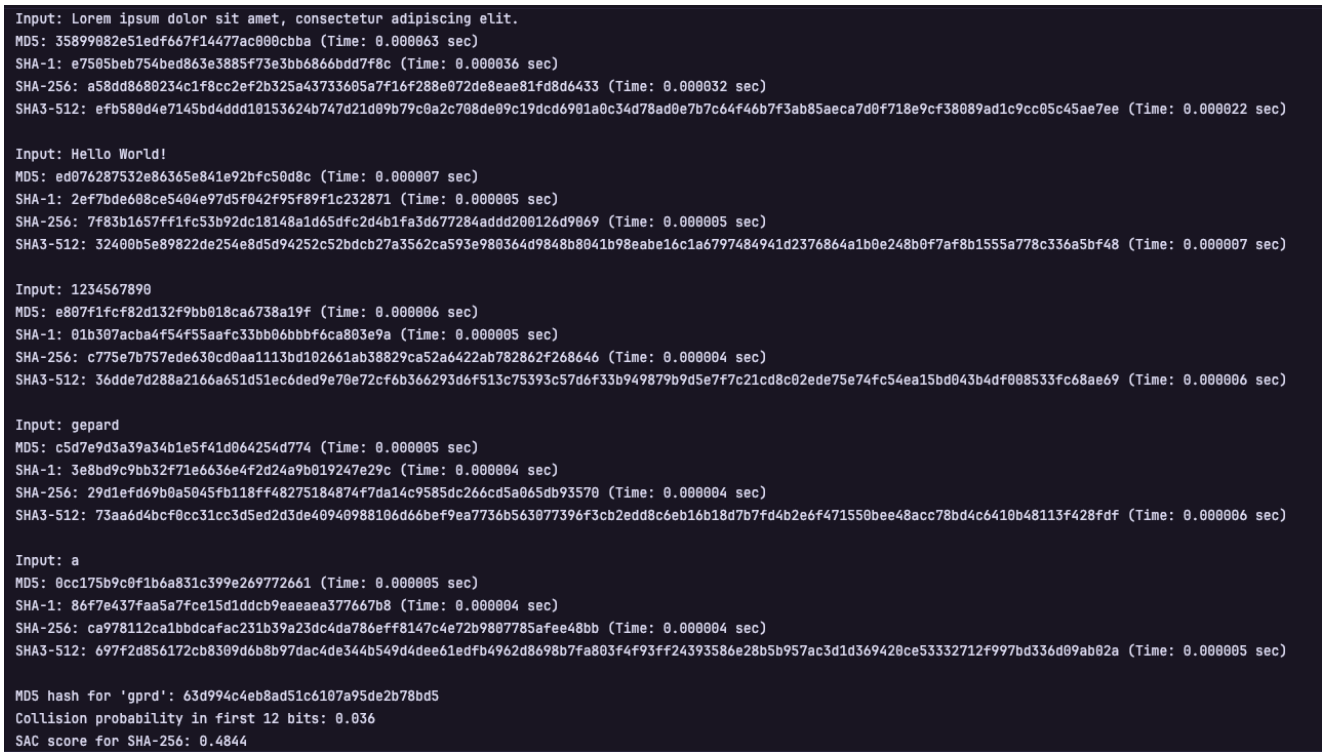
Katarzyna Jabłońska 151761

Podstawy kryptografii – funkcje skrótu

1. Screenshot z aplikacji:



2. Sposób implementacji

Aplikacja została zaimplementowana w języku Python, przy użyciu modułu „hashlib”, który zapewnia dostęp do różnych algorytmów skrótu. W mojej implementacji porównuję czasy działania następujących algorytmów: MD5, SHA-1, SHA-256 oraz SHA3-512. Badam również szansę wystąpienia kolizji na pierwszych 12 bitach oraz losowość wyjścia funkcji skrótu SHA-256.

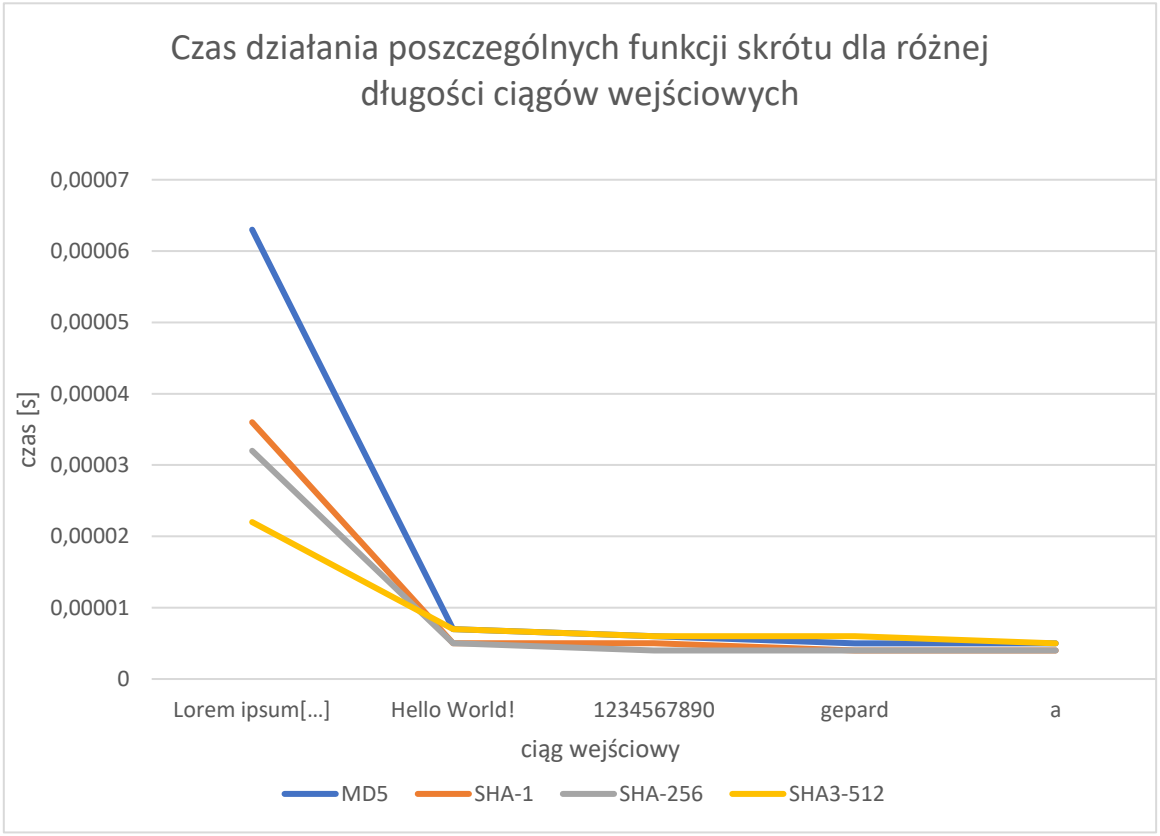
3. Rola soli w tworzeniu skrótów

Sól jest dodatkowym ciągiem danych dodawanym do tekstu przed wyznaczeniem skrótu, co powoduje, że nawet 2 identyczne hasła generują inne skróty. Sól zwiększa zabezpiecza przed atakami typu „rainbow table” korzystających z bazy skrótów, która pozwala na zaoszczędzenie mocy obliczeniowej koniecznej do złamania hasła metodą „brute force” W mojej implementacji nie została ona uwzględniona, co mogłoby prowadzić do potencjalnych luk w bezpieczeństwie.

4. Czy funkcję MD5 można uznać za bezpieczną

Funkcja MD5 jest uznawana za przestarzałą i niebezpieczną ze względu na znalezione kolizje oraz wykazaną podatność na ataki. Nie zaleca się stosowania MD5 w sytuacjach, w których bezpieczeństwo danych jest istotne.

5. Zestawienie wyników



Wnioski:

- Czas działania MD5 jest dłuższy od pozostałych funkcji, co najlepiej widać przy ciągach dłuższych ciągach wejściowych
- Generowanie skrótów dla długich ciągów zajmuje więcej czasu niż dla krótkich ciągów.
- Prawdopodobieństwo wystąpienia kolizji na pierwszych 12 bitach skrótu wyznaczonego przy użyciu SHA-256 wynosi ok. 3,6%
- Losowość wyjścia funkcji skrótu SHA-256 wynosi 48,44%, czyli jest zbliżona do idealnego wyniku, którym byłoby 50%