



Nazwa implementacji: Szyfrowanie

Autor:

Adam Jurkiewicz

Opis implementacji: Zaawansowane użytkowanie Linuksa, podstawowe informacje o bezpieczeństwie danych i szyfrowaniu.

- Na kartce rozpisz szyfrowanie w stylu ROT-13, np.:

A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	R	S	T	U

I w ten sposób zapisz zaszyfrowane zdanie: Linux jest super. Porównaj z pracami innych.

W katalogu domowym tworzymy plik tekstowy za pomocą menadżera Nautilus.

W terminalu sprawdzamy, czy plik istnieje (polecenie ls).

Wworzymy klucze dla siebie.

(gpg --gen-key)





```
Terminal - uczen@swol-uczen: ~
Plik Edycja Widok Terminal Karty Pomoc
uczen@swol-uczen:~$ gpg --gen-key
gpg (GnuPG) 1.4.11; Copyright (C) 2010 Free Software Foundation, Inc.
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: zbiór kluczy './home/uczen/.gnupg/secring.gpg' został utworzony
gpg: zbiór kluczy './home/uczen/.gnupg/pubring.gpg' został utworzony
Proszę wybrać rodzaj klucza:
(1) RSA i RSA (domyślne)
(2) DSA i ElGamala
(3) DSA (tylko do podpisywania)
(4) RSA (tylko do podpisywania)
Twój wybór? 1
RSA klucze mogą mieć pomiędzy 1024 a 4096 bitów.
Jakiej długości klucz wygenerować? (2048)
Żądana długość klucza to 2048 bitów.
Okres ważności klucza.
  0 = klucz nie ma określonego terminu ważności
  <n> = termin ważności klucza upływa za n dni
  <n>w = termin ważności klucza upływa za n tygodni
  <n>m = termin ważności klucza upływa za n miesięcy
  <n>y = termin ważności klucza upływa za n lat
Okres ważności klucza ? (0)
Key does not expire at all
w ogóle nie wygasa
Czy wszystko się zgadza (y/n)? y

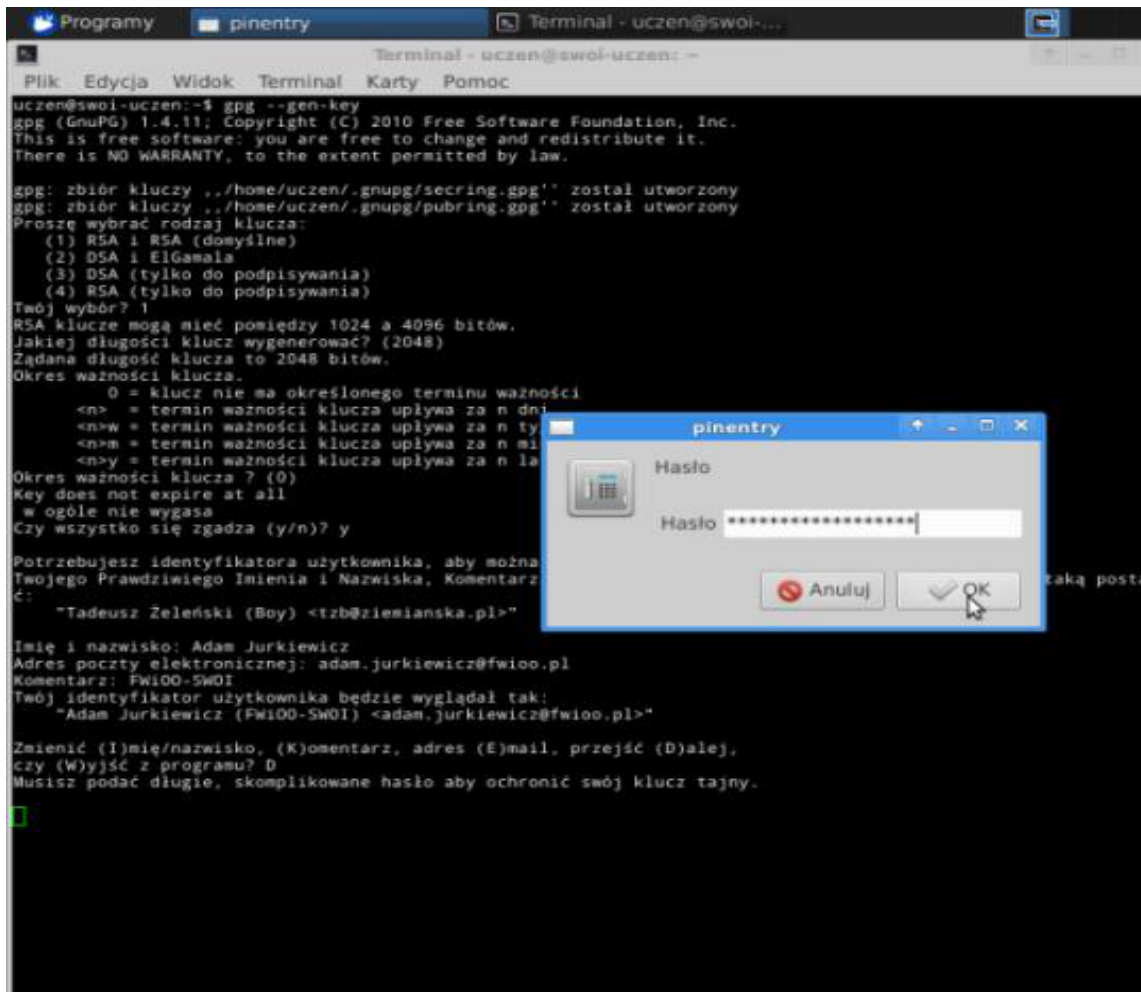
Potrzebujesz identyfikatora użytkownika, aby można było rozpoznać Twój klucz; program złoży go z
Twojego Prawdziwego Imienia i Nazwiska, Komentarza i Adresu E-mail. Będzie on miał, na przykład, taką posta
ć:
  "Tadeusz Żeleński (Boy) <tzb@ziemianska.pl>"

Imię i nazwisko: Adam Jurkiewicz
Adres poczty elektronicznej: adam.jurkiewicz@fwioo.pl
Komentarz: FW100-SW01
Twój identyfikator użytkownika będzie wyglądał tak:
  "Adam Jurkiewicz (FW100-SW01) <adam.jurkiewicz@fwioo.pl>"

Zmienić (I)mię/nazwisko, (K)omentarz, adres (E)mail, przejść (D)alej,
czy (W)yjść z programu? █
```

Aby chronić nasz klucz w bazie kluczy systemu, konieczne będzie wymyślenie długiego i skomplikowanego hasła – nie wolno go zapomnieć!





- Należy również pamiętać, że dla dobrego klucza konieczne jest dużo danych losowych. Można to uzyskać, dając systemowi dużo operacji dyskowych do wykonania. Ja proponuję w tym momencie uruchomić program SYSTEM | Analizator wykorzystania Dysku - to pozwoli systemowi generować dużo danych.





Katalog	Użycie	Rozmiar	Zawartość
/	--%		Skanowanie...
home	--%	6,2 GB	2 elementy
var	--%	447,8 MB	10 elementów
opt	--%	82,4 MB	2 elementy
boot	--%	34,4 MB	8 elementów
etc	--%	26,7 MB	288 elementów
srv	--%	204,8 kB	1 element
tmp	--%	53,2 kB	9 elementów
selinux	--%	4,1 kB	0 elementów
cdrom	--%	4,1 kB	0 elementów
dev	--%	4,1 kB	12 elementów
usr	--%		Skanowanie...

- W tym momencie mamy już wygenerowany nasz klucz PGP:





```
Terminal - uczen@swoi-uczen: -
Plik Edycja Widok Terminal Karty Pomoc
<n>= termin ważności klucza upływa za n miesięcy
<n>y = termin ważności klucza upływa za n lat
Okres ważności klucza ? (0)
Key does not expire at all
w ogóle nie wygasa
Czy wszystko się zgadza (y/n)? y

Potrzebujesz identyfikatora użytkownika, aby można było rozpoznać Twój klucz; program złoży go z
Twojego Prawdziwego Imienia i Nazwiska, Komentarza i Adresu E-mail. Będzie on miał, na przykład, taką postać:
" Tadeusz Żeleński (Boy) <tzb@ziemianska.pl>"

Imię i nazwisko: Adam Jurkiewicz
Adres poczty elektronicznej: adam.jurkiewicz@fw100.pl
Komentarz: FW100-SWOI
Twój identyfikator użytkownika będzie wyglądał tak:
"Adam Jurkiewicz (FW100-SWOI) <adam.jurkiewicz@fw100.pl>"

Zmienić (I)mię/nazwisko, (K)omentarz, adres (E)mail, przejść (D)alej,
czy (W)yjść z programu? D
Musisz podać długie, skomplikowane hasło aby ochronić swój klucz tajny.

Musimy wygenerować dużo losowych bajtów. Dobrym pomysłem aby pomóc komputerowi
podczas generowania liczb pierwszych jest wykonywanie w tym czasie innych
działań (pisanie na klawiaturze, poruszanie myszką, odwołanie się do dysków);
dzięki temu generator liczb losowych ma możliwość zebrania odpowiedniej ilości
entropii.

Brakuje możliwości wygenerowania odpowiedniej liczby losowych bajtów.
Proszę kontynuować inne działania aby system mógł zebrać odpowiednią
ilość entropii do ich wygenerowania (brakuje 284 bajtów).
.....*****
..*****
Musimy wygenerować dużo losowych bajtów. Dobrym pomysłem aby pomóc komputerowi
podczas generowania liczb pierwszych jest wykonywanie w tym czasie innych
działań (pisanie na klawiaturze, poruszanie myszką, odwołanie się do dysków);
dzięki temu generator liczb losowych ma możliwość zebrania odpowiedniej ilości
entropii.
.....*****
.....*****
gpg: /home/uczen/.gnupg/trustdb.gpg: baza zaufania utworzona
gpg: klucz B01CA7BD został oznaczony jako obdarzony absolutnym zaufaniem
klucz publiczny i prywatny (tajny) zostały utworzone i podpisane.

gpg: sprawdzanie bazy zaufania
gpg: potrzeba 3 marginalny(ch), 1 zupełny(ch), model zaufania PGP
gpg: głębia: 0 poprawność: 1 podpisany: 0 zaufany: 0-, 0q, 0n, 0a, 0f, 1u
pub 2048R/B01CA7BD 2013-08-09
Odcisk palca = 6EE6 319F E888 C54B FDC2 26E5 F9D8 83D4 B01C A7BD
uid Adam Jurkiewicz (FW100-SWOI) <adam.jurkiewicz@fw100.pl>
sub 2048R/F92BCD42 2013-08-09
uczen@swoi-uczen:~$
```

Eksportujemy klucze na serwer kluczy.
(gpg --keyserver hkp://keys.gnupg.net --send-keys key_ID)





```
Terminal - uczen@swoi-uczen: ~
Plik Edycja Widok Terminal Karty Pomoc
uczen@swoi-uczen:~$ gpg --list-keys
/home/uczen/.gnupg/pubring.gpg
-----
pub 2048R/B01CA7BD 2013-08-09
uid          Adam Jurkiewicz (FW100-SWOI) <adam.jurkiewicz@fw100.pl>
sub 2048R/F92BCD42 2013-08-09

uczen@swoi-uczen:~$ gpg --keyserver hkp://keys.gnupg.net --send-keys B01CA7BD
gpg: wysłanie klucza B01CA7BD na hkp serwera keys.gnupg.net
uczen@swoi-uczen:~$
```

Na serwerze kluczy wyszukujemy klucze innych osób.
(`gpg --keyserver hkp://keys.gnupg.net --search-keys key_ID`)

- Importujemy wybrany klucz innej osoby.
(`gpg --keyserver hkp://keys.gnupg.net --recv-keys key_ID`)

```
Terminal - uczen@swoi-uczen: ~
Plik Edycja Widok Terminal Karty Pomoc
uczen@swoi-uczen:~$ gpg --keyserver hkp://keys.gnupg.net --search-keys adam.jurkiewicz@fw100.pl
gpg: szukanie "adam.jurkiewicz@fw100.pl" od hkp serwera keys.gnupg.net
(1) Adam Jurkiewicz (FW100-SWOI) <adam.jurkiewicz@fw100.pl>
    2048 bit RSA key B01CA7BD, utworzony: 2013-08-09
(2) Adam Jurkiewicz (FW100) <adam.jurkiewicz@fw100.pl>
    2048 bit RSA key 5AB4D85D, utworzony: 2012-07-27
(3) Adam Jurkiewicz (FW100) <adam.jurkiewicz@fw100.pl>
    2048 bit RSA key B1F0E79A, utworzony: 2012-07-22
(4) Adam Jurkiewicz (Wolontariusz Fundacji W100) <adam.jurkiewicz@fw100.pl>
    2048 bit RSA key 138A1D29, utworzony: 2011-08-18
Keys 1-4 of 4 for "adam.jurkiewicz@fw100.pl". Podaj numer(y), N)astępy, lub Q) zamknij >
Podaj numer(y), N)astępy, lub Q) zamknij > q
uczen@swoi-uczen:~$ gpg --keyserver hkp://keys.gnupg.net --recv-keys 5AB4D85D
gpg: zapytanie o klucz 5AB4D85D z hkp serwera keys.gnupg.net
gpg: klucz 5AB4D85D: zaimportowano klucz publiczny „Adam Jurkiewicz (FW100) <adam.jurkiewicz@fw100.pl>”
gpg: Ogółem przetworzonych kluczy: 1
gpg: dołączono do zbioru: 1 (RSA: 1)
uczen@swoi-uczen:~$ gpg --keyserver hkp://keys.gnupg.net --recv-keys 5AB4D85D
gpg: zapytanie o klucz 5AB4D85D z hkp serwera keys.gnupg.net
gpg: klucz 5AB4D85D: "Adam Jurkiewicz (FW100) <adam.jurkiewicz@fw100.pl>" bez zmian
gpg: Ogółem przetworzonych kluczy: 1
gpg: bez zmian: 1
uczen@swoi-uczen:~$
```

- Szyfrujemy własny plik własnym key_ID.
(`gpg -r key_ID -e nazwa_pliku`)





```
Terminal - uczen@swoi-uczen: ~
Plik Edycja Widok Terminal Karty Pomoc
uczen@swoi-uczen:~$ gpg --list-keys
/home/uczen/.gnupg/pubring.gpg
-----
pub   2048R/B01CA7BD 2013-08-09
uid         Adam Jurkiewicz (FWi00-Sw01) <adam.jurkiewicz@fwioo.pl>
sub   2048R/F92BCD42 2013-08-09

pub   2048R/5AB4D85D 2012-07-27
uid         Adam Jurkiewicz (FWi00) <adam.jurkiewicz@fwioo.pl>
sub   2048R/AD5AA9F2 2012-07-27

uczen@swoi-uczen:~$ ls
Arduino_Lib      Dokumenty  Obrazy    Publiczny  Scratch Projects  Szablony      Wideo
Arduino_Projekty Muzyka     Pobrane   Pulpit    sketchbook        wzne_dane.txt workspace
uczen@swoi-uczen:~$ cat wzne_dane.txt
***
To jest plik z ważnymi danymi
---

uczen@swoi-uczen:~$ gpg -r B01CA7BD -e wzne_dane.txt
uczen@swoi-uczen:~$ ls
Arduino_Lib      Dokumenty  Obrazy    Publiczny  sketchbook        wzne_dane.txt.gpg
Arduino_Projekty Muzyka     Pobrane   Pulpit    Szablony         Wideo
Dokumenty       Pobrane   Scratch Projects  wzne_dane.txt  workspace
uczen@swoi-uczen:~$ cat wzne_dane.txt.gpg
****
-----BEGIN PGP MESSAGE-----
Version: 1.0
mQIwX-milI...
-----END PGP MESSAGE-----
uczen@swoi-uczen:~$
```

Zadanie 1 - ćwiczenie:

Zaszyfruj plik tekstowy, następnie skasuj go. Potem dzięki manualowi systemowemu sprawdź, jakie opcje trzeba dać dla polecenia gpg, aby odszyfrować zaszyfrowany plik. Sprawdź, czy ci się to udało poprawnie i czy widzisz tekst w pliku.

Zadanie 2 - ćwiczenie:

Zaszyfruj plik tekstowy, korzystając z key_ID innej osoby, zaimportowanego z keyserversa. Potem skopiuj plik do osoby, której klucz był użyty do szyfrowania. Sprawdź, czy ci się to udało poprawnie i czy druga widzi tekst po rozszyfrowaniu swoim **key_ID**.

key_ID

= w powyższych poleceniach key_ID należy zamienić odpowiednim ID klucza, które powstaje podczas generowania nowego klucza.

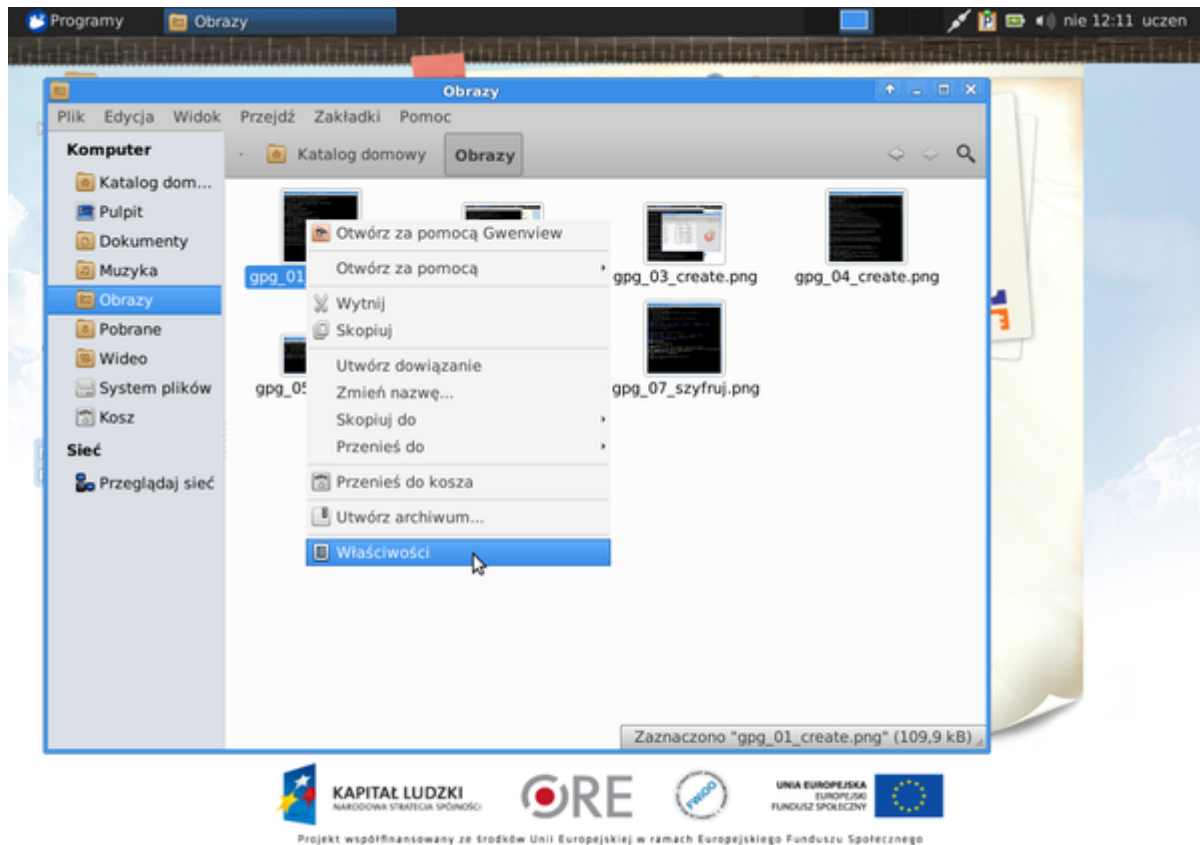
W tych przykładach key_id = B01CA7BD





Dla weryfikacji poprawności danych przeprowadzamy ćwiczenie:

W menadżerze plików zrób kopię obrazu pobranego z internetu na pierwszych zajęciach, a następnie funkcją Prawy Przycisk Myszy - Właściwości | Sumy Kontrolne sprawdź sumy kontrolne MD5 obu plików i porównaj, czy są identyczne.



Następnie otwórz program PROGRAMY | Programowanie | Ghex i otwórz w nim plik kopii obrazka. Zmień 2gi bajt pliku na 00. Zapisz zmiany.



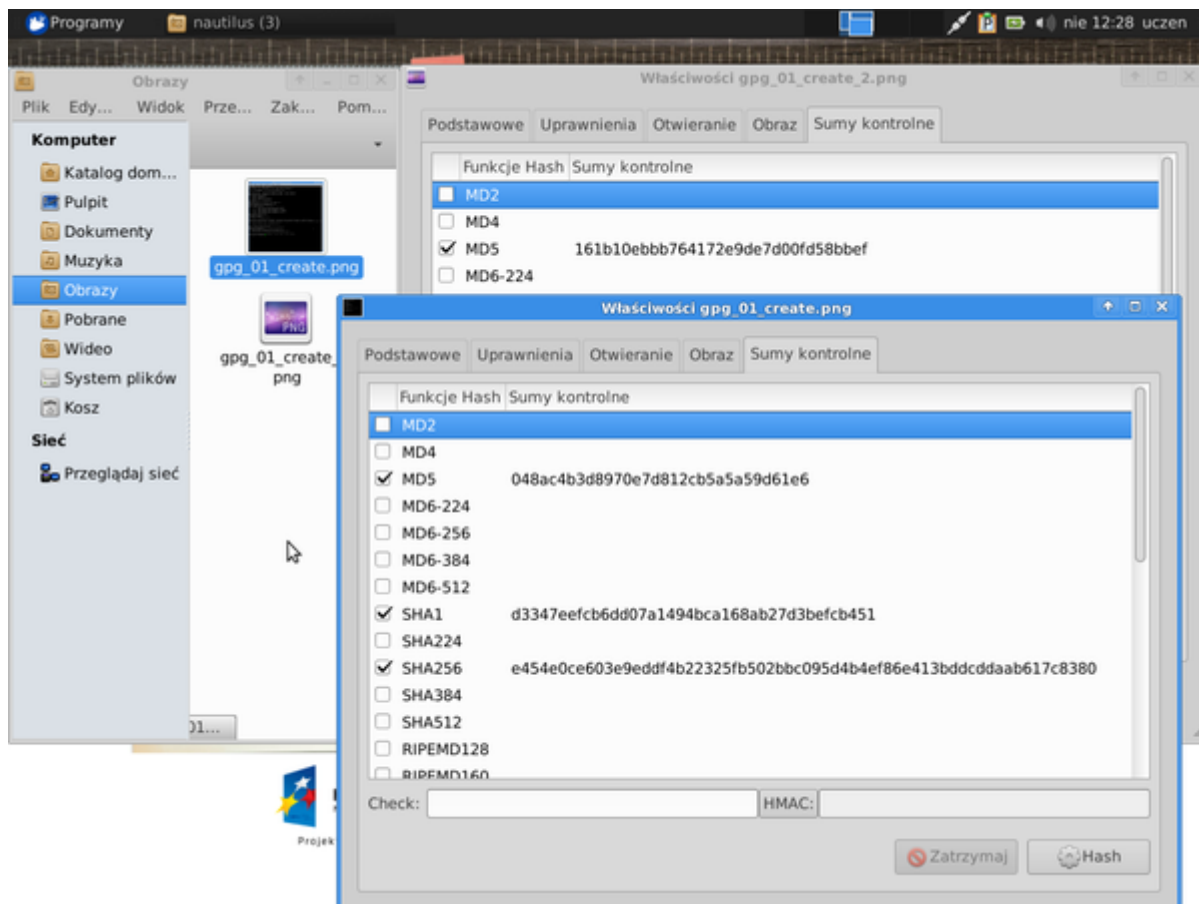
The screenshot shows the GHex application window titled 'pgg_01_create.png - GHex'. The main area displays hex and ASCII data for a PNG file. The right sidebar shows a file menu with 'Zapisz jako...' selected. The bottom panel shows various metadata fields:

Podpisane 8 bitów:	80	Podpisane 32 bity:	222776912	Szesnastkowo:	50
Niepodpisane 8 bitów:	80	Niepodpisane 32 bity:	222776912	Ósemkowo:	120
Podpisane 16 bitów:	20048	Zmiennoprzecinkowe 32 bity:	6,141587e-31	Binarnie:	01010000
Niepodpisane 16 bitów:	20048	Zmiennoprzecinkowe 64 bity:	1,404816e-308	Długość potoku:	8

Additional options at the bottom include checkboxes for 'Wyświetlanie dekodowania cienkości' (checked) and 'Wyświetlanie niepodpisanych i zmiennoprzecinkowych jako szesnastkowe' (unchecked). The status bar shows 'Przesunięcie: 1'.

Następnie sprawdź ponownie sumy kontrolne MD5 dla tych plików. Omów różnice.





Zadanie dla dociekliwych:

Sprawdź w manualu systemowym opis polecenia `md5sum` i wykonaj analogiczne czynności sprawdzające, ale z wykorzystaniem tego polecenia (pamiętaj o przekierowaniu wyników polecenia do pliku poprzez potok systemowy „>”) oraz z wykorzystaniem funkcji sprawdzającej `md5sum`.

