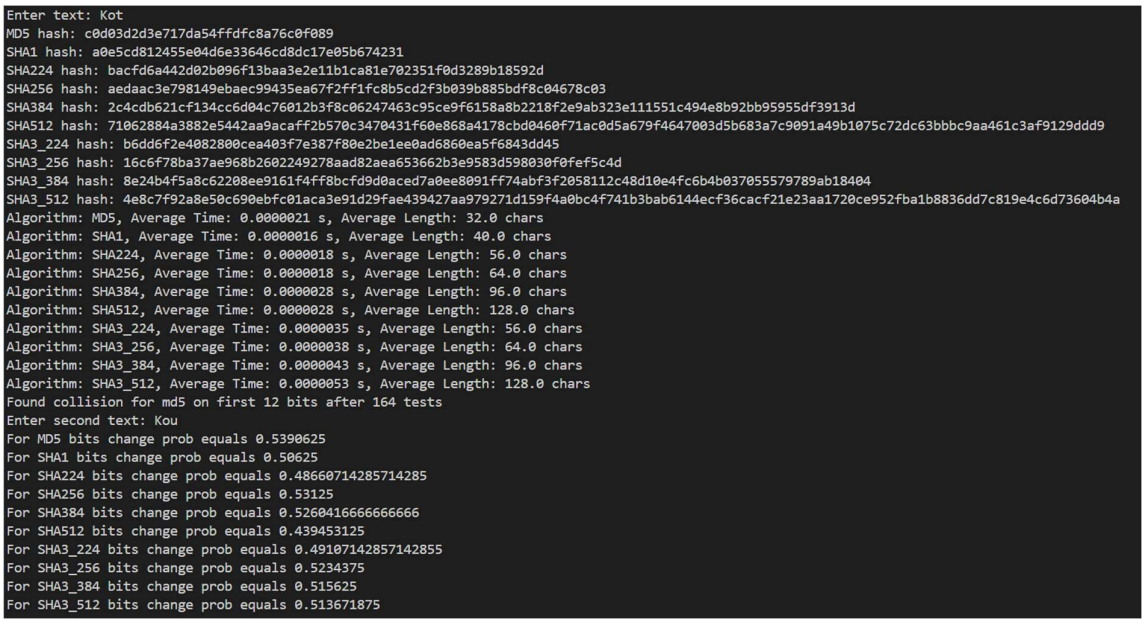
Sprawozdanie – Funkcje skrótu

Mateusz Kreczmer 151736

1. Screenshot z aplikacji.



2. Omówienie sposobu implementacji.

Implementacja tego kodu obejmuje kilka funkcji. Funkcja **generate_hash** przyjmuje tekst i algorytm jako argumenty i zwraca wynik skrótu dla danego tekstu za pomocą wybranego algorytmu. Do hashowania tekstów odpowiednimi algorytmami użyłem biblioteki **hashlib**.

Funkcja **calculate_average_speed_and_length** oblicza średni czas wygenerowania skrótu i średnią długość skrótu dla zestawu danych tekstowych.

Funkcja **collision_on_first_12_bits** szuka kolizji na pierwszych 12 bitach wyniku skrótu dla wybranego algorytmu. Badam występowanie kolizji poprzez generowanie losowych 8-znakowych tekstów, aż do momentu wygenerowania takiego, który koliduje z wcześniej już wygenerowanym.

Funkcja **check_bit_flip_probability** oblicza prawdopodobieństwo zmiany bitu między dwoma tekstami dla danego algorytmu skrótu.

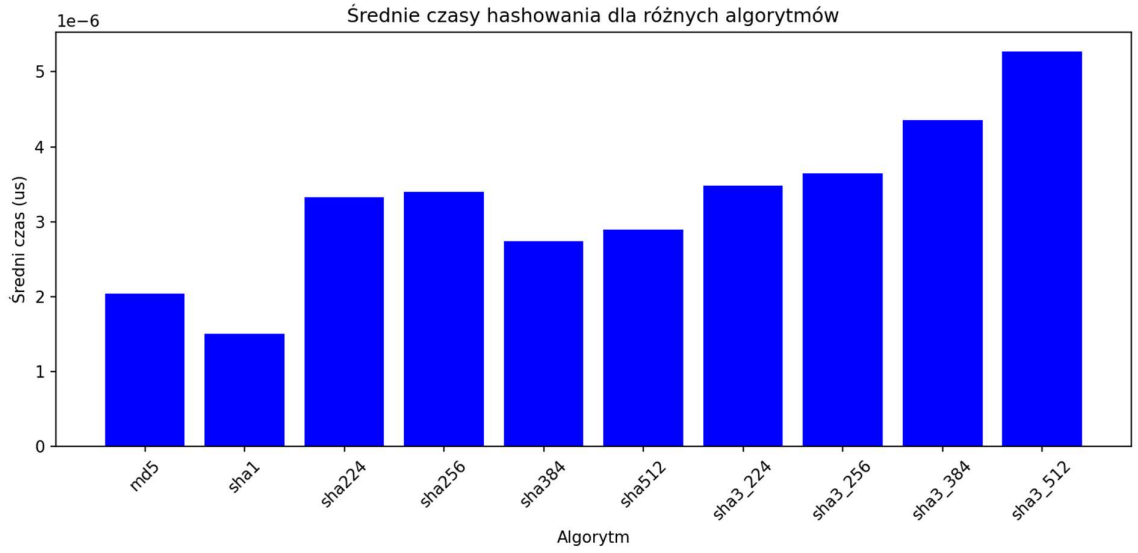
3. Określenie roli soli w tworzeniu skrótów.

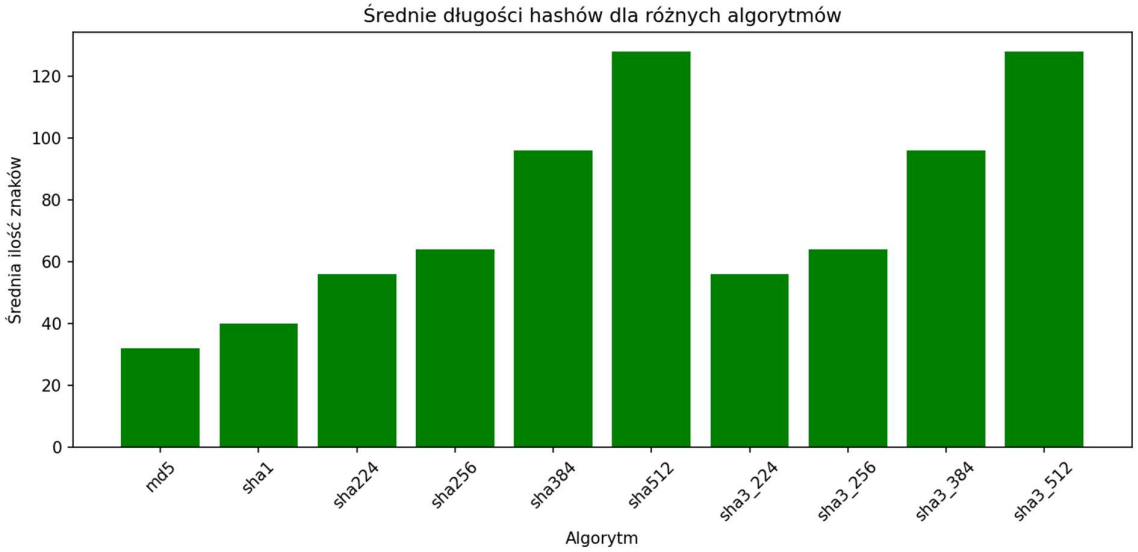
Rola soli w tworzeniu skrótów polega na dodaniu losowego ciągu danych (soli) do tekstu przed obliczeniem skrótu. Sól zapobiega atakom typu "rainbow table", gdzie wcześniej obliczone skróty są przechowywane w tabelach dla szybkiego odwzorowania skrótów na ich pierwotne wartości. Dodanie soli powoduje, że każde wygenerowane hasło ma unikalny skrót, nawet jeśli są to te same hasła.

4. Na podstawie powszechnie dostępnych źródeł odpowiedz na pytanie – czy funkcję MD5 można uznać za bezpieczną? Czy dotychczas zostały znalezione dla niej jakiekolwiek kolizje?

Na podstawie powszechnie dostępnych źródeł, funkcję MD5 nie można uznać za bezpieczną. Dotychczas znaleziono szereg ataków kryptograficznych, które wykazują jej słabość. Przykładowo, ataki takie jak "collision attack" umożliwiają wygenerowanie dwóch różnych wiadomości, które mają taki sam skrót MD5. W przypadku zastosowań wymagających wysokiego poziomu bezpieczeństwa, zaleca się wybieranie bardziej bezpiecznych algorytmów skrótu, takich jak SHA-2 lub SHA-3 ponad MD5.

5. Zestawienie uzyskanych wyników wraz ze stosownymi wnioskami.





Na podstawie uzyskanych wyników można zauważyć, że różne algorytmy skrótu mają różne czasy wykonania oraz generują skróty o różnych długościach. Na przykład, algorytm MD5 wykazuje krótsze czasy wykonania niż algorytmy SHA-3, ale generuje skróty o stałej długości 32 znaków. Z drugiej strony, algorytmy SHA-3 mają dłuższe czasy wykonania, ale generują skróty o zmiennych długościach, w zależności od wariantu. Wniosek wynika stąd, że wybór algorytmu skrótu powinien być dokładnie przemyślany, z uwzględnieniem równowagi między czasem wykonania a bezpieczeństwem skrótu oraz wymaganą długością skrótu. Na przykład, w przypadku zastosowań, gdzie ważne są krótkie czasy obliczeń, algorytm MD5 może być wystarczający, podczas gdy w bardziej wymagających sytuacjach zalecane jest stosowanie bardziej bezpiecznych algorytmów, takich jak SHA-3, kosztem wydłużenia czasu obliczeń.