Komprimace dat a kryptologie

Hashovací funkce SHA1

Jonáš Petrovský
Provozně ekonomická fakulta,
Mendelova univerzita v Brně,
Česká republika
jond@post.cz

4. ledna 2016

1 Úvod

Cílem práce je vytvořit aplikaci, která bude implementovat vybranou hashovací funkci. Funkce musí být netriviální, musí být použit vhodný programovací jazyk, aplikace musí být user-friendly.

Bylo zvoleno, že bude implementována funkce SHA1 (Secure Hash Algorithm 1), která vznikla roku 1995. Je faktem, že již není považována za absolutně bezpečnou, jelikož byla popsána možnost jak generovat kolize s časovou složitostí 2^{60}. Přesto zatím nebyly žádné kolize objeveny. Každopádně pro ukázku základních principů hashování lze dobře použít.

2 Použití aplikace

Aplikace je vytvořená v interpretovaném jazyku Python (verze 2.7) a používá pouze součásti základní instalace (není nutné instalovat externí moduly). Jedná se o aplikaci bez GUI, je spouštěná přes příkazový řádek.

Po rozbalení archívu lze aplikaci spustit pomocí souboru `sha1.py`, který se nachází v adresáři `HashProject`. Popis parametrů:

```
python sha1.py [-h] [-s <string>] [-f <filename>] [-v] [input]
```

- `sha1.py -h` ... zobrazí nápovědu.
- `sha1.py -s "test"` ... najde hash pro řetězec “test”.
- `sha1.py -s "test" -v` ... vypíše určité informace během hashování.
- `sha1.py -f "file.txt"` ... najde hash pro soubor “file.txt”.
- `sha1.py -f "file.txt" -v` ... vypíše určité informace během hashování.
- `sha1.py "test"` ... najde hash pro řetězec “test”.

Pokud není zadán přepínač `-s` nebo `-f`, musí být jako vstupní parametr alespoň řetězec. Pokud není přítomen žádný parametr, program vypíše chybu.

Soubor `sha1_test.py` obsahuje automatické testy, které spuštěním souboru proběhnou a poté se zobrazí výsledek. Soubor musí být spuštěn z adresáře `HashProject`, aby správně našel testovací soubory v adresáři `tests`.

2.1 Poznámky

1. Hashování souboru o velikosti 3 MB trvá asi 10 vteřin. Aplikace je tedy dobře použitelná spíše pro menší soubory resp. textové řetězce.
2. Pokud se program spouští interaktivně z konzole, mohou se na některých platformách (na kterých nemá konzole výchozí kódování UTF-8) špatně zobrazovat české znaky. Tento problém se autorovi práce, i přes četné pokusy, bohužel nepodařilo vyřešit.

3 Implementace aplikace

Aplikace má následující strukturu:

`\src\Sha1Algo.py` ... hlavní třída
`\tests` ... soubory pro testování
`\sha1.py` ... výkonný skript – zajišťuje práci s příkazovou řádkou
`\sha1_test.py` ... testovací skript – spouští testy

Jak funguje algoritmus SHA1 popisuje standard z USA nazvaný "FIPS PUB 180-1", který je dostupný např. zde: `http://www.nymphomath.ch/crypto/moderne/fip180-1.html`. V aplikaci je použita standardní verze výpočtu hashe ze sekce 7.

Třída `Sha1Algo` má dvě veřejné metody. Metoda `hash_text` zajišťuje hashování textového řetězce, přičemž je možné zadat jeho kódování (výchozí je UTF-8). Znaky jsou převedeny na číselné kódy (byty) a ty jsou uloženy do pole bytů – objekt typu `bytearray`. Metoda `hash_file` zajišťuje hashování zadaného souboru jakéhokoliv typu. Soubor je otevřen v režimu čtení po bytech a je celý načten do pole bytů.

Samotné hashování má na starost soukromá metoda `_hash_bytes`, která jako argument přijímá pole bytů a vrací výsledný hash v HEX formátu. Probíhá ve stručnosti následovně:

1. Připrav vstup – přidat bit 1, doplnit nuly, přidat délku vstupu.
2. Rozděl vstup – na bloky po 64 bytech.
3. Zpracuj bloky:
   (a) Rozděl blok na 16, 32 bitových, slov – 16×4 B čísel.
   (b) Rozšiř blok na celkem 80 slov (pomocí XORování).
   (c) Zpracuj všechna slova – podle pořadí projdou některou ze 4 funkcí.
   (d) Aktualizuj H proměnné (h0–h4).
4. Převeď H proměnné do HEX podoby a spoj je za sebe – vznikne hash.

Pro práci s byty resp. bity je použit modul `struct` a jeho metody `pack` a `unpack`.

Zdrojový kód aplikace je poměrně dost okomentovaný, takže dále to již asi není nutné rozebírat.

Pro otestování třídy `Sha1Algo` byla vytvořena testovací třída `TestSha1Algo`, která využívá modul `unittest`. Obsahuje dva testy pro řetězce (jeden v AJ a jeden v ČJ) a dva testy pro soubory (o velikosti 302 bytů a 57 kB). Výsledek aplikace se porovnává s výsledkem modulu `hashlib`.