Politechnika Gdańska
Wydział Elektroniki,
Telekomunikacji i Informatyki
Katedra Architektury Systemów
Komputerowych
Al. Gabriela Narutowicza 11/12 80-952 Gdańsk
tel. (58) 347-12-30 tel./fax (58) 347-28-63

Bezpieczeństwo Systemów Komputerowych

Instrukcja projektowa do projektu 1 – Szyfrowanie plików wraz z przekazaniem klucza sesyjnego – protokół przesyłania wiadomości z sesyjnym kluczem szyfrującym

Opracował: dr inż. Piotr Szpryngier

Gdańsk 2017

Wprowadzenie. Cel zajęć praktycznych.

Wymagania stawiane studentom

- znajomość uruchamiania programów w środowisku WINDOWS i/lub LINUX
- umiejętność programowania C, C++, Java, ewent. .NET
- umiejętność zaprojektowania i wykonania graficznego interfejsu użytkownika.

Stosowane narzędzia i technologie

- narzędzia programistyczne dostępne w laboratoriach,
- implementacje algorytmów blokowych i RSA dostępne w sieci Internet,
- przeglądarka WWW.

Materiały wprowadzające i pomocnicze

- dostępne materiały wykładowe: https://enauczanie.pg.gda.pl/moodle/
- podręczniki dostępne w czytelni.

Cel zajęć

- poznanie uczestników zajęć z protokołem przesyłania wiadomości zaszyfrowanych razem z kluczem sesyjnym,
- poznanie i utrwalenie wiedzy nt. trybów pracy szyfrów blokowych, zarządzania kluczami oraz cech różnych algorytmów szyfrujących.

Przebieg zajęć.

1. Analiza wymagań postawionych przez prowadzącego zajęcia,
2. Projekt struktury pliku wynikowego,
3. Projekt interfejsu użytkownika,
4. Wybór metody generowania klucza sesyjnego,
5. Implementacja aplikacji i testy,
6. Podczas zaliczania przedłożenie raportu końcowego
7. Zasady oceniania

Ad. 1
Celem wyznaczonym studentowi jest zaprojektowanie, implementacja i uruchomienie aplikacji służącej do szyfrowania i odszyfrowania plików.

Wymagania

- interfejs użytkownika powinien umożliwiać swobodny wybór pliku wejściowego oraz wybór nazwy pliku wynikowego,
- praca w czterech trybach szyfrowania: ECB,CBC,CFB,OFB; opcjonalnie można także użyć trybu licznikowego;
- dla trybu CFB i OFB wybór długości podbloku mniejszego od długości bloku algorytmu – kolejne długości podbloku powinny być potęgą liczby 2 lub wielokrotnością bajtu;
- należy opracować (zaadaptować) strukturę pliku wynikowego;
- technologia wykonania – dowolna;
- należy dobrać dobrej jakości generatory pseudolosowe do generowania kluczy sesyjnych. Wartością początkową generatora powinny być przypadkowe ciągi binarne pobrane z otoczenia, np. aktualny czas systemowy (32 lub 64 bity), numer sektora dyskowego z ostatniej transmisji, wskazanie kursora myszki, itp.
- klucz sesyjny powinien być zaszyfrowany kluczem publicznym RSA zamierzonego odbiorcy;
- W przypadku użycia RSA do transportu klucza sesyjnego może wystąpić większa liczba odbiorców tego samego szyfrogramu;
- Klucze prywatne i publiczne powinny być przechowywane oddzielnie (w odrębnych katalogach);
- W przypadku użycia RSA aplikacja powinna udostępnić listę odbiorców pliku; następnie odbiorca powinien wskazać siebie na tej liście i na tej podstawie automatycznie jego nazwa powinna zostać powiązana z kluczem prywatnym;
- W przypadku użycia RSA klucze prywatnie MUSZĄ być przechowywane w postaci zaszyfrowanej w trybie ECB, a kluczem szyfrującym jest skrót hasła (np. uzyskanym z hasła za pomocą funkcji SHA-1, SHA-256 lub inna dobrej jakości funkcja skrótu) dostępu do klucza prywatnego danego użytkownika,
- Funkcja skrótu powinna być wskazana w dokumentacji projektu;
- należy użyć implementacji algorytmów szyfrowania dostępnych w Internecie;
- uruchomienie projektu na innym komputerze nie powinno wymagać konieczności dodatkowych uzupełnień, czyli powinna wystarczyć obecność środowiska np. MS Visual Studio lub środowiska uruchomieniowego Java.

Ad. 2

Możliwe są dwa zasadnicze sposoby zarządzania kluczem sesyjnym:

a) klucz sesyjny zostaje zaszyfrowany z użyciem tego samego algorytmu w trybie ECB oraz kluczem szyfrującym klucz sesyjny w postaci hasła dostępu do pliku. W niniejszym projekcie nie będziemy korzystać z tej metody.

b) klucz sesyjny zostaje zaszyfrowany z użyciem algorytmu RSA i klucza publicznego zamierzonego odbiorcy (odbiorców) pliku szyfrogramu. Może wystąpić więcej jak jeden odbiorca tego samego pliku. Plik wynikowy może mieć np. taką strukturę np. napisaną w XML:

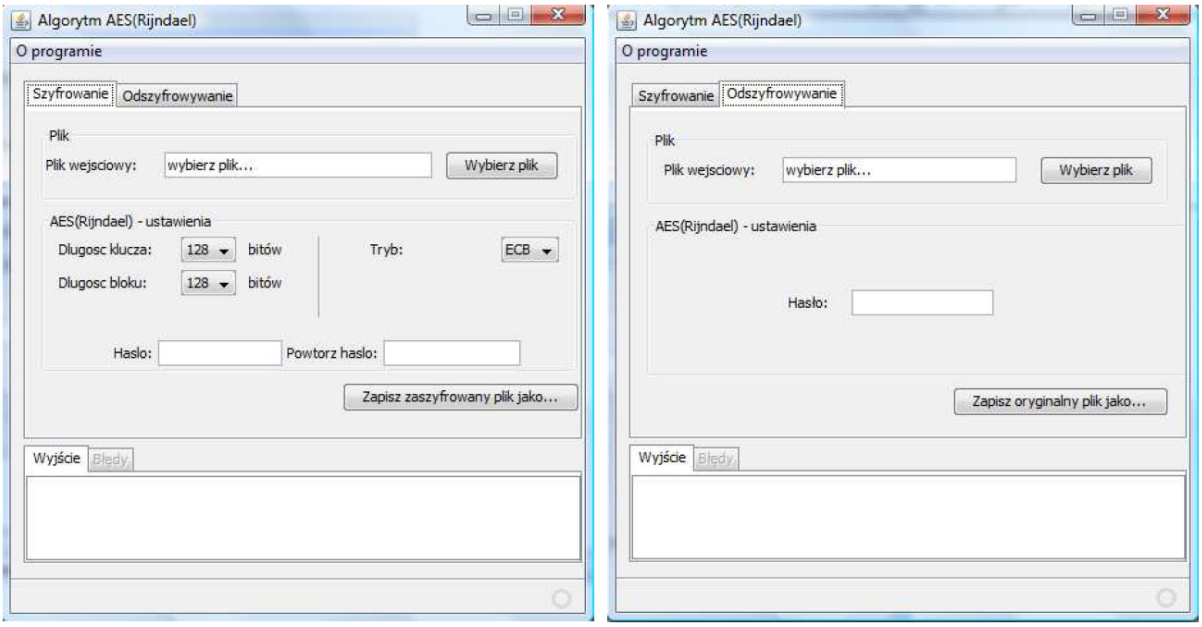
```
<EncryptedFileHeader>
        <Algorithm>nazwa</Algorithm>
        <KeySize>rozmiar</KeySize>
        <BlockSize>rozmiar</BlockSize>
        <CipherMode>TRYB</CipherMode>
        <IV>wektor_początkowy</IV>
        <ApprovedUsers>
                <User>
                <Email>adress_e-mail</Email>
                <SessionKey>klucz sesyjny zaszyfrowany kluczem publicznym
                </SessionKey>
                </User>
                …
        </ApprovedUsers>
</EncryptedFileHeade>
Zaszyfrowane dane
```

Zamiast adresu e-mail może wystąpić dowolny inny identyfikator.

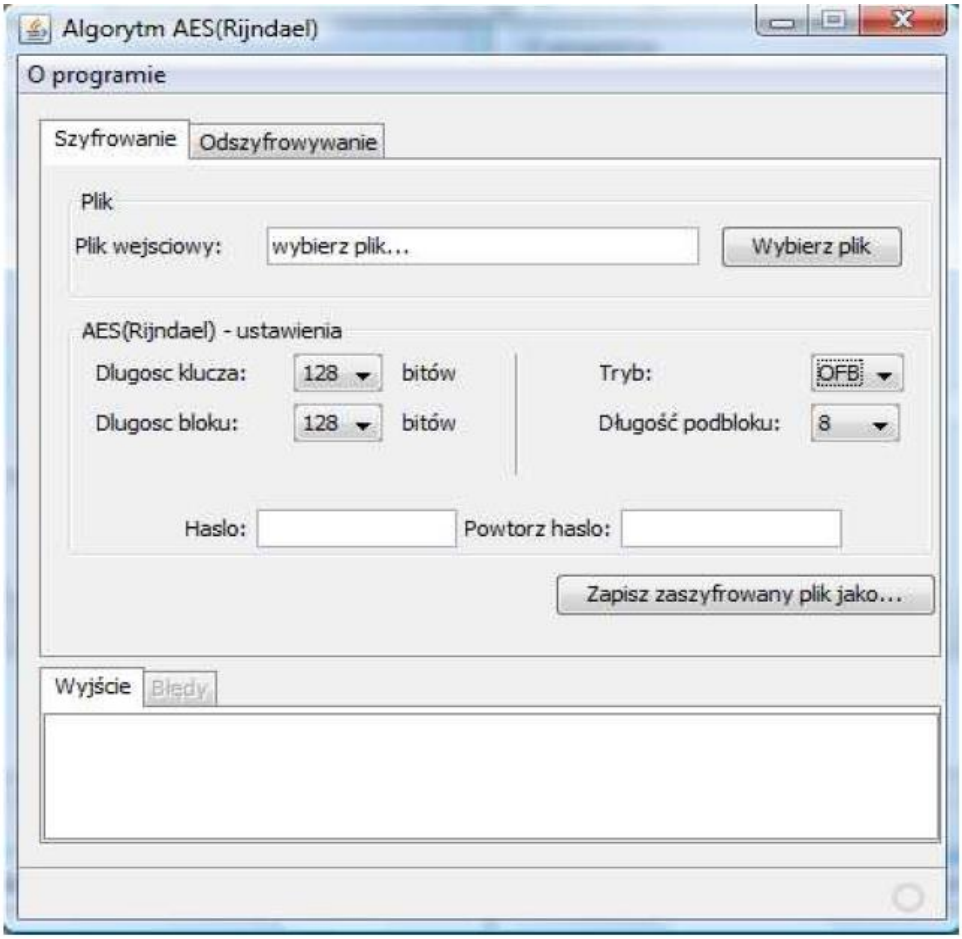
Ad. 3

Student powinien zaprojektować interfejs użytkownika, uwzględniając wymagania podane w p. 1. Technika wykonania – dowolna.

Przykłady rozwiązań interfejsu użytkownika:



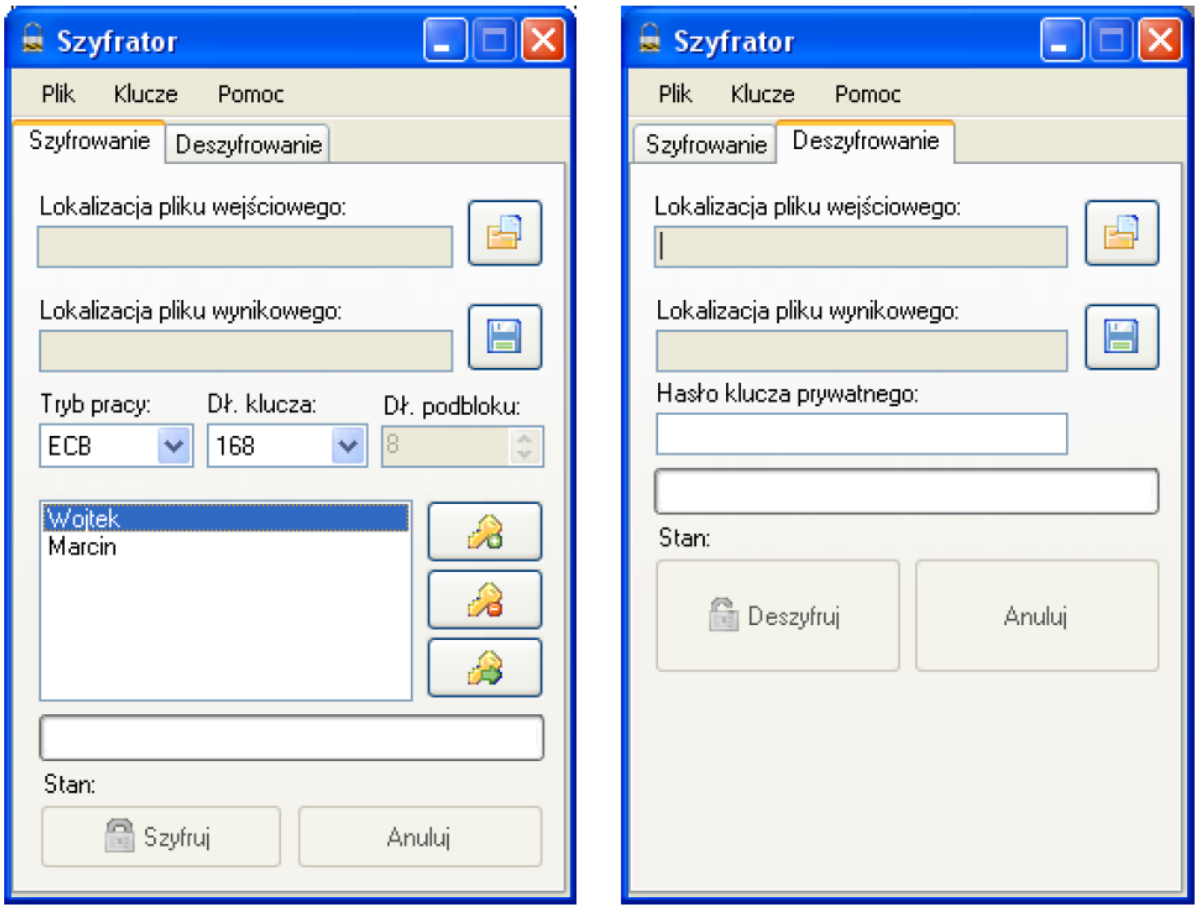
Rys. 1 Przykład interfejsu dla algorytmu Rijndael zrealizowanego w środowisku Java Netbeans (Windows) z wykorzystaniem biblioteki kryptograficznej Bouncy Castle (http://www.bouncycastle.org/java.html).



Rys. 2 Przykład interfejsu w środowisku Java Beans z możliwością wyboru długości podbloku w trybach CFB i OFB podczas konfiguracji operacji szyfrowania.

Przykładowa struktura nagłówka ma postać:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EncryptedFile xmlns="http://www.studiofresh.net/cipher">
<Algorithm>AES</Algorithm>
<CipherMode>tryb szyfrowania</CipherMode>
<BlockSize>rozmiar bloku</BlockSize>
<SegmentSize>rozmiar podbloku</SegmentSize>
<KeySize>rozmiar klucza</KeySize>
<IV>wektor początkowy</IV>
      <ApprovedUsers>
            <User>
            <Email>adress_e-mail</Email>
            <SessionKey>klucz sesyjny zaszyfrowany kluczem publicznym
            </SessionKey>
            </User>
            …
      </ApprovedUsers>
</EncryptedFile>
{szyfrogram}
```



Rys. 3 Przykład interfejsu aplikacji szyfrującej dla kombinacji algorytmów 3DES+RSA z możliwością wyboru trybu pracy, długości podbloku oraz wskazania wybranych odbiorców szyfrogramu, zrealizowanego w środowisku Microsoft Visual Studio (C# na platformie .NET). **Uwaga**: po prawej stronie należy umieścić dodatkowo możliwość wskazania lokalizacji zaszyfrowanego klucza prywatnego odbiorcy.

Zasadniczo aplikacja powinna automatycznie zapewnić wskazanie lokalizacji klucza prywatnego odbiorcy po wybraniu identyfikatora odbiorcy z listy.

Uwaga! Generowanie par kluczy prywatny/publiczny odbiorców wiadomości może odbywać się poza aplikacją, byle klucze publiczne odbiorców były do niej dostarczone po stronie nadawczej, a zaszyfrowane klucze prywatne były dostępne po stronie odbiorczej.

Ad. 4

Wybór metody generowania klucza sesyjnego ma istotne znaczenie dla bezpieczeństwa aplikacji. Najczęstszym atakiem na produkty kryptograficzne jest poszukiwanie słabości generatorów kluczy. Należy posłużyć się generatorem ciągów pseudolosowych dostępnym w obrębie używanego systemu operacyjnego, natomiast wartością wejściową generatora powinna być losowa liczba pobrana z otoczenia (bądź połączenie wielu takich wartości), np. aktualna wartość zegara systemowego.

Ad. 5

Implementacja aplikacji powinna być zrealizowana w środowisku najbardziej przyjaznym dla studenta. Prowadzący zajęcia nie narzuca wyboru narzędzi do realizacji projektu, aczkolwiek należy tu uwzględnić sposób implementacji dostępnej wersji algorytmów szyfrowania. Ponadto student powinien przeprowadzić testy aplikacji przed jej przekazaniem do oceny.

Ad. 6

Krótki raport powinien zawierać informacje o temacie zadania, kształcie interfejsu użytkownika, strukturze pliku wynikowego oraz wynikach testów.

Zasady oceniania.

Zadanie zostanie ocenione wg następujących zasad:
- dokumentacja – do 5 pkt.
- funkcjonalność interfejsu użytkownika – do 5 pkt.
- poprawna realizacja tematu w terminie – do 24 pkt.

I termin kontrolny – przedstawienie projektu interfejsu użytkownika – 3pkt.
II termin kontrolny – przedstawienie częściowo działającej aplikacji – 3 pkt.

I termin zaliczenia projektu – 22-24 maja 2017 w godzinach i salach zgodnych z planem zajęć dla każdej grupy oraz w godzinach konsultacji.
.
Za każdy tydzień opóźnienia od tej daty (22-24.05.2017) **końcowa ocena** będzie pomniejszana o 5pkt.
Ostateczny termin zaliczeń – 20 czerwca 2017. Po tym terminie projekty nie będą oceniane z wyjątkiem usprawiedliwionych sytuacji szczególnych (zwolnienia lekarskie, przypadki losowe).