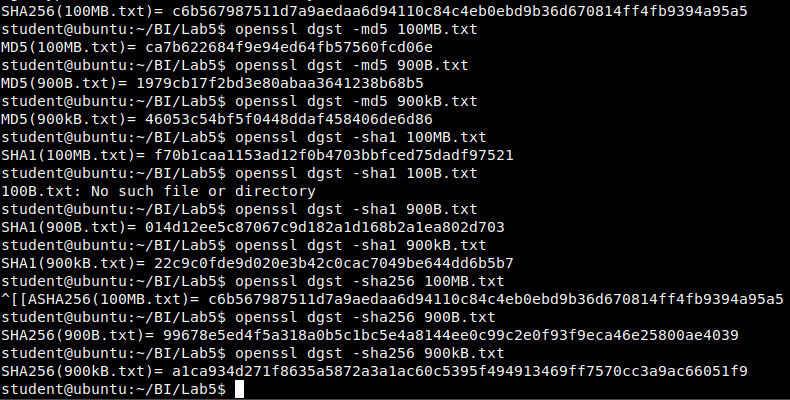
**Jakub Dudek**

**Sprawozdanie Laboratorium 5**

**P.5.1. Dla każdego z tych plików oblicz wartość funkcji skrótu za pomocą trzech wybranych algorytmów wykorzystując funkcjonalność openssl. Omów i porównaj otrzymane skróty wiadomości.**

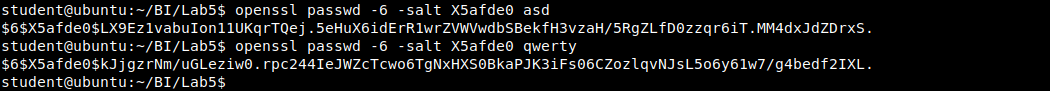
****

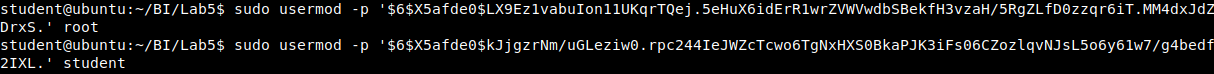
Dla tego samego zbioru danych zawsze jest ten sam skrót kryptograficzny zależnie od użytego algorytmu. Długość klucza jest zależna od użytego algorytmu.

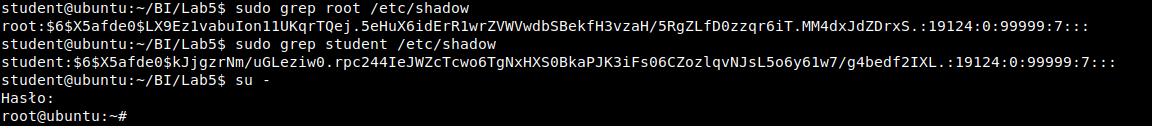
**P.5.2. Określ średni czas przetworzenia 1MB danych dla każdej z kombinacji: rozmiar pliku – algorytm obliczenia skrótu. W celu określenia czasu wykonania polecenia skorzystaj z systemowego polecenia "time". Jako wynik zapisz sumy czasów user+sys. Wynik należy zapisać, jako średnia z minimum 100 pomiarów. Otrzymane wyniki umieścić w tabeli, a następnie sporządzić wykresy dla zastosowanych algorytmów.**

****

**P.5.3. Jakie zmiany zauważyłeś w pliku shadow? Zaloguj się na konta root lub student. Jakiego hasła użyłeś? Udokumentuj wykonanie ćwiczenia. Omów wykonane ćwiczenie i jego efekty. Czy dana metoda zmiany hasła jest bezpieczna? Jeśli nie zaproponuj zmiany i wyjaśnij dlaczego.**

****

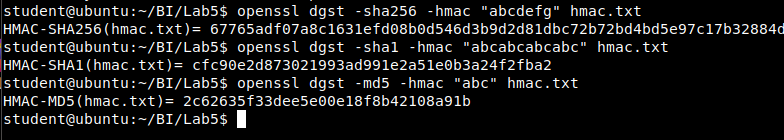
****

****

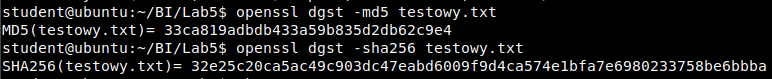
W pliku shadow zmieniły się skróty kryptograficzne haseł dla root i student. Użyłem hasła „asd”, aby zalogować się na konto root. Taka metoda nie jest bezpieczna. Ktoś może podejrzeć skrót kryptograficzny, który wygenerowałem lub hasło, które wpisałem w poleceniu oraz taką metodą pomijam wszelkie reguły ustawiania hasła.

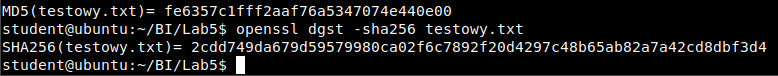
**P.5.4. Czy w algorytmie HMAC konieczne jest używanie kluczy o stałej długości?** **P.5.5. Jeśli tak, to, o jakiej długości? Odpowiedź uzasadnij w oparciu o wyniki przeprowadzonych testów.**

Nie jest konieczne. Jednak, gdy podamy za krótki zostanie uzupełniany zerami, a gdy jest za długi zostanie obcięty.



**P.5.6. Proszę przeanalizować otrzymane wyniki i odpowiedzieć w sprawozdaniu na pytanie czy H1 oraz H2 są podobne czy też nie. Odpowiedź należy uzasadnić.**

****

****

H1 i H2 nie są podobne. Skrót kryptograficzny zależy od danych w pliku.

**P.5.7. Proszę napisać program obliczający ile jest zgodnych bitów w H1 oraz H2.**

#!/bin/bash

ile=0

p1=`openssl dgst -sha256 $1 | cut -d " " -f2`

p2=`openssl dgst -sha256 $2 | cut -d " " -f2`

l=${#p1}

let "l=l-1"

#printf "$p1\n$p2\n"

for i in `seq 0 $l`

do

x=${p1:$i:1}

y=${p2:$i:1}

a=$(printf '%04d\n' $(echo "ibase = 16; obase = 2; ${x^^}" | bc))

b=$(printf '%04d\n' $(echo "ibase = 16; obase = 2; ${y^^}" | bc))

#printf "$a\t$b\n"

for j in {0..3}

do

if [ ${a:$j:1} == ${b:$j:1} ]

then

let "ile+=1"

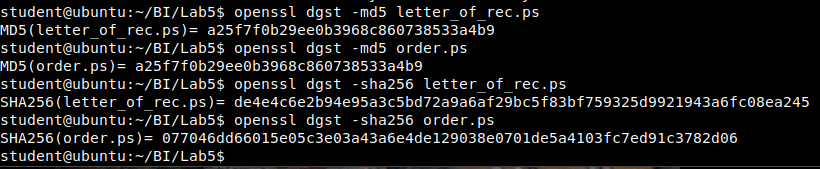
fi

done

done

echo "Zgodnych bitow: $ile"

**P.5.8. Czy pliki po rozpakowaniu się różnią? Zamieść odpowiednie zrzuty z wygenerowanymi wartościami funkcji skrótu. Omów czy otrzymane z plików wartości funkcji skrótu dla danych algorytmów są takie same czy różne? Omów wpływ stosowania algorytmów starszego typu na poziom bezpieczeństwa „haszowania”. Czy użyłbyś algorytmu MD5 do podpisywania plików? Odpowiedź uzasadnij.**



Pliki po rozpakowaniu różnią się. Przy zastosowaniu algorytmu md5 otrzymane skróty kryptograficzne są takie same, a przy sha256 różnią się.

Algorytmy starszego typu mają większą kolizję od nowych, czyli jest większe prawdopodobieństwo, że para wiadomości ma taką samą wartość skrótu. Do starszych algorytmów jest dużo tęczowych tabel. Nie jest to więc bezpieczne.

Nie użyłbym z powodów powyżej.