Matematyka dla Ciekawych Świata

Kryptograficzne Funkcje Skrótu

ICM UW

Maciej Skórski
Uniwersytet Warszawski
maciej.skorski@mimuw.edu.pl

Opis kursu: Celem zajęć jest przekazanie uczestnikom podstawowej wiedzy na temat kryptograficznych funkcji skrótu (eng. *cryptographic hash functions*) i ich praktycznych zastosowań w charakterze "odcisku palca" takich jak: (a) bezpieczne przechowywanie haseł (np. listy haseł użytkowników na serwerze), (b) weryfikacja poprawności i autentyczności plików (np. plików instalacyjnych czy obrazów ISO pobieranych przez internet), czy (c) tzw. dowodu wykonanej pracy (używany do ochrony przed spamem i w kryptowalucie BitCoin).
Wymagania: Przydatne będzie intucyjne pojęcie o prawdopodobieństwie. Do ćwiczeń praktycznych wymagana jest znajomość podstaw programowania (materiały pomocnicze zawierają skrypty w języku Python.
Uwagi: Brak.
Literatura:

- *Kryptograficzne funkcje skrótu*, P.Rodwald. Zeszyty Naukowe Akademii Marynarki Wojennej, nr 2 (193) ISSN 0860-889X, pp. 91-102 (dostępne online)
- *Introduction to Cryptography with Maple*, José Luis Goómez Pardo. Springer, ISBN 3642321658. (w języku angielskim)

Cel kursu:
Uczestnik:

1. Zna i rozumie pożądane własności funkcji skrótu. Potrafi uzasadnić czym te wymagania są podyktowane i podać praktyczne przykłady.
2. Rozpoznaje nazwy popularnych funkcji haszujących, potrafi wyszukać informacje na temat obecnie rekomendowanych rozwiązań.
3. Jest świadomy ryzyka ataku na funkcje skrótu. Potrafi ocenić skuteczność powodzenia ataku brutalnego bądź urodzinowego, i zaprogramować w prostym przypadku, na przykład (a) złamać krótkie bądź naiwne hasło mając dany jego skrót (b) wygenerować dwa podobne pliki o identycznym skrócie, dla nie za długich wartości funkcji skrótu

Składowe oceny:

Prace domowe 100%

Ocena końcowa:

liczba punktow	ocena
>= 90%	wyróżnienie

Program zajęć:
Program przewiduje dwa spotkania łączące elementy wykładu (W) i ćwiczeń (C).

Blok zajęciowy	Tematyka
Tydzień 1	W: Kryptograficzne funkcje skrótu, pożądane własności. C: Własność jednokierunkowości i zastosowania: przechowywanie haseł i ochrona pobieranych plików. • Zadanie 1 (C.1.1) Jakie są konkretne przykłady programów których dostawcy poblikują sumy kontrolne? • Zadanie 2 (C.1.2) Sprawdź zgodność opublikowanych w internecie dokumentów z sumami kontrolnymi. • Zadanie 3 (C.1.3) Jak złamać krótkie hasło mając dany jego skrót? • Zadanie 4 (C.1.4) Jak złamać naiwne hasło mając dany jeg skrót?
Tydzień 2	W: Paradoks urodzin i atak na funkcje haszujące. C: Własność bezkolizyjności i zastosowania: sygnatury. • Zadanie 5 (C.3.1) Jak fałszować cyfrowy podpis? Spreparuj dwa podobne dokumenty o tej samej wartości (niedostatecznie mocnej) funkcji skrótu.

Table 1: Harmonogram zajęć

Materiały:
Przydatne skrypty w języku Python zamieszczone są w załączniku A. Przydatne polecenia dla konsoli systemu Linux opisane są w załączniku B. Opisy zadań domowych znajdują się w załączniku C. Dodatkowymi materiałami będą krótkie notatki do zajęć, podsumowujące i porządkujące wiadomości.

A Przydatne skrypty

A.1 Obliczanie funkcji skrótu

```
# -*- coding: utf-8 -*-

napis1 = "ten napis zostanie zahaszowany"

napis2 = "ten napis zostanie  zahaszowany"

# potrzebna biblioteka

import hashlib

# odpowiednia funkcja haszująca - w tym przypadku SHA256

h = hashlib.sha256(napis1)

# metoda 'hexdigest' podaje skrót w systemie 16-tkowym

print h.hexdigest()

# tym razem haszujemy lekko zmnieniony napis.

h = hashlib.sha256(napis2)

print h.hexdigest()
```

A.2 Generowanie permutacji

```
# -*- coding: utf-8 -*-

alfabet = ['a','b','c','d']

# potrzebna użyteczna biblioteka

from itertools import permutations

# skrypt wypisuje wszystkie możliwe przestawienia napisu abcd

for p in permutations(alfabet):
    napis = ''.join( str(x) for x in p )
    print napis
```

B Przydatne polecenia w terminalu (Linux)

Jeżeli chcemy zastosować funkcję haszującą SHA256 do wiadomości m = tu Twoja wiadomość wpisujemy w terminalu

```
echo -n "tu Twoja wiadomość" | shasum256
```

Chcąc obliczyć skrót całego pliku o pełnej nazwie ToMojPlik.txt przy funkcji haszującej SHA256, przechodzimy w konsoli do odpowiedniego katalogu i używamy polecenia

```
shasum256 ToMojPlik.txt
```

lub podajemy pełną ścieżkę

```
shasum256 /pelna_sciezka/ToMojPlik.txt
```

C Prace domowe

C.1 Integralność plików

Kryptograficzne funkcje skrótu są używane do sprawdzania autentyczności plików. Schemat jest następujący:

1. producent/dystrybutor programu F publikuje jego skrót $\#F = \mathsf{Hash}(F)$, dla ustalonej (i znanej!) funkcji skrótu Hash
2. użytkownik który pobrał plik F' (być może z innego źródła!) sprawdza autentyczność przez porównanie $\mathsf{Hash}(F') = \#F$, tj. porównanie skrótu posiadanego pliku ze skrótem opublikowanego oryginału

metoda zawodzi jeżeli plik F' jest inny (uszkodzony lub podmieniony!) ale ma indentyczną wartość funkcji skrótu. Jest to mało prawdopodobne, ponieważ przy założeniu że użyta funkcja Hash ma odpowiednio wysoki poziom bezpieczeństwa, znalezienie dwóch różnych plików o tym samym skrócie jest *niewykonalne obliczeniowo*. Rozwiązanie to jest powszechnie stosowane, patrz np. strona projektu Open Office https://www.openoffice.org/download/

C.1.1 (1pkt) Zadanie 1

Podaj 5 różnych produktów (programów) których producenci umożliwiają werfyfikację zgodności pobieranych plików przez kryptograficzne funkcje skrótu (SHA256, MD5...).

C.1.2 (1pkt) Zadanie 2

W załączniku znajduje się 10 kopii pliku zawierającego rozwnięcie liczby π do 10 tys. miejsc po przecinku w formacie txt. Wiadomo że w dwóch przypadkach do zapisu wkradł się błąd. Podaj które to pliki, i uzasadnij odpowiedź?

C.1.3 (2pkt) Zadanie 3

W załączniku znajduje się 5 plików oraz odpowiadające im skróty SHA256. Sprawdź które z plików są niezgodne z zapisem i wyjaśnij skąd niezgodność?

C.1.4 (2pkt) Zadanie 4

W załączniku znajduje się 20 kopii pewnego dokumentu w formacie doc. Treści wszystkich dokumentów są identyczne, jednak jeden z tych dokumentów różni się od pozostałych. Podaj który to dokument i wyjaśnij różnicę?

C.2 Ochrona haseł

Kryptograficzne funkcje skrótu stosuje się dla ochrony haseł. W rozwiązaniu tym serwer nie przechowuje prawdziwego hasła użytkownika a jedynie jego skrót. W przypadku wycieku informacji trudno będzie ustalić jakie było prawdzie hasło (trudność opiera się na założenie jednokierunkowości). W celu jeszcze większej ochrony stosuje się tzw. zasalanie, czyli wydłużanie haseł o losowy ciąg jeszcze przez haszowaniem. Dla prostoty w poniższych zadaniach zakładamy że nie stosujemy zasalania i że przechowywane dane są skrótami prawdziwych haseł (do poczty e-mail, kodów PIN, itd.)

C.2.1 (2pkt) Zadanie 5

Wiadomo, że 4-cyfrowy 'naiwny' PIN ma skrót

`03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4`

przy funkcji SHA256. Jaki to PIN?

C.2.2 (3pkt) Zadanie 6

Maciek używał do niedawna "naiwnego" hasła `maciek87`. Postanowił je zmienić, ale przestawiając jednie kolejność znaków. Jego skrót to

`019ef75b8e2f98dd874e2c7c7e86d0ebb8eaaa2c62bf00c5146cfdb4d40f2d31`

Jakie jest nowe hasło?

C.3 Bezpieczeństwo podpisów cyfrowych

Załóżmy, że Ewa przedstawia Alicji propozycję kupna samochodu wartego 200tys PLN. Ewa nie zamierza nabyć go jednak uczciwie, płącąc za niego nie więcej niż 50tys PLN. Kontrakt ma być podpisany tzw. podpisem cyfrowym, gdzie - w uproszczeniu - dokument zostanie najpierw zahaszowany a potem przetworzony dalej (w jedneym z wariantów - z wykorzystaniem szyfru RSA). Ewa zamierza stworzyć dwie wersje umowy o identycznej wartości funkcji skrótu, podobnie brzmiące lecz opiewające na rożne kwoty. Można tego dokonać bazując na paradoksie urodzin (przy dużych zasobach). Problem ten pokazuje dlaczego istotna jest *bezkolizyjność*.

C.3.1 Zadanie 7

Wygeneruj (np. poprzez podwajanie spacji) dwa teksty o treści podobnej do poniższego wzorca

`Podpisując tą umowę Alicja zgadza się niniejszym sprzedać Ewie samochód osobowy o dowodzie rejestracyjnym WA 0000 za kwotę` x `PLN, którą Ewa zobowiązuje się uiścić przelewem na konto bankowe`

różniące się jedynie kwotą x, dające identyczny skrót SHA256.