

Komprimace dat a kryptologie

Hashovací funkce SHA1

Jonáš Petrovský
Faculty of Business and Economics,
Mendel University in Brno,
Czech Republic
jond@post.cz

January 2, 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 náročností 2^{60} . Přesto zatím nebyly žádné kolize objeveny. Každopádně pro ukázkou 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 kořenovém adresáři. Popis parametrů:

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

- `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.

Je nutné zadat jeden z parametrů `-s` nebo `-f`. Pokud není zadán ani jeden z nich, program vypíše chybu. Pokud jsou zadány oba dva, použije se pouze `-s`.

Soubor `sha1_test.py` obsahuje automatické testy, které spuštěním souboru proběhnou a poté se zobrazí výsledek.

Poznámka 1: Hashování souboru o velikosti 3 MB trvá asi 10 vteřin. Aplikace je tedy dobře použitelná spíše pro menší soubory.

Poznámka 2: České znaky se ve výstupu do konzole nezobrazí správně, jelikož zajistit toto pro více platforem je velmi obtížné.

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). Znaký jsou převedeny na číselné kódy (byty) a ty jsou uloženy do pole bytů – objekt typu `bytearray`. Metoda `hash_file` zajišťuje hashování zadaného souboru jakéhokoli 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. Připravit vstup – přidat bit 1, doplnit nuly, přidat délku vstupu.
2. Rozdělit vstup – na bloky po 64 bytech.
3. Zpracuj bloky:
 - (a) Rozdělit blok na 16, 32 bitových, slov – 16×4 B čísel.
 - (b) Rozšíř 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`.