Sprawozdanie kryptografia - Funkcje skrótu

Wojciech Chwaciński 151924

Zad1 i 2.

Funkcja obliczająca skróty za pomocą algorytmów dostępnych w bibliotece ‘hashlib’:

```
def hash(wiadomosc, alogorytm):
    if alogorytm == "MD5":
        hs = hashlib.md5()

    elif alogorytm == "SHA-1":
        hs = hashlib.sha1()
    elif alogorytm == "SHA-224":
        hs = hashlib.sha224()
    elif alogorytm == "SHA-384":
        hs = hashlib.sha384()
    elif alogorytm == "SHA-256":
        hs = hashlib.sha256()
    elif alogorytm == "SHA-512":
        hs = hashlib.sha512()

    elif alogorytm == "SHA-3-224":
        hs = hashlib.sha3_224()
    elif alogorytm == "SHA-3-256":
        hs = hashlib.sha3_256()
    elif alogorytm == "SHA-3-384":
        hs = hashlib.sha3_384()
    elif alogorytm == "SHA-3-512":
        hs = hashlib.sha3_512()

    else:
        print("Nie ma takiego algorytmu jak: ", alogorytm)
        return 0

    hs.update(wiadomosc.encode('utf-8'))

    return hs.hexdigest()
```

Funkcja przyjmuję na wejściu wiadomość dla której będzie generowany skrót oraz algorytm za pomocą którego danych skrót ma być wygenerowany. Po wykonaniu algorytmu funkcja zwróci hexadecymalna reprezentację wygenerowanego wcześniej skrótu.

Wyniki działania funkcji skrótu dla danych wejściowych o długości:

-1000000 losowych znaków:

Algorytm hash'ujacy: MD5

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.0

Dlugosc na wyjsciu: 32

Ciag wyjsciowy: 73d41a36d3e76deaa07e8d37c477b7c8

Algorytm hash'ujacy: SHA-1

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.008003950119018555

Dlugosc na wyjsciu: 40

Ciag wyjsciowy: 9ce93badf0776633bffddf0c77d85582a0d6c922

Algorytm hash'ujacy: SHA-224

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.0

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 90cd38ffcf0a4cebe5435aabdd9521bfaf8a4a618d41e43c968e9add

Algorytm hash'ujacy: SHA-256

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.0

Dlugosc na wyjsciu: 64

Ciag wyjsciowy: 1e75f36b2e8cf953e42b15b30329f4905cd5d2303855a3685f62adaa853b18d1

Algorytm hash'ujacy: SHA-384

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.007996559143066406

Dlugosc na wyjsciu: 96

Ciag wyjsciowy:
edb468cfadade1c40b6a7505a3c5412cf61ca54f73349a6f7f817eb37d45bf57aa928b9f06f2e245
a5fa02944a04e6ff

Algorytm hash'ujacy: SHA-512

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.0

Dlugosc na wyjsciu: 128

Ciag wyjsciowy:
872cd9dbc394cb12a7936f262ea532ca8d44e27e6e98630a29d08d62849fbd61389bb091e13b5f
5eb277cce6702a763c1e529b193eb637087dae4ff11f2058c8

Algorytm hash'ujacy: SHA-3-224

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.00800466537475586

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 1bd00f84d46c80c88f81d9a842c2dc72ade3fab3b239beb416fb01a5

Algorytm hash'ujacy: SHA-3-256

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.00842595100402832

Dlugosc na wyjsciu: 64

Ciag wyjsciowy: 6a9714ba29c1b3c29cb0205820e9d1ce262619339e73609a35abf645a8748791

Algorytm hash'ujacy: SHA-3-384

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.009045124053955078

Dlugosc na wyjsciu: 96

Ciag wyjsciowy:
fd5a8c0e12b6d080c38a08d039eb3aa38dff81ff7169fde07ba4e1270c2b3b0dd4db53768883cbe
9aac9bb72d9471bd5

Algorytm hash'ujacy: SHA-3-512

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.015480279922485352

Dlugosc na wyjsciu: 128

Ciag wyjsciowy:
4b7eddf1ccacaaefa647605314067eb998de56af422a743a112f376830e9e127ddafcbdc2caa208605b50b2366dec1065c68faf1297e9393ebb7718a50bc13b4

-**10000000 losowych znaków:**

Algorytm hash'ujacy: MD5

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.03323793411254883

Dlugosc na wyjsciu: 32

Ciag wyjsciowy: b5aeb2731f68b96f68b280e1fe417bfe

Algorytm hash'ujacy: SHA-1

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.01612257957458496

Dlugosc na wyjsciu: 40

Ciag wyjsciowy: 6eb3c435d079f23d8f6553d5315f4c90ac164212

Algorytm hash'ujacy: SHA-224

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.016307592391967773

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 9a019abae23a22d8bbcc88305b840fbc2895a65c273ba220bc9d81aa

Algorytm hash'ujacy: SHA-256

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.016992568969726562

Dlugosc na wyjsciu: 64

Ciag wyjsciowy: cb726774dbff0218c601e7f170512af2c1569ae73cf5e986bb39121e0d04aed2

Algorytm hash'ujacy: SHA-384

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.03359818458557129

Dlugosc na wyjsciu: 96

Ciag wyjsciowy:
476865419e6662f89239c6398fb49f3353ab1ce2bfba5c2f2aeacd1cee47303f64066802a4ff3164e8673e081258b9cb

Algorytm hash'ujacy: SHA-512

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.03329205513000488

Dlugosc na wyjsciu: 128

Ciag wyjsciowy:
f11366515be34860dc12318bc60c297aba0357cf537b30dac7cf388e39b85afef2113a77bb5b404b1b9e8df956170c10ce972f0088b09f65a8f6feb9f7f586a1

Algorytm hash'ujacy: SHA-3-224

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.05805540084838867

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 81ea96fa58b889fb60d3e5fb7217b2302bcb58de1559052b05d17296

Algorytm hash'ujacy: SHA-3-256

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.10570025444030762

Dlugosc na wyjsciu: 64

Ciag wyjsciowy: 4f5c8d6d640367d348c77a5e89fdeb20366ab1f2ebb5b71f9a94b3bc4eb43074

Algorytm hash'ujacy: SHA-3-384

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.06684732437133789

Dlugosc na wyjsciu: 96

Ciag wyjsciowy:
d0512777718c93ad45fa0dab36636fe4c6c51ed31e3ef4c36ddb6bfb3ab576f087f4fcf8999962f94bd6f872eee5aa9e

Algorytm hash'ujacy: SHA-3-512

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.13390421867370605

Dlugosc na wyjsciu: 128

Ciag wyjsciowy:
d531ebf4c2fd7ef33d2d7f138b107aba22c8c413576c21ad1cda2ac410b9469f409e875e2f8f18df
19ac5e6f8107e20ee46479089c424a68870f919c5c45abd6

-100000000 losowych znaków:

Algorytm hash'ujacy: MD5

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.3388509750366211

Dlugosc na wyjsciu: 32

Ciag wyjsciowy: 94a67e8950360818a530eafe22ade7f3

Algorytm hash'ujacy: SHA-1

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.14000749588012695

Dlugosc na wyjsciu: 40

Ciag wyjsciowy: 56e1a2e196c066bdd0b27e2dd64f4e59a217a074

Algorytm hash'ujacy: SHA-224

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.2743351459503174

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 0f074c2a8e94023abe9e63a6a15f7e8743536630fc9e2bbe78f38348

Algorytm hash'ujacy: SHA-256

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.14690279960632324

Dlugosc na wyjsciu: 64

Ciag wyjsciowy: 6a47340ba525d61b5734b9eb3918b22e8a37650ddb930fbeea023c4a4cadb6e5

Algorytm hash'ujacy: SHA-384

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.42431068420410156

Dlugosc na wyjsciu: 96

Ciag wyjsciowy:
f52921a6cf684c9043f00e5a3a6b73621157a6d87c208d71b35c0da395b5f3d112a3bd826292eca1d6931ad2f3385469

Algorytm hash'ujacy: SHA-512

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.35126566886901855

Dlugosc na wyjsciu: 128

Ciag wyjsciowy:
d99d2347cd63497315c9be710ef7ec12c66c9bdf7d6b61d17d38533d885fcb719a71fb0f6b637345b744826c2f3dc71e1226b7fdaf6a86e73f778be02e01a53c

Algorytm hash'ujacy: SHA-3-224

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.5989181995391846

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 631b3c9028b4ef174203084ff1c50ace9eccbaec88fab7d75416ccff

Algorytm hash'ujacy: SHA-3-256

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.6558928489685059

Dlugosc na wyjsciu: 64

Ciag wyjsciowy: b8fa162977ebc438657d08f3ce13af56de4751aa6e908dcb2c33beab6e279d80

Algorytm hash'ujacy: SHA-3-384

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 0.7973756790161133

Dlugosc na wyjsciu: 96

Ciag wyjsciowy:
b13df15e77f59c32be8c34bae6c2d4489c465e7cd93e53b0064b99cfd7a486c8146f425d2de9446b4f284383b66cd951

Algorytm hash'ujacy: SHA-3-512

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 1.2672154903411865

Dlugosc na wyjsciu: 128

Ciag wyjsciowy:
2ef9a128d821556c87a5eba32ff88a8ce3e0c892cc7c8b944406ea0437730525a3483478a1124d7894a53915f17f16fb13f9abb66d0d1ebbe0c8683f14722ce4

Wnioski.

Czasy działania funkcji skrótu są bardzo niskie nawet dla bardzo długich ciągów znaków (zaczęły być zauważalne dopiero przy ciągach długości 1000000 znaków), jednakże poprzez obserwację pozyskanych wyników możemy zauważyć że czas w miarę zwiększania instancji wejściowej zwiększa się znacznie względem poprzedniej instancji np.:

Algorytm hash'ujacy: SHA-3-512

Ciag wejsciowy dlugosc: 1000000

Czas dzialania w [s]: 0.015480279922485352

Algorytm hash'ujacy: SHA-3-512

Ciag wejsciowy dlugosc: 10000000

Czas dzialania w [s]: 0.13390421867370605

Algorytm hash'ujacy: SHA-3-512

Ciag wejsciowy dlugosc: 100000000

Czas dzialania w [s]: 1.2672154903411865

Widzimy na tym przykładzie że czasy z każdym zwiększeniem instancji o 10 razy również zwiększały się podobną ilość razy. Ważną rzeczą którą również wykazał ten eksperyment jest to że funkcje skrótu generują zawsze ciągi tej samej długości nie zależnie od wielkości instancji wejściowej np.:

Algorytm hash'ujacy: SHA-224

Ciag wejsciowy dlugosc: 1000000

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 90cd38ffcf0a4cebe5435aabdd9521bfaf8a4a618d41e43c968e9add

Algorytm hash'ujacy: SHA-224

Ciag wejsciowy dlugosc: 10000000

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 9a019abae23a22d8bbcc88305b840fbc2895a65c273ba220bc9d81aa

Algorytm hash'ujacy: SHA-224

Ciag wejsciowy dlugosc: 100000000

Dlugosc na wyjsciu: 56

Ciag wyjsciowy: 0f074c2a8e94023abe9e63a6a15f7e8743536630fc9e2bbe78f38348

Zad3.

Wartość którą wygenerowałem jest powszechnie znana co oznacza że gdy tworzymy bardzo krótkie hasła musimy się liczyć z tym że skróty do nich generowane przez takie funkcje jak MD5 są znane co czyni takie hasła bardzo słabymi.

Zad4.

W dzisiejszych czasach nie możemy uznać funkcji MD5 za bezpieczną gdyż możliwe jest doprowadzenie do kolizji (praktyczny atak przeprowadzony w 2004 przez Xiaoyun Wang i Hongbo Yu) co dyskwalifikuję MD5, gdyż kolizję pozwalają na fałszowanie podpisów cyfrowych ponad to hasła haszowne przez MD5 są stosunkowo łatwe do złamania atakami siłowymi i słownikowymi.

Zad5.

Liczba kolizji dla funkcji skrótu SHA-256 szukanych na pierwszych bitach (od 0 – 99 pierwszych bitów) równała się średnio połowie sprawdzanych bitów (50,61%)

Liczba kolizji w pierwszych 1 bitach dla losowych ciagow wejsciowych: 1

Liczba kolizji w pierwszych 2 bitach dla losowych ciagow wejsciowych: 1

Liczba kolizji w pierwszych 3 bitach dla losowych ciagow wejsciowych: 2

Liczba kolizji w pierwszych 4 bitach dla losowych ciagow wejsciowych: 3

Liczba kolizji w pierwszych 5 bitach dla losowych ciagow wejsciowych: 2

Liczba kolizji w pierwszych 6 bitach dla losowych ciagow wejsciowych: 4

Liczba kolizji w pierwszych 7 bitach dla losowych ciagow wejsciowych: 6

Liczba kolizji w pierwszych 8 bitach dla losowych ciagow wejsciowych: 5

Liczba kolizji w pierwszych 9 bitach dla losowych ciagow wejsciowych: 6

Liczba kolizji w pierwszych 10 bitach dla losowych ciagow wejsciowych: 5

...

Liczba kolizji w pierwszych 90 bitach dla losowych ciagow wejsciowych: 45

Liczba kolizji w pierwszych 91 bitach dla losowych ciagow wejsciowych: 47

Liczba kolizji w pierwszych 92 bitach dla losowych ciagow wejsciowych: 47

Liczba kolizji w pierwszych 93 bitach dla losowych ciagow wejsciowych: 39

Liczba kolizji w pierwszych 94 bitach dla losowych ciagow wejsciowych: 54

Liczba kolizji w pierwszych 95 bitach dla losowych ciagow wejsciowych: 44

Liczba kolizji w pierwszych 96 bitach dla losowych ciagow wejsciowych: 51

Liczba kolizji w pierwszych 97 bitach dla losowych ciagow wejsciowych: 51

Liczba kolizji w pierwszych 98 bitach dla losowych ciagow wejsciowych: 53

Liczba kolizji w pierwszych 99 bitach dla losowych ciagow wejsciowych: 56

Zad6.

Procentowa wyrażona liczba bitów, które uległy zmianie przy porównaniu stringów „kotek” i „koutek”, wynosi 50,39% .

Pyt.: Określ rolę soli w tworzeniu skrótów.

Odp.: Sole są losowymi, unikalnymi ciągami danych dodanymi do początkowej wiadomości przed użyciem na niej funkcji skrótu, dzięki temu zwiększane jest bezpieczeństwo haszy co jest szczególnie ważne w przypadku haseł np.: gdy pojawią się dwa takie same hasła po użyciu soli ich hasze będą różne co utrudni potencjalnemu napastnikowi złamanie haseł.