AES128

Interfejs wyjściowy

Jako wyjście dla elementu AES128 wykorzystany został interfejs Intel Avalon MM. Aby wykorzystać komponent AES wymagane jest podanie klucza oraz textu. Kolejnym krokiem jest podanie wektora kontrolnego który określa działanie AES ( szyfrowanie 0000 0011/ deszyforwanie 0000 0010).

przedział adresów CSR 0x0000 0000 - 0x0000 007f

wektor kontrolny 0x0000 0001

klucz 0x0000 0002 - 0x0000 0005

input 0x0000 0006 0x0000 0009

output 0x0000 000A 0x 0000

Dostępny jest też skrypt ( console.tcl) pozwalający przyspieszyć testowanie algorytmu. Jako wejście przyjmuje plik input.txt, który powinien być wypełniony liniami znaków o długości 128bitów. Wynik szyfrowania jest wyświetlony w konsoli systemowej.

SubBytes():

Jest to nieliniowe zastąpienie bajtów, które działa indywidualnie dla każdego bajtu, korzystając z tabeli podstawienia (S box). S box jest skonstuowany poprzez wykorzystanie odwrotności mnożenia w ciele Galois GF2^8 z nieskracalnym wielomianem wielomian m(x) = x 8 + x 4 + x 3 + x +1. Transformacja jest wykonana dla każdego elementu GF2^8. Element {00} jestodwzorowany w siebie.



ShiftRow()

Element ten cyklicznie przesuwa rzędy danych o różne wartości przesunięcia. Operacja ta jest niemalże identyczna w procesie deszyfrowania ( różni się jedynie wartościami przesunięć).

MixColumns()

Transformacja ta działa na kolumny danych wejściowych. Każda kolumna jest traktowana jako wielomian stopnia czwartego w ciele GF2^8. Następnie każdy wielomian jest mnożony przez wielomian a(x) = {03} x3 + {01} x 2 + {02} x w modulo x 4 + 1.

AddRoundKey()

W tym kroku wykonywany jest XOR każdego bajtu klucza z odpowiadającym mu bajtem danych weściowych.

