Wyższa Szkoła Bankowa w Poznaniu

Wydział Finansów i Bankowości

Studia stacjonarne I stopnia – Informatyka

Dokumentacja projektu zaliczeniowego

Przedmiot: **Bezpieczeństwo w systemach i sieciach komputerowych**

Grupa: **K01**

Rok akad.: **2020/2021**

Prowadzący: mgr inż. Marcin Zajdowski

Temat: **Implementacja algorytmu kryptograficznego AES-256-CBC**

Student: **Jacek Obst**

Data wykonania: **16.05.2021**

AES to jeden z obecnie najbardziej popularnych rodzajów szyfrowania. Wykorzystuje on tak zwany szyfr blokowy. Polega on na tym, że dostarczamy mu treść oraz tekst klucza. Algorytm zwraca treść, ale tak zniekształconą, że bez znajomości klucza praktycznie nie jesteśmy w stanie cofnąć się do oryginału wiadomości.

Projekt został wykonany w formie aplikacji internetowej, napisanej we frameworku języka PHP – Laravelu. Wykorzystuje on architekturę MVC (Model – View – Controller). Laravel dostarcza nam wbudowany już mechanizm szyfrowania, oparty o openssl, którego obsługa jest natomiast zaimplementowana już w czystym PHP. Stosowany algorytm możemy wybrać spośród dwóch następujących:

- AES-128-CBC (ten został przeze mnie wykorzystany w projekcie)

- AES-256-CBC

Klucz szyfrujący zapisany jest w pliku środowiskowym aplikacji (.env) i, w zależności od wybranego algorytmu, przyjmuje, odpowiednio, postać 16-znakowego lub 32-znakowego ciągu znaków zapisanego w formacie Base64.

Na potrzeby projektu utworzone zostały 3 endpointy:

GET / - zwraca widok z dwoma formularzami

POST /encrypt - przyjmuje zwykły plik tekstowy (.txt), szyfruje go i zwraca w formie zaszyfrowanego pliku do pobrania

POST /decrypt - przyjmuje zaszyfrowany plik tekstowy (.txt), odszyfrowywuje go i zwraca w formie zwykłego pliku do pobrania

Dane wysyłane przez formularze są walidowane, a w razie błędów, wyświetlone zostają odpowiednie komunikaty.

Samo szyfrowanie odbywa się dzięki tzw. helperom, czyli funkcjom, które możemy użyć w dowolnym miejscu w projekcie, bez importowania ich.

```
$encryptedText = encrypt($files['file']->getContent());
$decryptedText = decrypt($files['file']->getContent());
```

Źródła:

- https://trybawaryjny.pl/co-oznacza-szyfrowanie-256-aes/