Motorola Science Cup²
Zadania II Etapu

Kryptografia

OMÓWIENIE

Celem tego zadania jest napisanie programu do zabezpieczenia informacji poprzez szyfrowanie i umożliwienie odczytu poprzez odszyfrowanie wcześniej zabezpieczonego tekstu.
Ponadto aplikacja powinna przedstawić charakterystykę użytego algorytmu szyfrowania i porównać go z innymi.

Aplikacja może zostać wykonana w dowolnym języku programowania, ale musi uruchomić się na systemie Windows 10. Ponadto prosimy o przesłanie wszystkich plików źródłowych. Źródła możecie udostępnić jako archiwum załączone do maila, umieścić w serwisie chmurowym (Dysk Google, OneDrive, etc) lub w systemie kontroli wersji (GitHub, BitBucket, etc).

Prosimy też aby nadsyłać odpowiedzi wraz z dokładną instrukcją opisującą proces budowania gotowej do uruchomienia aplikacji. Prace, które będą dostarczone bez kodu źródłowego i nie odpalą się przy użyciu dostarczonego installera (i/lub pliku wykonywalnego) nie będą oceniane. Aplikacja będzie uruchamiana na komputerze pozbawionym dostępu do internetu, bez zainstalowanych narzędzi deweloperskich, dostępna przeglądarka: Chrome. Jeśli do uruchomienia gry będą potrzebne jakieś dodatkowe narzędzia lub konfiguracja środowiska, aplikacja w ramach instalacji powinna to zapewnić.

CO OCENIAMY

Oceniać będziemy:

1. Zgodność rozwiązania ze specyfikacją (zobacz opisy poszczególnych etapów zadania)
2. Zaimplementowaną architekturę aplikacji, czystość kodu
3. Dokumentację projektu
4. Zaproponowany interfejs graficzny aplikacji. Jego wygląd, czytelność i ogólne wrażenie z użytkowania aplikacji
5. Strategia testów dla każdego z etapów: testy automatyczne (w tym jednostkowe) i jeżeli konieczne - zdefiniowane testy (kampanie) manualne

ETAPY ZADANIA

Etap I - Wersja podstawowa

Za ten etap zadania uzyskacie do **45** punktów.

W trakcie implementacji nie można używać bibliotek kryptograficznych. Dopuszczone użycie wszelkich bibliotek matematycznych, generatorów kluczy itp. Implementacja algorytmów szyfrowania i deszyfrowania musi zostać wykonana w całości przez drużynę.

1. Implementacja szyfru Cezara
    - Aplikacja powinna pozwolić na wybranie pliku tekstowego z dysku oraz zapisać wynik szyfrowania/deszyfrowania w pliku obok.
    - Po załadowaniu pliku aplikacja powinna pozwolić na wybór jednej z operacji: szyfrowanie lub deszyfrowanie
    - Tekst jawny powinien zostać zaszyfrowany przy użyciu algorytmu ROT-13.
    - Po wykonaniu operacji aplikacja powinna wyświetlić tekst wejściowy oraz wyjściowy.

2. Analiza statystyczna (częstości)
    - Aplikacja powinna pozwolić na wybranie pliku tekstowego z dysku
    - Plik wejściowy będzie zawierał zaszyfrowany tekst w języku angielskim
    - Dla zadanego tekstu zaszyfrowanego szyfrem Cezara o nieznanym kluczu należy narysować wykres przedstawiający częstość występowania liter
    - Bazując na charakterystyce przesłanego tekstu aplikacja powinna znaleźć klucz deszyfrujący oraz określić jego prawdopodobieństwo. Aby ta analiza miała sens, sprawdzany tekst musi mieć odpowiednią długość.

3. Implementacja szyfru Vernama (One-Time-Pad)
    - Aplikacja powinna pozwolić na wybranie pliku tekstowego z dysku oraz zapisać wynik szyfrowania/deszyfrowania w pliku obok.
    - Po załadowaniu pliku aplikacja powinna pozwolić na wybór jednej z operacji: szyfrowanie lub deszyfrowanie
    - W przypadku szyfrowania aplikacja powinna wygenerować klucz szyfrujący zgodny z wymaganiami szyfru. Klucz szyfrujący powinien zostać zapisany do pliku tekstowego w tej samej lokalizacji co wynik szyfrowania/deszyfrowania
    - W przypadku deszyfrowania aplikacja powinna pozwolić dodatkowo na wybranie pliku tekstowego z dysku z kluczem szyfrującym
    - Wynik operacji szyfrowania/deszyfrowania wraz z kluczem szyfrującym powinien zostać wyświetlony na ekranie

Etap II - Wersja rozszerzona - punkty dostępne po zrealizowaniu założeń poziomu podstawowego

Za ten etap zadania uzyskacie do **30** punktów.

W tym etapie w trakcie implementacji dopuszczone jest użycie bibliotek kryptograficznych.

1. Szyfrowanie
    - Aplikacja powinna pozwolić na wybranie pliku tekstowego z dysku oraz zapisać wynik szyfrowania/deszyfrowania w pliku obok.
    - Po załadowaniu pliku aplikacja powinna pozwolić na wybór jednej z operacji: szyfrowanie lub deszyfrowanie
    - W przypadku szyfrowania aplikacja powinna wygenerować klucz szyfrujący zgodny z wymaganiami szyfru. Klucz szyfrujący powinien zostać zapisany do pliku tekstowego w tej samej lokalizacji co wynik szyfrowania/deszyfrowania
    - W przypadku deszyfrowania aplikacja powinna pozwolić dodatkowo na wybranie pliku tekstowego z dysku z kluczem szyfrującym
    - Aplikacja powinna pozwolić na wybór szyfrów: AES, DES, Triple DES, Twofish, RSA
    - Aplikacja powinna pozwolić na wybór rozmiaru bloku czy klucza/-y (o ile dany algorytm na to pozwala) oraz zaproponować najbezpieczniejszą z opcji.
    - Wynik operacji szyfrowania/deszyfrowania wraz z kluczem szyfrującym powinien zostać wyświetlony na ekranie

2. Analiza
    - Aplikacja powinna pozwolić na wybranie pliku tekstowego z dysku i zapisać wynik szyfrowania/deszyfrowania w pliku obok.
    - Po załadowaniu pliku aplikacja powinna wykonać operację szyfrowania a następnie deszyfrowania dla każdego algorytmu AES, DES, Triple DES, Twofish, RSA.
    - Dla operacji szyfrowania należy ustawić najbezpieczniejszą możliwą konfigurację
    - Aplikacja powinna przedstawić podsumowanie, w ramach którego będzie można prześledzić:
        - Łączny czas potrzebny na wygenerowanie klucza/-y oraz wykonanie operacji szyfrowania i deszyfrowania dla każdego algorytmu
        - Czas potrzebny na wygenerowanie klucza/-y
        - Czas potrzebny na wykonanie operacji szyfrowania
        - Czas potrzebny na wykonanie operacji deszyfrowania
        - Możliwość sortowania listy szyfrów wg wymienionych wyżej kryteriów