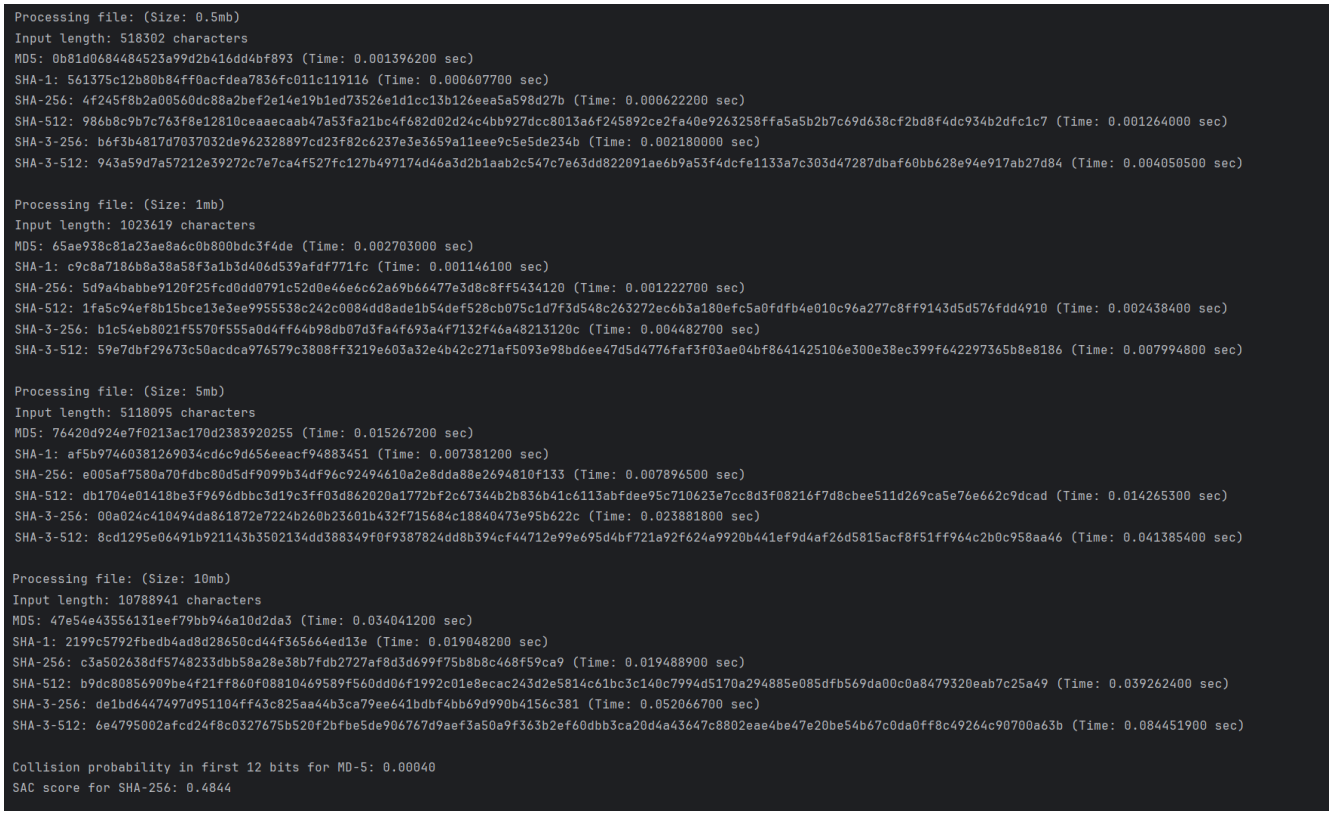
Podstawy kryptografii – funkcje skrótu

Screenshot z aplikacji:



Sposób implementacji

Aplikacja została stworzona w Pythonie z wykorzystaniem modułu „hashlib”, który obsługuje algorytmy skrótu takie jak MD5, SHA-1, SHA-256, SHA-512, SHA-3-256 i SHA-3-512. Program porównuje czasy wykonania tych funkcji dla różnych długości tekstów, analizując ich wydajność.

Również przeprowadzono badanie kolizji w MD-5 dla pierwszych 12 bitów oraz ocenę losowości wyjścia funkcji SHA-256 za pomocą testu SAC, który sprawdza, jak zmiana jednego bitu w wejściu wpływa na wynik.

W wyniku badania MD-5 uzyskano wynik:
“Collision probability in first 12 bits for MD-5: 0.00040”, co oznacza, że algorytm MD-5 co wskazuje, że algorytm nie jest zbyt bezpieczny, ponieważ występują kolizje.

natomiast wynik testu SAC dla SHA-256 wynosi:
"SAC score for SHA-256: 0.4844”, co jest bliskie 0.5, co oznacza bardzo bliski 0.5 rozkład zmian, potwierdzający wysoką losowość i bezpieczeństwo tej funkcji skrótu.

Rola soli w tworzeniu skrótów

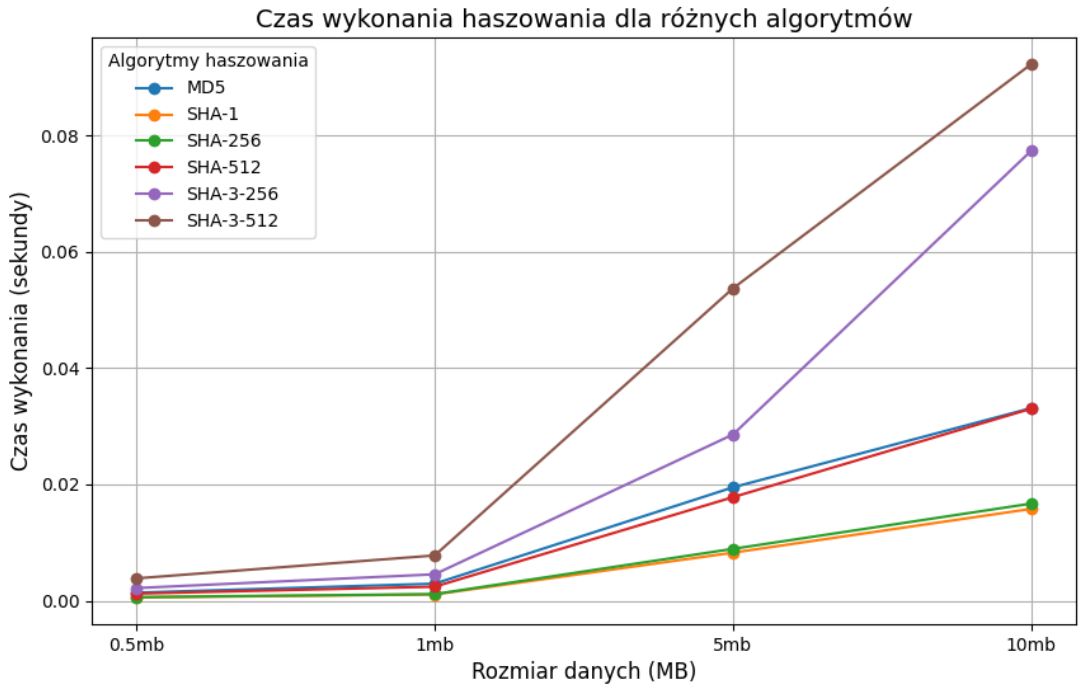
Sól to losowy ciąg znaków, który dodaje się do danych przed obliczeniem funkcji skrótu, w celu zwiększenia bezpieczeństwa. Dzięki soli nawet identyczne dane wejściowe będą miały różne skróty, co zapobiega ich łatwej

identyfikacji. Sól utrudnia ataki, takie jak używanie tabel tęczowych, które polegają na wstępnie obliczonych skrótach, oraz ataki słownikowe, w których używa się popularnych haseł. W skrócie, sól zwiększa bezpieczeństwo funkcji skrótu, zapewniając większą losowość i odporność na ataki.

Czy funkcja MD5 jest bezpieczna

Funkcja MD5 nie jest uważana za bezpieczną, ponieważ zostały udowodnione kolizje, co oznacza, że różne dane mogą generować ten sam skrót. Z tego powodu MD5 nie jest odpowiednia do zastosowań wymagających wysokiego poziomu bezpieczeństwa. Zamiast niej zaleca się używanie algorytmów takich jak SHA-256, SHA-3 i innych.

Zestawienie uzyskanych wyników wraz ze stosownymi wnioskami



Z wykresu wynika, że im dłuższy tekst, tym więcej czasu zajmuje obliczenie funkcji skrótu. Najszybsze algorytmy to SHA-1 i SHA-256, podczas gdy najwolniejszy jest SHA-3-512. MD5, choć dość szybkie, nie jest już uważane za bezpieczne.
Zwiększenie rozmiaru tekstu powoduje, że czas wykonania większości funkcji skrótu rośnie płynnie. Jednak algorytmy SHA-3-512 i SHA-3-256, po zwiększeniu rozmiaru tekstu do 1-5MB, wykazują nagły wzrost czasu wykonania, co oznacza, że czas pracy tych funkcji skrótu znacząco się wydłuża.
Można wywnioskować, że wybór algorytmu to kompromis między szybkością a poziomem bezpieczeństwa.