SPRAWOZDANIE
KRYPTOGRAFIA I BEZPIECZEŃSTWO SYSTEMÓW INFORMATYCZNYCH

“Funkcje haszujące i ich zastosowanie w zapewnianiu integralności danych”

Marek Pięta
Automatyka i Robotyka, ISS

Funkcja haszująca (funkcja skrótu) - jest to funkcja, która przyporządkowuje podanej na wejście liczbie (wiadomości) unikalny skrót (krótką wartość liczbową/wiadomość, zazwyczaj o określonej długości - "hash value"). Dane wejściowe mogą mieć praktycznie dowolną postać (plik o dowolnym rozszerzeniu, ciąg znaków itp.).

Z uwagi na ograniczoną długość skrótu muszą istnieć zbiory danych, dla których generowany skrót będzie taki sam (wystąpienie wielu danych o takiej samej wartości funkcji skrótu określane jest mianem kolizji). Wyeliminowanie tego zjawiska przy ograniczonej długość generowanych hash'ów nie jest możliwe (zazwyczaj hash jest krótszy niż dane wejściowe). Z uwagi na złożoność obliczeniową praktycznie nie powinno jednak być możliwe wygenerowanie dwóch dowolnych wiadomości o takim samym skrócie ani wiadomości o takim samym skrócie jak podana (odporność na kolizje). W przypadku podobnych do siebie wiadomości, generowane skróty powinny znacząco się różnić. Na podstawie znanej funkcji skrótu nie można określić danych podanych na wejście funkcji hashującej (funkcja ta jest jednokierunkowa).

Istnieje wiele powszechnie używanych funkcji skrótu. Różnią się one od siebie m.in. długością uzyskiwanego hash'a, złożonością obliczeniową, oraz odpornością na ataki. W wielu znanych funkcjach skrótu zostały odnalezione istotne słabości (np. **MD4**, **MD5**, **SHA-0**), co powinno skutkować zastąpieniem tych funkcji przez inne (np. należące do rodziny **SHA-2**).

Integralność danych - cecha danych uniemożliwiająca wprowadzenie do nich zmian w sposób nieautoryzowany. Do modyfikacji danych może dojść w sposób przypadkowy (podczas wykonywania pewnej operacji na danych - np. ich transmisji) lub celowy (modyfikacja przez osoby nieupoważnione np. w celu osiągnięcia określonych korzyści).
Zapewnienie integralności danych jest bardzo istotnym problemem w wielu dziedzinach.

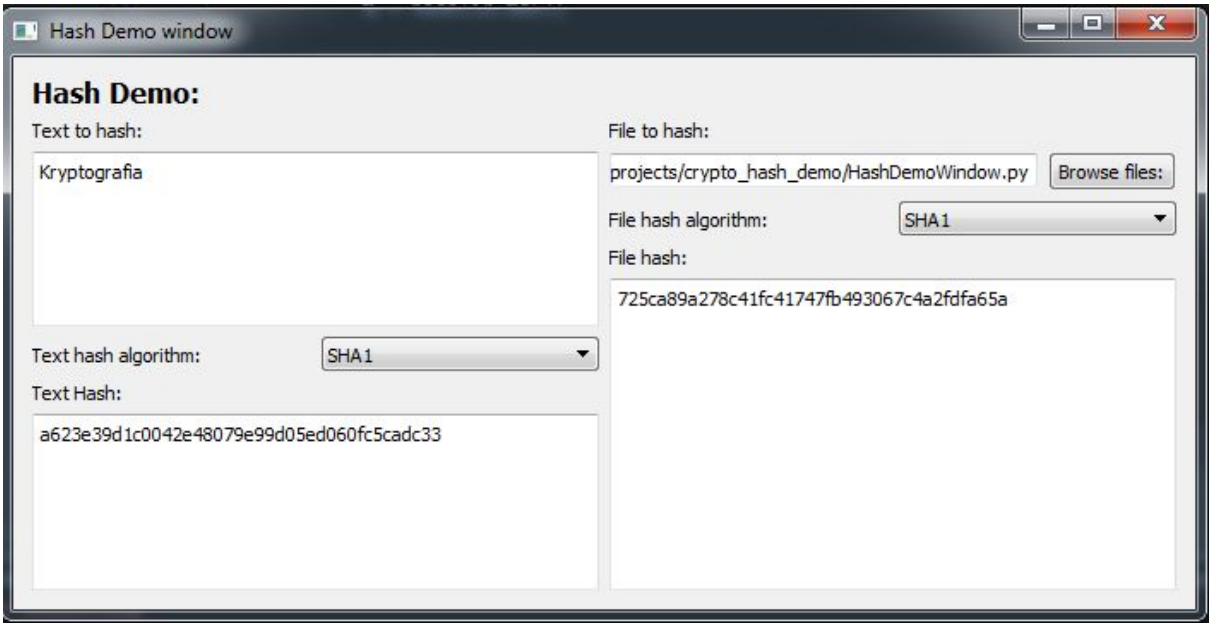
Bazując na obliczonej funkcji skrótu można stwierdzić, że plik nie został uszkodzony ani zmodyfikowany w sposób nieautoryzowany bez konieczności porównywania całej zawartości pliku (w przypadku modyfikacji pliku zmieni się jego hash; dokonanie celowej modyfikacji, tak aby hash nie uległ zmianie jest praktycznie niemożliwe). Nawet w przypadku niewielkich modyfikacji danych wejściowych uzyskiwane funkcje skrótu będą znacząco różne od siebie. Ponadto na podstawie znanego jedynie hash'u nie ma możliwości odtworzenia całej zawartości pliku.

Przykłady zastosowań funkcji skrótu:

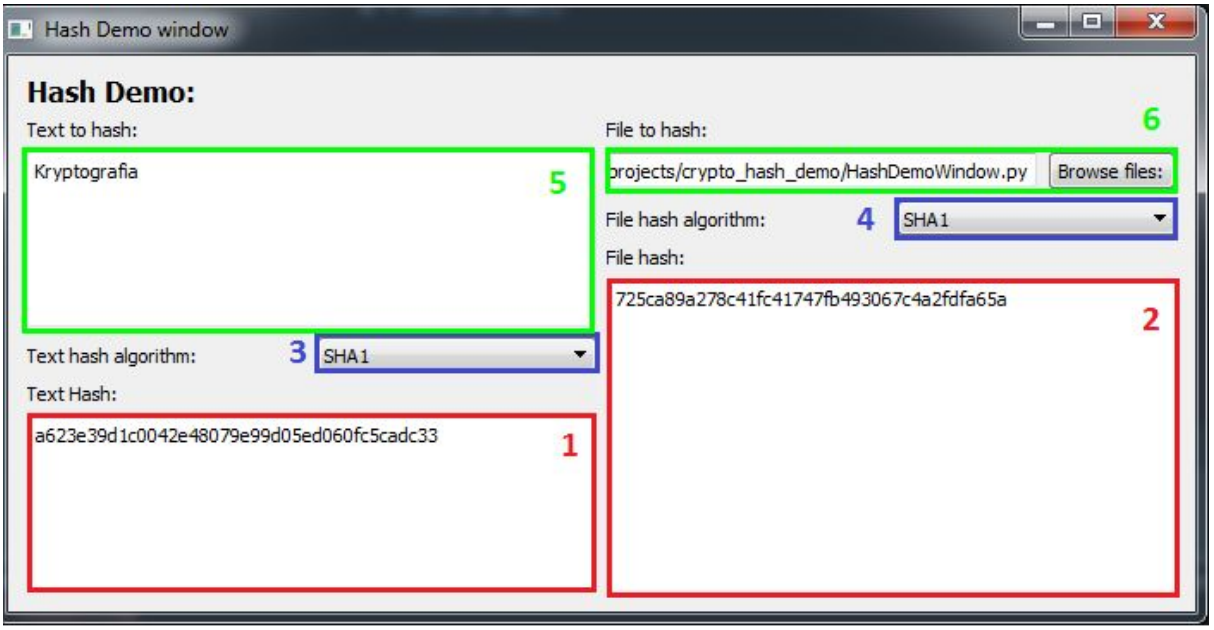
- Przechowywanie haseł użytkowników w postaci jedynie ich skrótu (zamiast w postaci jawnej) praktycznie uniemożliwia odzyskanie jawnej postaci hasła i jednocześnie umożliwia autoryzację użytkownika (zamiast haseł w postaci jawnej porównywane są wartości odpowiednich hash'ów). Pozwala to na zwiększenie bezpieczeństwa użytkowników.
- Określenie, czy przesyłane/pobrane dane nie zostały uszkodzone lub zmodyfikowane. W przypadku różnic w danych, obliczona wartość funkcji skrótu będzie inna niż dla poprawnych danych, co umożliwia wykrycie błędu.
- Tablica mieszające (z ang. "hash table") - wartość funkcji skrótu wyliczanej na podstawie określonego klucza pozwala na szybszy dostęp do odpowiadającego mu elementu w tablicy (cały hash lub jego fragment służy do określania indeksu w tablicy przechowującej określone dane). W przypadku wystąpienia kolizji indeksów stosowane są dodatkowe metody pozwalające na jednoznaczne określenie, do których danych chcemy uzyskać dostęp (metoda rozróżniania kluczy o tym samym hash'u może opierać się o porównywanie całych kluczy - są one jednak porównywane w całości dla znacznie mniejszego zbioru danych).
- Podpis cyfrowy - w celu autoryzacji wiadomości często wykorzystuje się algorytmy szyfrujące i deszyfrujące. Z uwagi na dość długi czas wykonywania tych operacji dla dużych danych często dokonuje się szyfrowania jedynie funkcji skrótu wyznaczonej dla danego zbioru danych (jest to znacznie szybsze rozwiązanie).
- Systemy kontroli wersji również opierają się na funkcji skrótu przy identyfikacji wersji plików. Dla przykładu GIT korzysta z algorytmu SHA-1 przy identyfikacji poszczególnych commitów oraz przy ich weryfikacji (np. przy pull'owaniu zawartości repozytorium).

Aplikacja:

W ramach pracy została napisana aplikacja desktopowa w języku Python z wykorzystaniem biblioteki PyQt (załączona w postaci pliku wykonywalnego "hash_demo.exe"). Celem aplikacji jest zademonstrowanie działania różnych algorytmów do generowania funkcji skrótu. Może ona być wyznaczana na podstawie wpisanego przez użytkownika ciągu znaków lub wybranego pliku z komputera.



Rys. 1: Okno główne aplikacji "Hash Demo window".



Rys. 2: Okno główne aplikacji "Hash Demo window" z oznaczonymi głównymi komponentami.

Po uruchomieniu aplikacji wyświetla się okno główne. W lewej części okna jest pole tekstowe umożliwiające podanie ciągu znaków, na podstawie którego wyznaczony zostanie hash [5] (wartość funkcji skrótu podawana jest w polu umieszczonym poniżej [1]). Rozwijane menu pozwala na określenie funkcji hashującej dla tekstu [3]. Wyświetlana wartość funkcji skrótu aktualizowana jest na bieżąco przy każdorazowej modyfikacji pola z tekstem traktowanym jako dane wejściowe oraz przy zmianie algorytmu hashowania.

Prawa część okna pozwala na obliczanie funkcji skrótu dla dowolnego pliku. Po wciśnięciu przycisku “Browse files:” [6] pojawia się okienko umożliwiające wybór pliku, dla którego wyliczana jest funkcja skrótu (po dokonaniu wyboru ścieżka do pliku pojawia się w polu tekstowym obok). Analogicznie jak w przypadku ciągu znaków istnieje możliwość wyboru algorytmu [4]. Obliczany hash [2] aktualizowany jest na bieżąco (analogicznie jak w przypadku ciągu znaków). W przypadku, gdy plik nie jest wybrany (w odpowiednim polu nie wyświetla się ścieżka dostępu) hash nie jest obliczany. Przy dużych rozmiarach pliku operacja obliczania wartości funkcji hashującej może chwilę potrwać.