|  |  |  |  |
| --- | --- | --- | --- |
| POLITECHNIKA LUBELSKA  Wydział Elektrotechniki i Informatyki  Informatyka | **Bezpieczeństwo Informacji**  **Prowadzący: dr Waldemar Suszyński** | | |
| Nazwisko i Imię | Semestr  IV | Rok akademicki  2023/2024 | Grupa:  IIST 4.5 |
| 1. |
| Data wykonania ćwiczenia: | GL 9 |
| Temat ćwiczenia:  KRYPTOGRAFIA SYMETRYCZNA, TRYBY PRACY ALGORYTMÓW KRYPTOGRAFICZNYCH | | Data oddania sprawozdania: | OCENA: |

1. **Zadanie 4.1** **Szyfrowanie i deszyfrowanie przy użyciu wybranych algorytmów szyfrów symetrycznych blokowych oraz trybów pracy algorytmów szyfrujących.**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie**

- W zadanym trybie wektor inicjalizujący nie jest potrzebny.  
- Aby uniknąć komunikatu „**hex string is too short, padding with zero bytes to length”** trzeba zwiększyć odległość klucza.

**- a**es-128 potrzebuje 128-bitowy klucz (czyli 16 bajtów), a polecenie zadaje tylko 8 bajtów.

**-** Nie. Na przykład **aes-ecb** nie wymaga.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

- Szyfrowanie i deszyfrowanie przy zastosowaniu -base64 oraz -salt nie daje tych samych rezultatów.

1. **Zadanie 4.2. Porównanie trybu szyfrowania EBC z CBC na przykładzie szyfrowania plików graficznych.**

**Obraz zawierający trawa, na wolnym powietrzu, niebo, wiosna

Opis wygenerowany automatycznie**Obraz zawierający zrzut ekranu, Prostokąt, szary

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

1. **Zadanie 4.3. Powstawanie błędów w szyfrogramie.**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**BŁĄD: 30 bit**

1. **Zadanie 4.4. Analiza dopełniania bloków danych podczas szyfrowania algorytmami Symetrycznymi.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tryb szyfrowania** | **Rozmiar pliku** | **Polecenie szyfrowania** | **Czy wymaga uzupełniania bloków** |
| aes-128-ecb | 20 bajt | openssl enc -e -aes-128-ecb -in 1.txt -out 1.e.txt -K 0123456789abcdef | Tak |
| aes-128-ecb | 32 bajty | openssl enc -e -aes-128-ecb -in 2.txt -out 2.e.txt -K 0123456789abcdef | Tak (rozmiar po szyfrowaniu 48 bajt) |
| aes-128-cbc | 20 bajt | openssl enc -e -aes-128-cbc -in 1.txt -out 1.e.txt -K 0123456789abcdef -iv 0123456789abcdef | Tak |
| aes-128-cbc | 32 bajty | openssl enc -e -aes-128-cbc -in 2.txt -out 2.e.txt -K 0123456789abcdef -iv 0123456789abcdef | Tak (rozmiar po szyfrowaniu 48 bajt) |
| aes-128-cfb | 20 bajt | openssl enc -e -aes-128-cfb -in 1.txt -out 1.e.txt -K 0123456789abcdef -iv 0123456789abcdef | Nie |
| aes-128-cfb | 32 bajty | openssl enc -e -aes-128-cfb -in 1.txt -out 1.e.txt -K 0123456789abcdef -iv 0123456789abcdef | Nie |
| aes-128-ofb | 20 bajt | openssl enc -e -aes-128-cfb -in 1.txt -out 1.e.txt -K 0123456789abcdef -iv 0123456789abcdef | Nie |
| aes-128-ofb | 32 bajty | openssl enc -e -aes-128-ofb -in 2.txt -out 2.e.txt -K 0123456789abcdef -iv 0123456789abcdef | Nie |
| aes-128-ctr | 20 bajt | openssl enc -e -aes-128-ctr -in 1.txt -out 1.e.txt -K 0123456789abcdef -iv 0123456789abcdef | Nie |
| aes-128-ctr | 32 bajty | openssl enc -e -aes-128-ctr -in 2.txt -out 2.e.txt -K 0123456789abcdef -iv 0123456789abcdef | Nie |

1. **Zadanie 4.5 Określenie entropii jądra systemowego.**

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie  
Wykonanie zadanych poleceń na komputerze domowym poskutkowało zaskakująco. Podczas realizowania zadania podczas laboratoriów zwracana była różna entropia. W domu za każdym razem niezależnie od podjętych starań entropia była stała.

1. **Zadanie 4.6. Pobieranie liczb pseudolosowych z /dev/random i /dev/urandom.**

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie** **Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**