DR INŻ. PIOTR SIEWNIAK

PODSTAWY KRYPTOGRAFII

Ćwiczenia #1

Zadanie 1

Napisać program pozwalający na wykonanie operacji haszowania hasła użytkownika portalu internetowego przy użyciu zasobów modułu `crypto`:

a) klasy `Hash` (*wariant I*);

b) klasy `Hmac` (*wariant II*).

Wykorzystać funkcję skrótu *SHA-256* i kodowanie heksadecymalne (*hex coding*).

Zadanie 2

Treść jak w **zadaniu 1** (wraz z wariantami). W implementacji rozwiązania uwzględnić dodatkowo wykorzystanie zdefiniowanego samodzielnie strumienia odczytywalnego (*readable stream*).

Zadanie 3

Napisać program pozwalający na wykonanie operacji haszowania zawartości źródłowego pliku tekstowego zawierającego wrażliwe dane. Zawartość pliku źródłowego jest następująca: pierwsza linia – imię i nazwisko, druga linia – adres domowy, trzecia linia – numer telefonu, czwarta linia – adres mailowy.

Wszystkie wrażliwe dane po wykonanym procesie haszowania powinny zostać zapisane w pliku docelowym.

Wykorzystać zasoby modułu:

a) `crypto` (*wariant I*);

b) `bcrypt` (*wariant II*) oraz „sól” (*salt*).

Zadanie 4

Napisać program pozwalający na sprawdzenie, czy wrażliwa dana (np. numer telefonu) zapisana w pliku tekstowym w postaci zakodowanej (haszowanej) jest zgodna z jej zwykłą postacią tekstową (*plain text*) zadeklarowaną w programie. Wykorzystać zasoby modułu:

c) `crypto` (*wariant I*);

d) `bcrypt` (*wariant II*) oraz „sól” (*salt*)..

Zadanie 5

Napisać program umożliwiający zaszyfrowanie zawartości źródłowego pliku tekstowego o postaci jak w **zadaniu 3**. Zaszyfrowane dane zapisać w pliku docelowym.

W celu weryfikacji poprawności wykonanego szyfrowania należy odczytać zaszyfrowane dane z pliku docelowego i poddać je operacji deszyfrowania. Porównać uzyskane rezultaty – dane ze źródłowego pliku tekstowego z danymi uzyskanymi w procesie deszyfrowania.

Wykorzystać metody:

a) `createCipher()`, `createDecipher()` (*wariant I*);

b) `createCipheriv()`, `createDecipheriv()` (*wariant II*).

Zadanie 6

Treść jak w **zadaniu 5**. W implementacji rozwiązania uwzględnić przetwarzanie strumieniowe obiektów – instancji klas `Cipher` i `Decipher` oraz wykorzystanie obietnic (*promises*).

Zadanie 7

Porównać hasła użytkownika używane w czasie jego autoryzacji (logowania):

- hasło podane w formularzu logowania;
- zaszyfrowane hasło zapisane w bazie danych.

Wykorzystać zasoby modułu `bcrypt` – metodę `compareSync()`).

Zadanie 8

Treść jak w **zadaniu 7**, ale zamiast metody `compareSync()` wykorzystać metodę `compare()`).

Operacje asynchroniczne (włącznie z błędami) obsłużyć za pomocą:

a) funkcji zwrotnych (*wariant I*);

b) promisów (*wariant II*);

c) funkcji `async` (*wariant III*).