GnuPG

Położenie: (nie dotyczy)

Ogólnie

Szyfrowanie symetryczne – istnieje jeden klucz (hasło), którym szyfruje się dane i rozszyfrowuje, np. AES (*Advanced Encryption Standard*), 3DES, Blowfish;
Szyfrowanie asymetryczne – istnieją dwa klucze (publiczny i prywatny), np. RSA, DSA.

GnuPG

GPG (*Gnu Privacy Guard*) to wolny, darmowy odpowiednik starego komercyjnego PGP (*Pretty Good Privacy*). Posiada parę kluczy: jeden z nich służy do szyfrowania (klucz publiczny, znany wszystkim, każdy może go użyć do zaszyfrowania dowolnych wiadomości), a drugi do deszyfracji (klucz prywatny, np. przechowywany na pendrivie; klucz ten dodatkowo chroniony jest hasłem).
Do obsługi programu można korzystać z graficznej nakładki „*Seahorse*" oraz z nakładki „*Kleopatra*", służącej głównie do zarządzania certyfikatami. Zasada działania jest następująca:

1. Generujemy swój własny klucz i wysyłamy go do wszystkich znajomych, z którymi chcemy prowadzić zaszyfrowaną korespondencję. Oni za pomocą naszego klucza publicznego będą szyfrować wiadomości przeznaczone dla nas. My za pomocą klucza prywatnego, będziemy te wiadomości deszyfrować.

2. Nasi znajomi powinni przesłać nam swoje własne wygenerowane klucze publiczne, za pomocą których my będziemy szyfrować wiadomości do nich pisane. Zaszyfrowany plik będzie miał końcówkę **.gpg*. Oni będą te wiadomości deszyfrować za pomocą swoich kluczy prywatnych.

Kolejność działań:

Tworzenie kluczy

\# **gpg --gen-key** (zgodzić się na domyślne propozycje; tworzony jest folder *.gnupg*, w którym znajduje się plik *pubring.gpg* z kluczami publicznymi oraz *secring.gpg* z kluczami prywatnymi; sam folder musi mieć prawa 700)
\# **gpg --list-keys** (sprawdzamy zawartość bazy danych z kluczami publicznymi)
\# **gpg --list-secret-keys** (sprawdzamy zawartość bazy danych z kluczami prywatnymi)

Publikowanie klucza

\# **gpg --keyserver „hkp://subkeys.pgp.net" --send-key identyfikatorKlucza** (wysyłamy klucz publiczny na publiczny serwer kluczy, np. *pgp.mit.edu, keyserver.pgp.com*) lub
\# **gpg --export naszAdres@domena.pl > nazwaPliku** (tworzymy blik binarny z kluczem w celu jego rozesłania do znajomych) lub alternatywnie
\# **gpg --output nazwaPliku --export naszAdres@domena.pl** (klucz jako plik binarny alternatywnie) lub
\# **gpg --armour --output nazwaPliku --export naszAdres@domena.pl** (klucz jako plik tekstowy)

Importowanie klucza

\# **gpg --keyserver „hkp://subkeys.pgp.net" --search-keys „Imię Nazwisko"** (szukamy kluczy naszych znajomych na publicznych serwerach)
\# **gpg --import /home/user/klucz** (importujemy klucz naszych znajomych z pliku binarnego lub tekstowego)
Po zaimportowaniu klucza, sprawdzamy jego „*odcisk palca*" (*fingerprint*) i pytamy się znajomego (np. telefonicznie) czy zgadza się z jego odciskiem.
\# **gpg --fingerprint**
Jeśli zgadza się, podpisujemy jego klucz uznając go za prawdziwy:
\# **gpg --edit-key znajomy@domena.pl** (wydajemy polecenie „*sign*"; polecenie „trust" służy zaś do określenia zaufania do rzetelności samego nadawcy, a nie do jego klucza)

Szyfrowanie plików

\# **gpg --recipient odbiorca@domena.net --output plik.jpg.gpg --sign --encrypt plik.jpg** (szyfrujemy plik jako binarny i podpisujemy go wykorzystując klucz prywatny; opcja *--armour*

stworzyłaby plik tekstowy, który można wkleić jako tekst do treści e-mail)
gpg --output plik.gpg --symmetric plik.txt (szyfrowanie symetryczne, czyli tylko na hasło, bez udziału kluczy)

Deszyfracja plików
gpg --output plik.jpg --decrypt plik.gpg

Ostatnia aktualizacja: 4 kwietnia 2014.