Funkcje skrótu

Sebastian Nowak - 152065

1. Screenshot wyników

2. Sposób implementacji

Aplikacja została napisana w pythonie. Do wszystkich funkcji skrótu została wykorzystana biblioteka *hashlib.* Na screenie powyżej widać porównianie czasów różnych funkcji skrótu oraz badanie kolizji i losowości hash'u funkcji SHA-256.

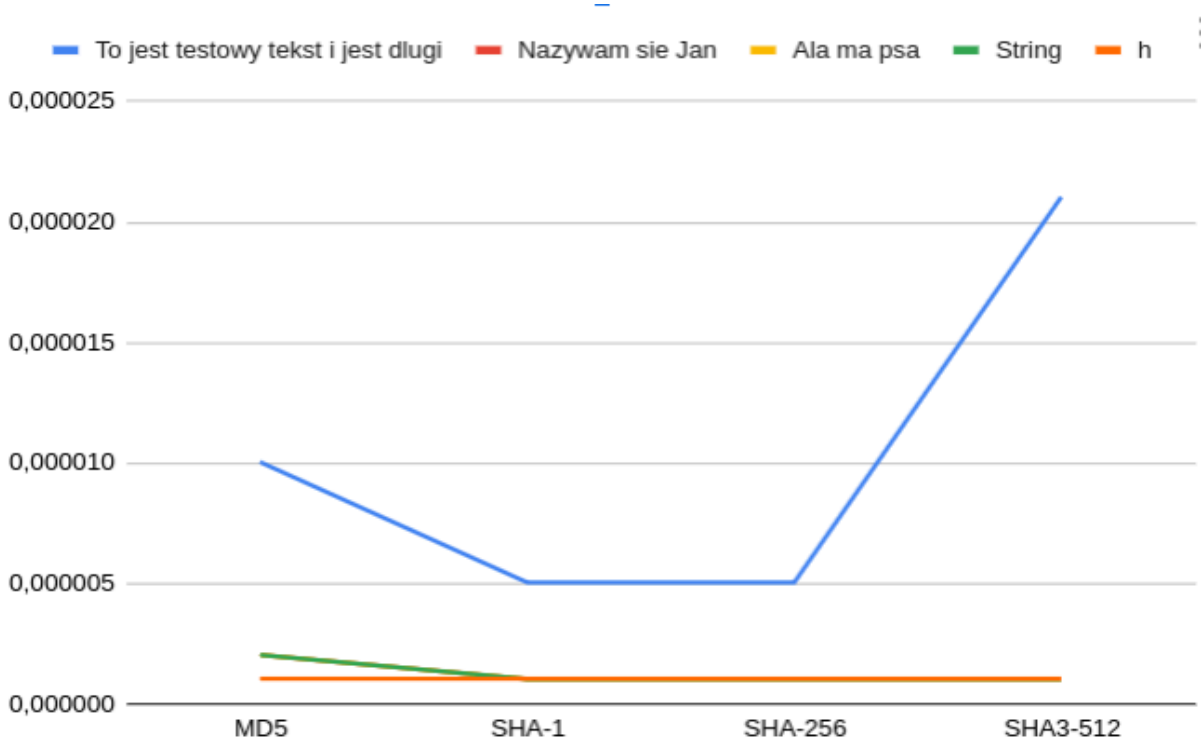
3. Rola soli w powstawaniu skrótów

Sól, dodatkowy ciąg danych dodawany do tekstu przed wyznaczeniem skrótu, powoduje, że nawet dwa identyczne hasła generują różne skróty. Sól zwiększa zabezpieczenia przed atakami typu „rainbow table", które korzystają z bazy skrótów, co pozwala zaoszczędzić moc obliczeniową potrzebną do złamania hasła metodą „brute force". W mojej implementacji nie uwzględniono soli, co mogłoby prowadzić do potencjalnych luk w bezpieczeństwie.

4. Bezpieczność funkcji MD5

Znalezione kolizje oraz wykazana podatność na ataki sprawiają, że funkcja MD5 jest uznawana za przestarzałą i niebezpieczną. Nie zaleca się stosowania MD5 w sytuacjach, w których bezpieczeństwo danych jest kluczowe.

5. Wnioski



- MD5 dłużej się wykonuje niż pozostałe funkcje, co jest szczególnie widoczne przy dłuższych ciągach wejściowych.
- Generowanie skrótów dla długich ciągów zajmuje więcej czasu niż dla krótkich.
- Prawdopodobieństwo wystąpienia kolizji na pierwszych 12 bitach skrótu wyznaczonego przy użyciu SHA-256 wynosi około 4,1%.
- Losowość wyjścia funkcji skrótu SHA-256 wynosi 51,56%, co oznacza, że jest zbliżona do 50%.