Dokumentacja projektowa

Szyfrowanie plików wraz z przekazaniem klucza sesyjnego – protokół przesyłania wiadomości z sesyjnym kluczem szyfrującym

Anna Ojdowska 160823

Katarzyna Trzcińska 160871

Spis treści

[1. Założenia wstępne 2](#_Toc509085720)

[2. Projekt UI i opis interakcji z użytkownikiem 3](#_Toc509085721)

[3. Struktura pliku wynikowego 6](#_Toc509085728)

[4. Źródła 6](#_Toc509085729)

Gdańsk, 17.03.2018 r.

# Założenia wstępne

Podstawową funkcjonalnością aplikacji jest szyfrowanie i deszyfrowanie plików za pomocą algorytmu AES (Advanced Encyrption Standard) w jednym z czterech trybów do wyboru: ECB, CBC, CFB, OFB.

Algorymt AES jest symetrycznym szyfrem blokowym, tzn., że do szyfrowania i deszyfrowania używany jest ten sam klucz. AES bazuje na zasadzie, zwanej siecią substytucji-permutacji. Posiada określony rozmiar bloku – 128 bitów, natomiast rozmiar klucza wynosi 128, 192, lub 256 bitów. Funkcja substytucyjna ma bardzo oryginalną konstrukcję, która uodparnia ten algorytm na znane ataki kryptoanalizy różnicowej i liniowej.

Rozmiar klucza używany w algorytmie określa liczbę powtórzeń transformacji, które przekształcają dane wejściowe w dane wyjściowe.

Liczba powtórzeń cykli w zależności od długości klucza:

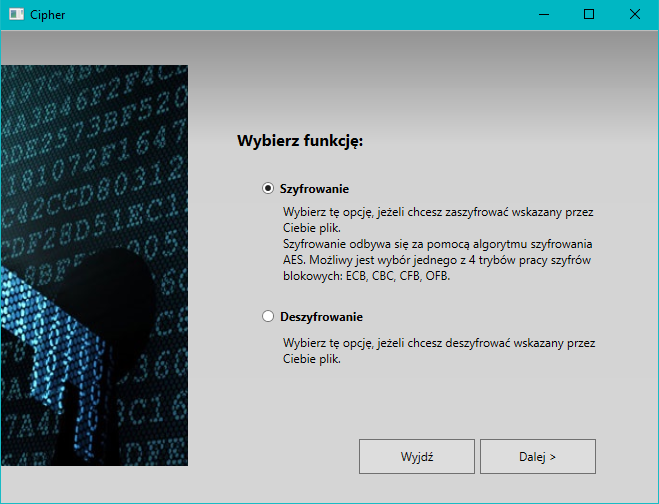
* 10 cykli powtórzeń dla klucza 128-bitowego,
* 12 cykli powtórzeń dla klucza 192-bitowego,
* 14 cykli powtórzeń dla klucza 256-bitowego.

Klucze sesyjne będą generowane przez generator liczby pseudolosowych, którego wartością początkową będzie aktualny czas systemu operacyjnego. Prywatne i publiczne klucze użytkowników zostaną wygenerowane za pomocą algorytmu RSA. Klucz sesyjny będzie zaszyfrowany kluczem publicznym RSA zamierzonego odbiorcy. Klucze prywatne RSA będą przechowywane w postaci zaszyfrowanej w trybie ECB, a kluczem szyfrującym będzie skrót hasła dostępu do klucza prywatnego danego użytkownika uzyskany z hasła za pomocą funkcji skrótu SHA-256.

Użytkownik może wybrać więcej niż jednego odbiorcę swojego szyfrogramu. Szyfrogram zostanie poprawnie odszyfrowany tylko, gdy dokona tego upoważniony odbiorca swoim kluczem prywatnym. Jeśli inny użytkownik będzie próbował dokonać nieupoważnionej deszyfracji nie zostanie powiadomiony o braku uprawnień do treści, ale otrzyma niepoprawnie odszyfrowany plik.

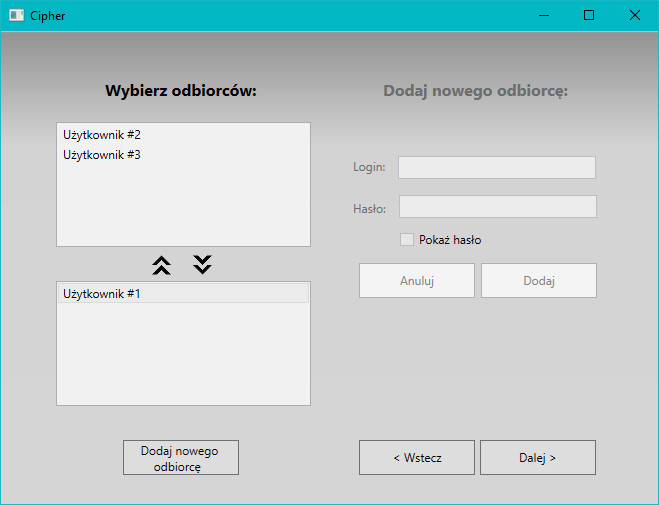
# Projekt UI i opis interakcji z użytkownikiem

## Ekran startowy - szyfrowanie



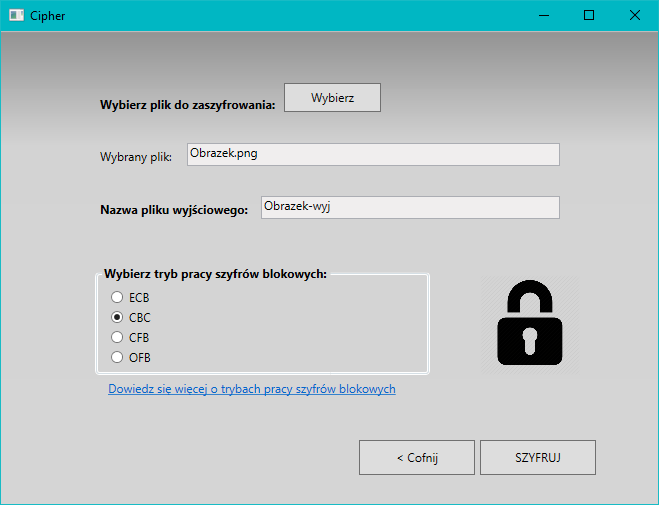
*Ekran umożliwia wybranie jednej z dwóch funkcji: szyfrowania lub deszyfrowania plików.*

## 2.2 Ekran wyboru odbiorców szyfrogramu



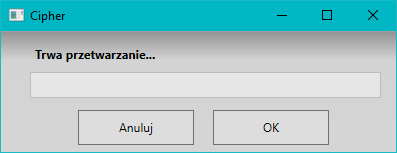
*Po wybraniu opcji „Szyfrowanie” użytkownik może wybrać odbiorców swojego docelowego szyfogramu spośród istniejących już kont lub dodać nowego odbiorcę. Panel rejestracji nowego użytkownika uaktywni się po wciśnięciu przycisku „Dodaj nowego odbiorcę”.*

## 2.3 Ekran wyboru parametrów szyfrowania



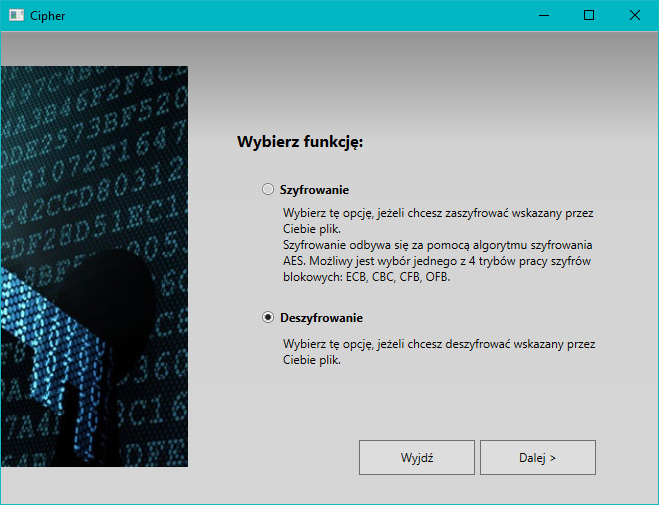
*Gdy użytkownik zatwierdzi wybór odbiorców szyfrogramu, zostaje przeniesiony do ekranu, w którym ustala dane wejściowe szyfrowania: plik wejściowy, nazwę pliku wyjściowego (rozszerzenie zostanie zachowane), tryb szyfrowania. Aplikacja udostępnia również hiperlink do strony internetowe, gdzie opisane są szczegóły dotyczące każdego z możliwych do wybrania trybów szyfrowania.*

## 2.4 Pasek postępu przetwarzania

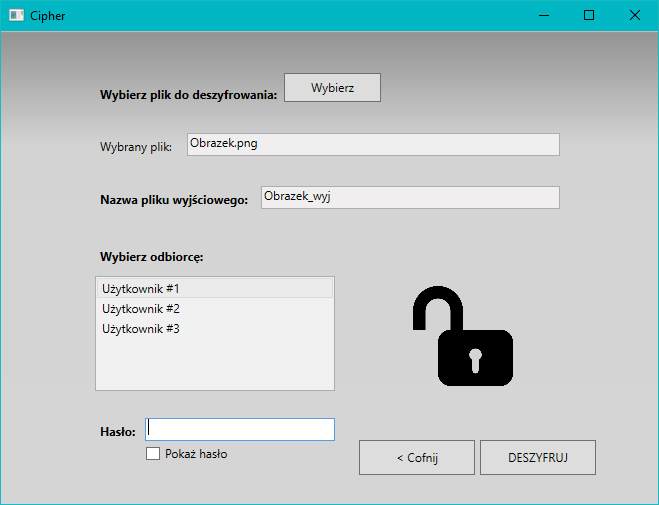


*Gdy użytkownik zatwierdzi dane wejściowe algorytmu przyciskiem „SZYFRUJ”, pojawia się pasek postępu przetwarzania pliku wejściowego na szyfrogram. W każdej chwili użytkownik może przerwać operację naciskając przycisk „Anuluj”, powróci wtedy do ekranu wybierania parametrów szyfrowania. Po zakończonej operacji użytkownik po wybraniu przycisku „OK” zostanie ponownie przeniesiony do ekranu startowego.*

## 2.5 Ekran startowy – deszyfrowanie



## 2.6 Wybór parametrów deszyfrowania



*Po wyborze opcji deszyfrowania, użytkownik zostaje przeniesiony do panelu, gdzie może wybrać plik, który chciałby odszyfrować i podać nazwę pliku wyjściowego (rozszerzenie zostanie zachowane). Musi także wskazać siebie na liście potencjalnych odbiorców pliku i potwierdzić swoją tożsamość hasłem podanym przy rejestracji. Jeśli użytkownik nie jest na liście zaaprobowanych odbiorców lub poda niepoprawne hasło nie zostanie o tym poinformowany i otrzyma błędnie odszyfrowany plik.*

# Struktura pliku wynikowego

Plik wynikowy zawierający zaszyfrowane dane będzie składał się z nagłówka w standardzie XML zawierającego dane niezbędne do poprawnego odszfrowania pliku oraz z właściwej treści, zawierającej zaszyfrowane dane. Przykład takiego pliku znajduje się poniżej.

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<EncryptedFile>

<EncryptedFileHeader>

<Algorithm>AES</Algorithm>

<KeySize>192</KeySize>

<BlockSize>128</BlockSize>

<CipherMode>ECB</CipherMode>

<IV>wektor początkowy</IV>

<ApprovedUsers>

<User>

<Login>Uzytkownik #1</Login>

<SessionKey>zaszyfrowany klucz sesyjny</SessionKey>

</User>

<User>

<Login>Uzytkownik #2</Login>

<SessionKey>zaszyfrowany klucz sesyjny</SessionKey>

</User>

<User>

<Login>Uzytkownik #3</Login>

<SessionKey>zaszyfrowany klucz sesyjny</SessionKey>

</User>

...

</ApprovedUsers>

</EncryptedFileHeader>

<Body>

Zaszyfrowana zawartość pliku

</Body>

</EncryptedFile>

# Źródła

1. https://pl.wikipedia.org/wiki/Advanced\_Encryption\_Standard