Implementacja aplikacji szyfrującej i deszyfrującej pliki z wykorzystaniem kryptografii symetrycznej

Kotłowski Julian 197694

Morawska Julia 198209

Rodziewicz Olga 198421

Sekula Natalia 197913

13 lipca 2025

1 Wstęp

Sprawozdanie z projektu dotyczącego implementacji aplikacji szyfrującj i deszyfrującej pliki z wykorzystaniem kryptografii symetrycznej wykonanej w ramach przedmiotu Wprowadzenie do Cyberbezpieczeństwa. Zostanie również zaprezentowana funkcjonalność programu w różnych trybach szyfrowania, takich jak ECB, CBC oraz CTR. Dodatkowo, omówiona będzie odporność poszczególnych trybów na błędy oraz wpływ modyfikacji bajtów w zaszyfrowanym pliku na proces deszyfrowania.

2 Szyfrowanie

Szyfrowanie to proces przekształcania danych w formę nieczytelną dla osób nieuprawnionych, za pomocą określonego algorytmu oraz klucza kryptograficznego. Celem szyfrowania jest zapewnienie poufności informacji, czyli uniemożliwienie jej odczytania przez osoby trzecie. Dane mogą zostać odszyfrowane z powrotem do postaci oryginalnej tylko przez podmiot posiadający odpowiedni klucz deszyfrujący.

Wyróżnia się dwa rodzaje szyfrowania:

- **Szyfrowanie symetryczne** wykorzystuje ten sam klucz kryptograficzny do szyfrowania i deszyfrowania danych. Jego główną zaletą jest wysoka wydajność, co czyni je szczególnie przydatnym przy przetwarzaniu dużych ilości danych. Wadą jest konieczność bezpiecznego przekazania klucza drugiej stronie, co stwarza ryzyko jego przechwycenia. Przykładowe tryby szyfrowania dla algorytmu AES to ECB, CBC oraz CTR.
- **Szyfrowanie asymetryczne** opiera się na dwóch różnych kluczach: publicznym do szyfrowania oraz prywatnym do deszyfrowania. Zapewnia to większe bezpieczeństwo podczas przesyłania danych, ponieważ klucz prywatny pozostaje znany wyłącznie właścicielowi. Choć mniej wydajne,

rozwiązanie to znajduje zastosowanie m.in. w bezpiecznej wymianie kluczy. Przykładowe algorytmy to RSA i ECC.

3 Porównanie trybów szyfrowania symetrycznego

3.1 ECB (Electronic Codebook)

- Równoległe szyfrowanie bloków
- Identyczne bloki wejściowe dają identyczne szyfrogramy
- Nie zalecany do szyfrowania danych z powtarzającą się strukturą
- Każdy blok danych jest szyfrowany niezależnie tym samym kluczem
- Błąd ograniczony do jednego bloku

3.2 CBC (Cipher Block Chaining)

- Brak możliwości równoległego szyfrowania bloków
- Eliminacja powtarzalności
- Zalecany dla danych z powtarzalną strukturą
- Każdy blok wejściowy jest połączony z poprzednim szyfrogramem przed zaszyfrowaniem
- Błąd w jednym bloku wpływa na dwa bloki podczas deszyfrowania
- Wymaga unikalnego wektora inicjalizującego

3.3 CTR (Counter Mode)

- Równoległe szyfrowanie bloków
- Eliminacja powtarzalności
- Zalecany dla danych z powtarzalną strukturą
- Licznik do generowania pseudolosowego ciągu
- Błędy transmisji ograniczone do pojedynczych bloków
- Wymagana unikalność wartości licznika dla każdej sesji

4 Zarządzanie kluczami

W systemach kryptografii symetrycznej procesy szyfrowania i odszyfrowywania opierają się na wykorzystaniu wspólnego klucza, znanego zarówno nadawcy, jak i odbiorcy danych. Bezpieczne zarządzanie kluczem ma istotne znaczenie dla integralności i poufności całej infrastruktury kryptograficznej.

4.1 Generowanie kluczy

Proces generowania kluczy kryptograficznych powinien spełniać określone standardy bezpieczeństwa. Klucz symetryczny musi być tworzony z użyciem kryptograficznie bezpiecznego generatora liczb losowych (CSPRNG), w celu zapewnienia jego nieprzewidywalność i odporność na próby odgadnięcia.

CSPRNG to specjalna klasa generatorów liczb pseudolosowych zaprojektowana z myślą o zastosowaniach kryptograficznych. W odróżnieniu od zwykłych generatorów nawet jeśli napastnik pozna część wygenerowanego ciągu, nie będzie w stanie przewidzieć kolejnych lub wcześniejszych wartości.

W przypadku stosowania algorytmu AES, należy stosować klucze o długości 128, 192 lub 256 bitów, długość 256 bitów oferuje najwyższy poziom bezpieczeństwa. Odradzane jest wykorzystywanie prostych lub łatwych do przewidzenia sekwencji znaków jako kluczy, ponieważ znacząco obniża to skuteczność ochrony danych.

4.2 Wymiana kluczy

Bezpieczna wymiana kluczy to jedną z największych trudności. Klucz musi zostać dostarczony do wszystkich uprawnionych stron, z zastosowaniem bezpiecznego mechanizmu jego przekazania. Istnieje kilka metod realizacji tego zadania:

- ręczne uzgodnienie klucza. Polega na wcześniejszym ustaleniu i przekazaniu klucza w sposób niezależny od samego systemu szyfrującego. Może odbywać się np. podczas wstępnej konfiguracji systemu lub za pośrednictwem fizycznych nośników.
- bezpieczny kanał fizyczny lub logiczny. Klucz może być przekazywany za pośrednictwem zaufanego kanału komunikacyjnego np. wewnętrznej sieci, VPN czy innych form fizycznej izolacji lub kontroli dostępu.
- uzgodnienie klucza z wykorzystaniem protokołów kryptograficznych. Zamiast bezpośredniego przesyłania klucza, możliwe jest wykorzystanie odpowiednich protokołów matematycznych (np. protokołu Diffie-Hellmana), które umożliwiają dwóm stronom uzgodnienie wspólnego klucza w sposób bezpieczny, nawet jeśli komunikacja odbywa się przez niezaufany kanał.

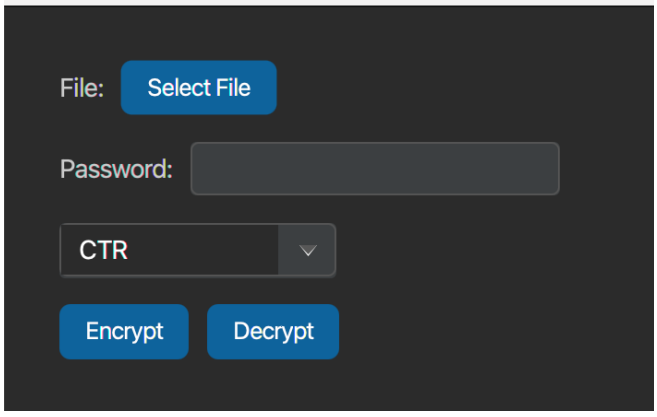
5 Rodzaje błędów i ich wpływ

- **ECB** to tryb, w którym błąd w jednym bloku zaszyfrowanych danych wpływa wyłącznie na konkretny blok po odszyfrowaniu. Pozostała część danych nie zostaje naruszona. Jednak ten tryb nie ukrywa wzorców w danych, jest on mniej bezpieczny.
- **CBC** to tryb, w którym błąd w jednym bloku szyfrogramu powoduje zniekształcenie: całkowite odszyfrowanie błędnie bloku bieżącego oraz częściowe zakłócenie kolejnego bloku. Zapewnia większą odporność na analizę wzorców, ale też powoduje propagację błędu.
- **CTR** to tryb, którym generowany jest strumień klucza niezależnie od danych. Błąd w transmisji wpływa tylko na konkretne bajty zaszyfrowanych danych, pozostała część pozostaje nienaruszona. Tryb CTR jest odporny na propagację błędów i dobrze sprawdza się w środowiskach podatnych na zakłócenia, takich jak transmisje sieciowe.

6 Prezentacja praktyczna

6.1 Interfejs aplikacji

- **Przycisk "Select File"** umożliwia użytkownikowi wybór pliku z systemu lokalnego, który ma zostać zaszyfrowany lub odszyfrowany.
- **Pole "Password"** umożliwia użytkownikowi wprowadzenie hasła, z którego generowany jest klucz szyfrujący za pomocą funkcji PBKDF2.
- **Lista trybu szyfrowania** umożliwia użytkownikowi wybór jednego z trzech sposobów szyfrowania AES pliku: ECB, CBC lub CRT.
- **Przycisk "Encrypt"** rozpoczyna proces szyfrowania piseli wybranego pliku.
- **Przycisk "Decrypt"** rozpoczyna proces deszyfrowania zaszyfrowanego pliku.



Rysunek 1: Interfejs aplikacji szyfrującej i deszyfrującej pliki

6.2 Przebieg szyfrowania

Proces szyfrowania rozpoczyna się od wyboru pliku, trybu AES (ECB, CBC lub CTR) oraz hasła użytkownika. Na podstawie hasła generowany jest klucz szyfrujący AES 128-bit z użyciem algorytmu PBKDF2 z HMAC-SHA256, wykorzystującego ustaloną sól i określoną liczbę iteracji. Następnie tworzony jest plik wyjściowy z rozszerzeniem .enc.

W trybach CBC i CTR generowany jest losowy wektor inicjalizujący (IV), który zapisywany jest na początku zaszyfrowanego pliku. Dane z pliku wejściowego są przetwarzane strumieniowo w blokach i szyfrowane bezpośrednio podczas odczytu. Tryb ECB nie wykorzystuje wektora inicjalizującego przez co charakteryzuje się najniższym poziomem bezpieczeństwa. CBC używa paddingu (mechanizm dopełniania, gdy dane wejściowe nie są wielokrotnością długości bloku algorytmu) i losowego IV, natomiast CTR działa bez paddingu, pełniąc funkcję szyfrowania strumieniowego.

6.3 Przebieg deszyfrowania

Proces deszyfrowania rozpoczyna się w podobny sposób co proces szyfrowania pliku: wybór pliku, trybu szyfrowania oraz hasła użytkownika. Następnie generowany jest klucz deszyfrujący przy użyciu algorytmu PBKDF2 z HMAC-SHA256, tej samej soli oraz liczby iteracji co w procesie szyfrowania. Otwierany jest plik wejściowy zawierający dane zaszyfrowane.

W trybach CBC i CTR pierwsze 16 bajtów pliku odczytywane są jako wektor inicjalizujący (IV). Algorytm deszyfrujący AES konfigurowany jest zgodnie z trybem oraz odpowiednim paddingiem (PKCS5 w CBC i ECB, brak paddingu w CTR). Dane są przetwarzane strumieniowo: odczytywanie danych z pliku, odszyfrowywanie w czasie rzeczywistym i zapisanie do nowego pliku z rozszerzeniem .dec.

6.4 Modyfikacja wybranych bajtów szyfrogramu

Modyfikacja pliku zaszyfrowanego odbywa się poprzez losową zmianę jego zawartości na poziomie bajtowym. Cały plik wejściowy jest wczytywany do pamięci jako tablica bajtów. Następnie realizowana jest wskazana liczba modyfikacji poprzez wielokrotne losowanie pozycji w pliku z pominięciem pierwszych i ostatnich 100 bajtów (ochrona wektora inicjalizującego (IV), nagłówków). Dla każdej wylosowanej pozycji oryginalna wartość bajtu zostaje nadpisana wartością 0xFF.

```
txt
Co dzień nas gna
W nowe strony zadyszany czas
Sto dat

Sto spraw
Wciąga nas
Gna nas...
I moje dni
Wszechobecny pośpiech, czasu znak
Naznaczył i...
Może przez to tak
Lubię...
Lubię wracać tam, gdzie byłem już
Pod ten balkon pełen pnących róż
Na uliczki te, znajome tak
Do znajomych drzwi
Pukać, myśląc, czy
Czy nie stanie w nich czasami
Ta dziewczyna z warkoczami
Lubię wracać w strony, które znam
Po wspomnienia zostawione tam
By się przejrzeć w nich, odnaleźć w nich
```

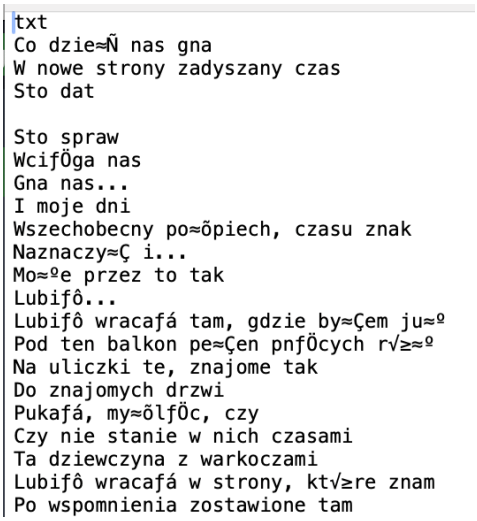
Rysunek 2: Oryginalny tekst bez zaszumień

```
txt
Co dzie≈Ñ nas gna
W nowe strony zadyszany czas
Sto dat

Sto spraw
WcifÔga nas
Gna nas...
I mçïiß™ª89/€Ltn
'ecny po≈õpiech, czasu znak
Naznaczy≈Ç i...
Mo≈ºe¨DçHÿii.d≠ ¨Ú4bifô...
Lubifô wracafá tam, gdzie by≈Çem ju≈º
Pod ten balkon pe≈Çen pnfÖcych r√≥◊ñQZ(p7µ;4OBÚ©?fi, znajome tak
Do znajomych drzwi
Pukafá, my≈õlfÖc, czy
Czy nie stanie w nich cza…«TVoÿ)Ã«ûıórd:na z warkoczami
Lubifô wracafá w strony, kt√≥re …ÙÆ„‹o∞M9¬är€nia zostawione tam
```

Rysunek 3: Tekst po odszyfrowaniu ECB z zaszumieniami

W trybie ECB każda zmiana w szyfrogramie, nawet pojedynczego bajtu, skutkuje kompletnym zniekształceniem całego bloku o długości 16 bajtów. Po odszyfrowaniu pliku z losowo zaszumionymi bajtami, widoczne są bardzo poważne zniekształcenia treści. Cały blok staje się nieczytelny, a uszkodzony fragment traci jakikolwiek sens językowy. Tryb ten jest wyjątkowo nieodporny na nawet drobne zakłócenia transmisji lub modyfikacje danych.

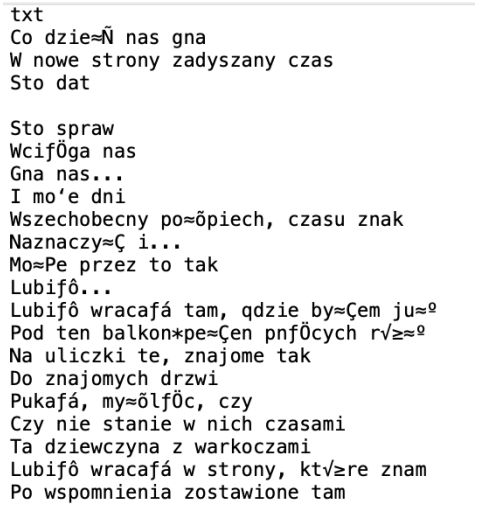


```
txt
Co dzie≈Ñ nas gna
W nowe strony zadyszany czas
Sto dat

Sto spraw
WcifÖga nas
Gna nas...
I moje dni
Wszechobecny po≈õpiech, czasu znak
Naznaczy≈Ç i...
Mo≈ºe przez to tak
Lubiƒô...
Lubiƒô wracafá tam, gdzie by≈Çem ju≈º
Pod ten balkon pe≈Çen pnƒÖcych r√≥≈º
Na uliczki te, znajome tak
Do znajomych drzwi
Pukafá, my≈ölfÖc, czy
Czy nie stanie w nich czasami
Ta dziewczyna z warkoczami
Lubiƒô wracafá w strony, kt√≥re znam
Po wspomnienia zostawione tam
```

Rysunek 4: Tekst po odszyfrowaniu CBC z zaszumieniami

W trybie CBC zmiana jednego bajtu zaszyfrowanego tekstu wpływa na dwa odszyfrowane bloki: bieżący oraz następny. Skutkiem jest, że jeden z bloków jest silnie zniekształcony, natomiast drugi zawiera drobniejsze błędy. W porównaniu do ECB, CBC wykazuje nieco lepszą odporność na błędy. Pozostała część tekstu poza uszkodzonymi blokami jest w dużej mierze zachowana. Pomimo widocznych zaszumień i pogarszonej jakość treści, całościowy sens tekstu da się odczytać.



```
txt
Co dzie≈Ñ nas gna
W nowe strony zadyszany czas
Sto dat

Sto spraw
WcifÖga nas
Gna nas...
I mo‘e dni
Wszechobecny po≈õpiech, czasu znak
Naznaczy≈Ç i...
Mo≈Pe przez to tak
Lubiƒô...
Lubiƒô wracafá tam, qdzie by≈Çem ju≈º
Pod ten balkon*pe≈Çen pnƒÖcych r√≥≈º
Na uliczki te, znajome tak
Do znajomych drzwi
Pukafá, my≈ölfÖc, czy
Czy nie stanie w nich czasami
Ta dziewczyna z warkoczami
Lubiƒô wracafá w strony, kt√≥re znam
Po wspomnienia zostawione tam
```

Rysunek 5: W

Tryb CTR wykazuje najwyższą odporność na zaszumienia spośród analizowanych trybów. Szyfrowanie i deszyfrowanie polega na operacji XOR z generowanym strumieniem klucza, każdy zaszumiony bajt szyfrogramu przekłada się dokładnie na jeden błędny znak w odszyfrowanym tekście a pozostała treść pozostaje nienaruszona. Pojedyncze zniekształcenia umożliwiają zrozumienia całości tekstu. Dzięki dobrej odporności na zakłócenia CTR nadaje się do szyfrowania danych, gdzie mogą wystąpić błędy transmisji.