*21.05.2014*

**Paweł Troka, nr indeksu: 132334, grupa 1A**

# Algorytm Rijndael

Krótki raport

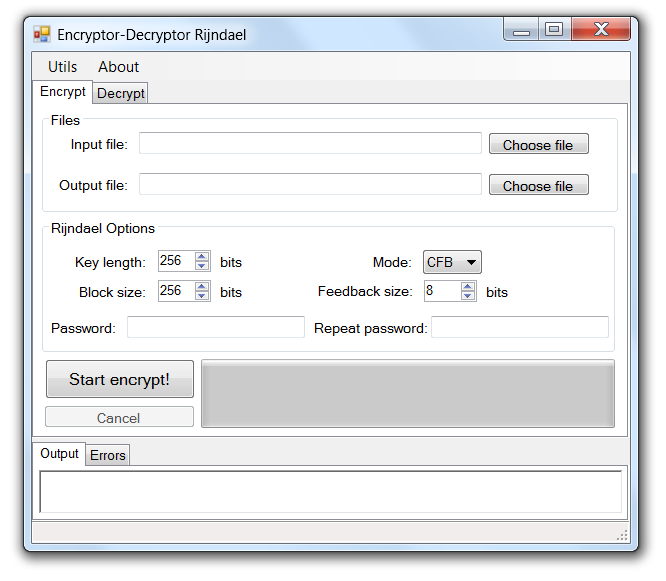
Bezpieczeństwo Systemów Komputerowych - projekt

1. **Informacje o temacie zadania**

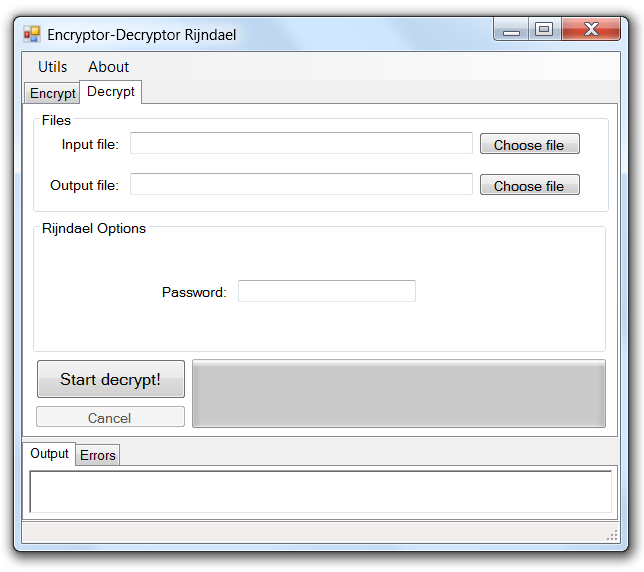
Tematem zadania jest realizacja aplikacji szyfrującej i deszyfrującej dane z użyciem algorytmu Rijndael. Została ona zrealizowana w języku C#, na platformę .NET z użyciem technologii WinForms.

1. **Kształt interfejsu użytkownika**

Interfejs użytkownika zrealizowany we wcześniej wspomnianym WinForms, został zaprojektowany z uwzględnieniem maksymalnego wykluczenia błędów użytkownika. Użyte kontrolki m.in. zapobiegają wprowadzania błędnych danych, dodatkowo po stronie logiki aplikacji dane wprowadzone są ponownie sprawdzane pod kątem poprawności. Zaplanowano wysoką responsywność aplikacji (m.in. wykonywanie szyfrowania/deszyfrowania w oddzielnym wątku) oraz łatwość użycia.



Rysunek 1.0.1 Widok interfejsu dla zakładki odpowiadającej szyfrowaniu plików



Rysunek 1.0.2 Widok interfejsu dla zakładki odpowiadającej deszyfrowaniu plików

1. **Struktura pliku wynikowego**

Zaprojektowana struktura pliku wynikowego jest silnie zainspirowana proponowanym rozwiązaniem w pliku „Instrukcja projektowa do projektu 1”, wykorzystane są m.in. wszystkie, potrzebne dla tej aplikacji proponowane tam nazwy węzłów xml-owych itp.

<EncryptedFileHeader>

<Algorithm>nazwa algorytmu</Algorithm>

<CipherMode>tryb pracy</CipherMode>

<BlockSize>rozmiar bloku</BlockSize>

<FeedbackSize>rozmiar podbloku</FeedbackSize>

<KeySize>rozmiar klucza</KeySize>

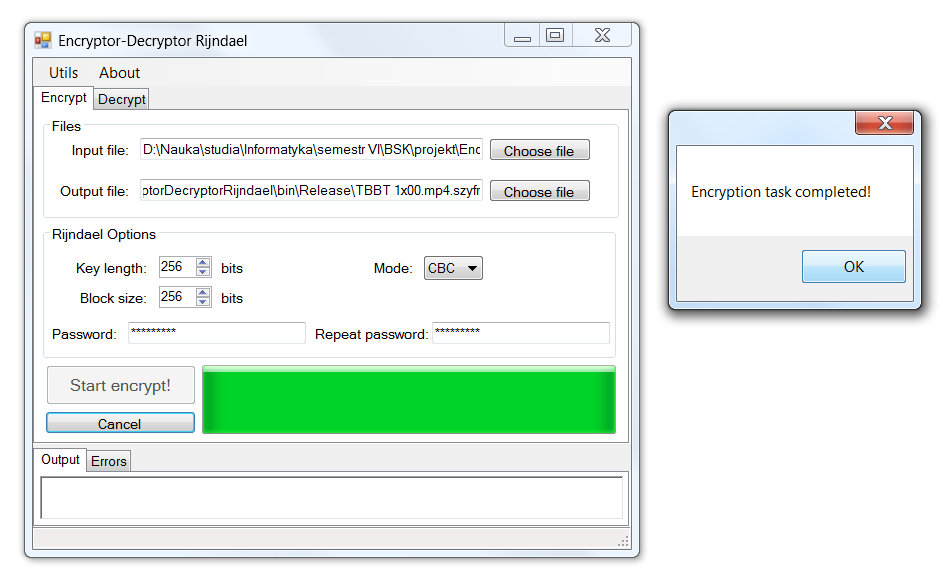
<EncryptedKey>klucz zaszyfrowany skrótem hasła</EncryptedKey>

<IV>wektor początkowy</IV>

</EncryptedFileHeader>

{szyfrogram}

1. **Testy**
2. Wybrano wszystkie parametry jak domyślnie w interfejsie, wpisano dwa razy hasło „alamakota”, jako input file wybrano plik wideo w formacie mp4 o rozmiarze 51,9MB i nazwie „TBBT 1x00.mp4”, jako output file wybrano tę samą scieżkę i nazwę pliku wejściowego z dopisanym rozszerzeniem „.szyfr”. Następnie kliknięto przycisk „Start encrypt!”. W przeciągu zaledwie kilkunastu sekund pasek postępu wypełnił się do końca i ukazał się komunikat o zakończeniu szyfrowania.



Rysunek 1.0.3 Końcowy stan interfejsu dla testu a

Uzyskano również plik wynikowy o niemal identycznym rozmiarze oraz strukturą zgodną z zaplanowaną.

<EncryptedFileHeader>

  <Algorithm>Rijndael</Algorithm>

  <CipherMode>CBC</CipherMode>

  <BlockSize>256</BlockSize>

  <FeedbackSize>8</FeedbackSize>

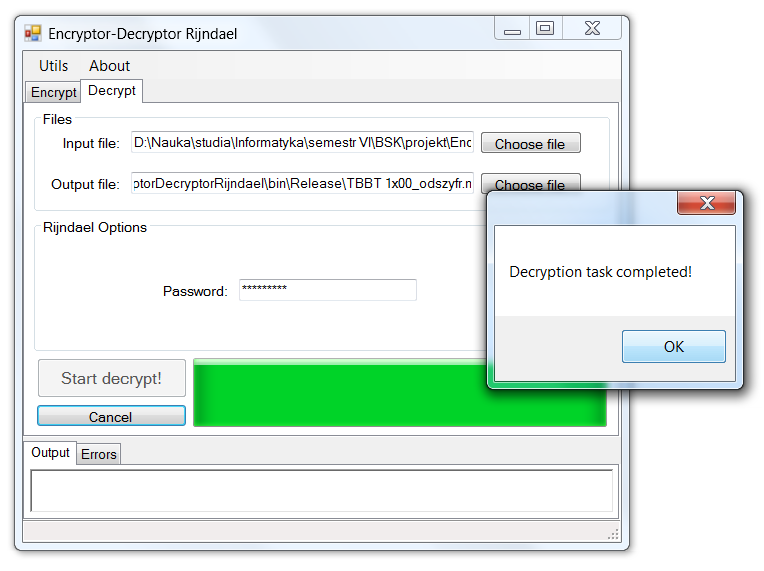
  <KeySize>256</KeySize>

  <EncryptedKey>188.156.7.185.239.135.232.106.120.161.197.58.186.32.92.120.164.12.233.223.168.1.47.57.64.146.227.28.49.233.143.122.213.32.185.71.154.123.177.40.152.158.36.49.175.137.5.134.62.70.198.37.187.32.188.70.169.17.233.172.63.211.155.159</EncryptedKey>

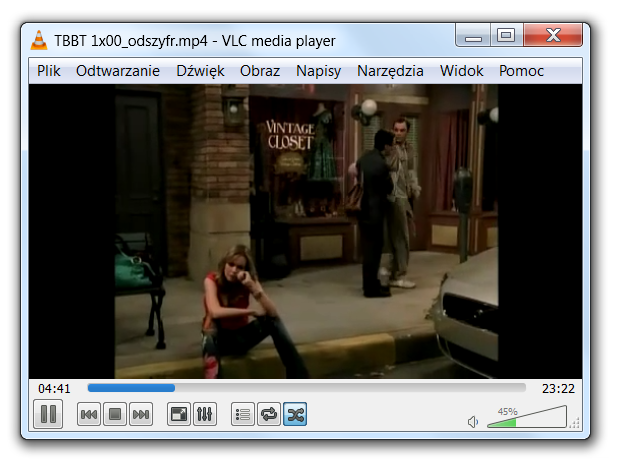
  <IV>48.145.227.61.127.60.72.126.89.47.251.123.249.195.39.175.39.94.81.249.89.119.148.145.93.230.21.73.175.232.163.118</IV>

</EncryptedFileHeader>QĹŔÔť‡  
ťŮŃhŢ1bĐ$`ĹpI{dalsza\_część\_szyfrogramu}

1. Wybrano zakładkę “decrypt” po czym wpisano hasło „alamakota” oraz został wskazany plik „TBBT 1x00.mp4.szyfr” utworzony w wyniku testu a. W polu „Output file” wpisano identyczną ścieżkę jak dla pliku zaszyfrowanego, zmieniono tylko nazwę pliku z „TBBT 1x00.mp4.szyfr” na „TBBT 1x00\_odszyfr.mp4”. Następnie kliknięto na przycisk „Start decrypt!”. W przeciągu kilku sekund pojawił się komunkat o zakończonym deszyfrowaniu, otwarto odszyfrowany plik i okazało się, że plik wideo działa.



Rysunek 1.0.4 Stan końcowy interfejsu dla testu b



Rysunek 1.5 Uruchomiony, działający plik wideo po odszyfrowaniu