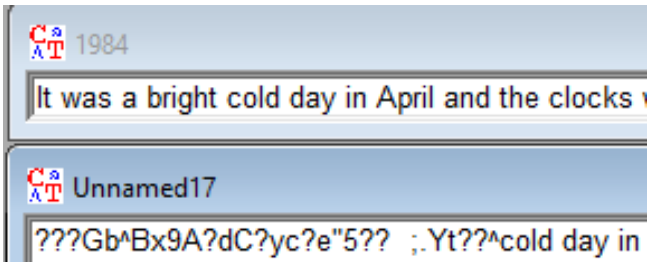
– **Zmiana jednego lub kilku bitów**



Zmiana w tekście jest bardzo podobna do tej przy ecb, jednak po przyjrzeniu się widać drobną różnicę. Nieczytelny jest blok 16 znaków w których zmieniono bit, **oraz znak w następnym bloku**. Zmieniony bit propaguje na następny blok. Wskazuje to na zależność kodowania następnego bloku od poprzedniego. Reszta tekstu pozostała nienaruszona

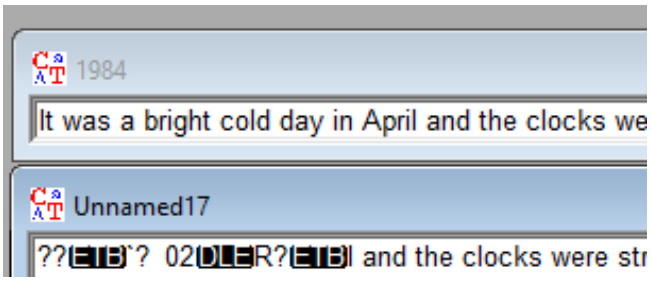
– **Dodanie/usunięcie bajtu** Po dodaniu lub usunięciu bajtu nie udało się odszyfrować krytptogramu. Dzieje się tak, ponieważ kryptogram powinien mieć długość równą wielokrotności długości bloku.

– **Dopisanie długości bloku (dopisanie losowych 32 znaków na początek kodu hex)**



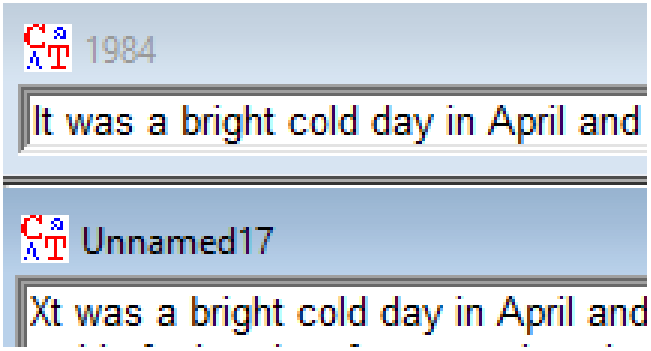
Dopisanie długości bloku spowodowało nieczytelny tekst tam gdzie został on dopisany, **oraz** nieczytelność pierwych 16 znaków oryginalnego tekstu.

– **Usunięcie długości bloku (usunięto pierwsze 32 znaki kodu hex)**



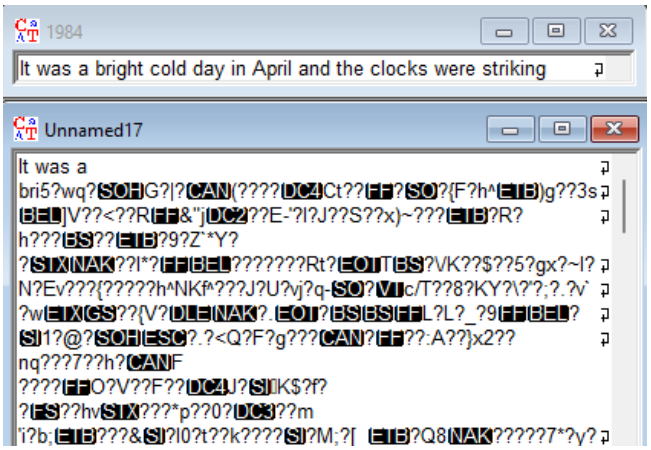
Usunięcię długości bloku usunęło pierwsze 16 znaków oraz zniekształciło kolejne 16. Potwierdza to że kodowanie bloku jest zależne od poprzedniego.

- **Tryb OFB**
  – **Zmiana jednego lub kilku bitów**



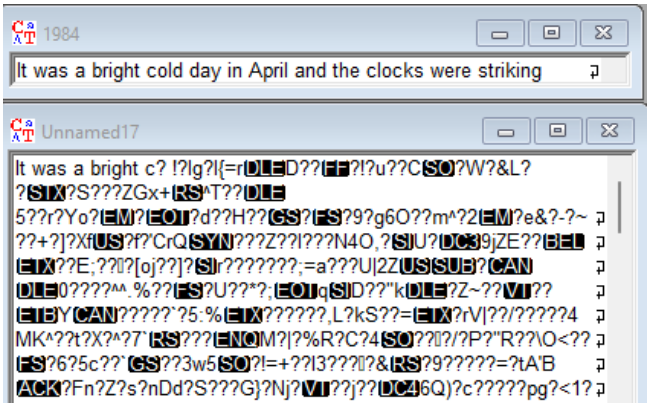
Zmiana jednego lub kilku bitów spowodowała zmianę tylko tych znaków, których bity zmieniono. Reszta tekstu pozostała czytelna.

– **Dodanie/usunięcie bajtu**



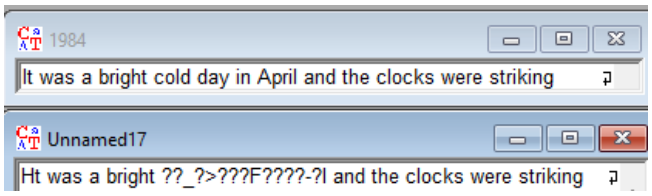
Zarówno dodanie jak usunięcie pojedynczego bajtu, spowodowało nieczytelność odszyfrowanego tekstu od miejsca, w ktorym sprowadzono zmianę.

– **Dopisanie długości bloku (dopisanie losowych 32 znaków na początek kodu hex)**



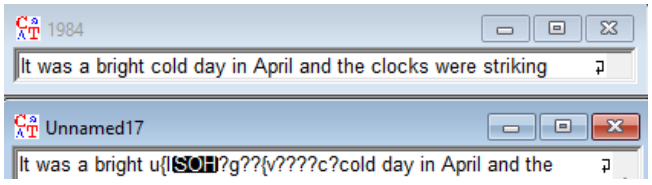
Analogiczna sytuacja ma miejsce po dodaniu długości bloku. Pierwsze wystąpienie skopiowanego bloku jest czytelne, jednak drugie, oraz cała reszta tekstu jest nieczytelna.

- **Tryb CFB**
  - **Zmiana jednego lub kilku bitów**



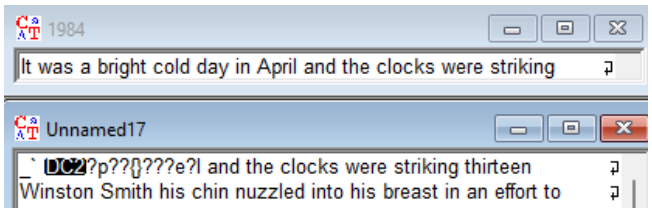
Zmiana pierwszego bitu zmieniła ten znak, tak jak w trybie OFB, jednak spowodowała także nieczytelność całego następnego bloku.

- **Dodanie/usunięcie bajtu** Dodanie i usunięcie bajtu dają taki sam efekt jak dla OFB.
- **Dopisanie długości bloku (dopisanie losowych 32 znaków na początek kodu hex)**



Dopisanie bloku sprawiło że następny blok jest nieczytelny.

- **Usunięcie długości bloku (usunięto pierwsze 32 znaki kodu hex)**



Usunięcie bloku spowodowało usunięcie go oraz spowodowało nieczytelnośc następnego.

Zadanie 2.7

Jak wyglądają kryptogramy i ich entropie dla tekstu jawnego o jednorodnej strukturze w zależności od wybranego trybu szyfrowania?

Wszystkie tryby poza ECB osiągnęły wysoki poziom entropii oraz różnorodność kryptogramu. Tryb ECB ma niską entropię i składa się z powtarzającego się bloku.

Zadanie 2.8

Jak propagują się błędy z kryptogramu do tekstu jawnego przy deszyfracji TT z przekłamanym bitem/wieloma bitami (odpowiedź zilustruj odpowiednimi przykładami).

W trybie ECB, zmiana bitów wpływa tylko na blok w którym go zmieniono. W trybie CBC, zmiana wpływa na blok oraz na następny blok. W trybie OFB, zmiana wpłynęła tylko na dany znak. W trybie CFB, zmian wpłynęła tylko na dany znak oraz cały następny blok.

Zadanie 2.9

Jak poszczególne tryby pracy algorytmów blokowych radzą sobie z utratą części wiadomości. Tzn. jaka jest możliwość odtworzenia TJ na podstawie TT z którego usunięto część zawartości (odpowiedź zilustruj odpowiednimi przykładami).

W trybie ECB utrata części wiadomości prowadzi do utraty bloku tekstu, w którym ta część się znajduje. W trybie OCB, utrata może wpłynąć na następne bloki.

Zadanie 2.10

Dla jakich zastosowań możemy wykorzystać tryby pracy: ECB, CBC?

Tryb EBC nadaje się do zastosowań gdzie każdy blok jest niezależny i niepowiązany z innymi blokami, czyli szyfrowanie danego bloku nie jest zależne od otoczenia. Pozwala to zrównoleglić szyfrowanie oraz deszyfrowanie, tym samym zwiększając szybkość tych procesów. CBC sprawdzi się w zastosowaniach, w których występują związki pomiędzy kolejnymi blokami tekstu. Tryb CBC jest bezpieczniejszy niż ECB. Jest w stanie doskonale ukryć naturę tekstu, nawet jawnego oraz jest trudniejszy w kryptoanalizie. W sytuacjach gdzie najważniejsze jest bezpieczeństwo i poufność, CFB jest znacznie lepszym rozwiązaniem.

Zadanie 2.11

W przypadku których trybów proces szyfrowania/deszyfrowania można prowadzić równolegle? W przypadku którego trybu prac wy można podzielić TT/TJ na kilka niezależnych części które będą deszyfrowane/szyfrowane równolegle na kilku komputerach, a następnie połączone dadzą ten sa wm rezultat co w przypadku realizowania całego procesu na jednym stanowisku.

- W trybie EFB można prowadzić oba procesy równolegle, ponieważ każdy blok jest szyfrowany i deszyfrowany w izolacji od otoczenia.
- W trybie CBC nie można zrównoleglić szyfrowania, ponieważ każdy blok jest zależny od poprzedniego. Deszyfrowanie można przeprowadzić równolegle.
- W trybie OFB nie da się zrównoleglić ani szyfrowania ani deszyfrowania.
- Tryb CFB, podobnie jak CBC pozwala na zrównoleglenie deszyfrowania.