Universitatea Politehnică București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Aplicaţii în reţele

Preety Good Privacy (PGP) şi GNU Privacy Guard(GnuPG)

-2016-

Conducător,
Conf. Dr. Ing. Ştefan Stăncescu

Masterand,
Valentin Florin Benciu
Anul II, Master IISC

Contents

0.Creatorul PGP-ului 3

1.Structura cheilor 3

1.1. Chei primare și subchei 6

1.2. Utilizatori și identități 7

1.3. Generarea și modificarea cheilor 7

2. Transmiterea cheilor publice 7

2.1. Transmiterea cheilor 7

2.2. Verificarea autenticității cheilor 8

2.3. Gestiunea cheilor secrete 9

3. Mesajele criptate/semnate și transmiterea acestora 9

5.Concluzii 11

5. Bibliografie 11

6. Anexa 12

0.Creatorul PGP-ului[1]

Biografie

Philip R. Zimmermann este creatorul produsului Pretty Good Privacy(PGP), un pachet de software pentru criptarea traficului de e-mail. Iniţial software-ul a fost conceput că un instrument de protecţie a drepturilor umane, dar în cele din urmă PGP a fost publicat gratuit pe internet în 1991. Zimmermann a devenit subiectul unei investigaţii penale timp de trei ani, deoarece guvernul a susţinut că restricţiile de export ale Statelor Unite referitoare la software-ul de criptare au fost încălcate în momentul în care PGP s-a răspândit în întreaga lume. În ciudă lipsei de fonduri, lipsei de personal plătit, lipsei unei companii, şi în ciudă persecuţie guvernului, PGP a devenit cel mai utilizat software de criptare a traficului de e-mail din lume. După 1996, Zimmermann a pus bazele fundamentale al structurii PGP Inc., companie achiziţionată de către Network Associates Inc (NAI), în 1997. În 2002 PGP a fost preluată de la NAI de o nouă companie numită PGP Corporation, unde Zimmermann a avut funcţia de consilier special şi consultant,până la achiziţionarea acesteia de către Symantec în 2010. În prezent, Zimmermann este consultant la mai multe companii şi organizaţii din industrie, pe teme criptografice, şi este, de asemenea Membru la Facultatea de Drept de la Stanford, Centrul pentru Internet şi Societate. A fost unul dintre designerii principali a algoritmului criptografic de protocol pentru ştandardul Wireless USB. Din 2004, eforturile sale s-au focalizat pe criptarea convorbirilor telefonice pe Internet, dezvoltând protocolul Zimmermann Real-time Transport Protocol (ZRTP) şi creând produse care să-l utilizeze, printre care şi Zfone. Noua iniţiativă a lui Zimmermann este Silent Circle, un furnizor de servicii de comunicare securizată.

Înainte de a fonda PGP Inc, Zimmermann a fost inginer de software, cu peste 20 de ani de experienţă, specializat în criptografie şi securitatea datelor, comunicaţii de date şi sisteme incorporate în timp real. Interesul său în domeniul politic a criptografiei se reflectă din trecutul său în probleme de politică militară .

Datorită spiritului său de inventator în domeniul criptografiei, Zimmermann a fost recunoscut prin nenumărate premii tehnice şi umanitare.În 2012 Internet Society l-au inclus în Internet Hall of Fame Internet Hall of Fame.

Zimmermann a absolvit Facultatea de Informatică în cadrul Universităţii Florida Atlantic, în 1978. Este membru al Asociaţiei Internaţionale de Cercetare Criptografică, a Asociaţiei Computing Machinery, şi al Ligii pentru Programare Libertate. A făcut parte din Roundtable on Scientific Communication and National Security, un proiect de colaborare între The National Research Council şi The Center for Strategic and International Studies. A facut parte din Comitetul de Directori pentru Computer Professionals for Social Responsibility, si in prezent face parte din Comitetul Consultativ de la Universitatea Santa Clara Departamentul Computer Engineering, Hush Communications, și Debix.

1.Structura cheilor

Statele Unite au o lege (22 U.S.C. 2778) care interzice cetăţenilor să exporte muniţii (material de război), cum ar fi tancuri şi avioane cu reacţie, fără autorizarea Department of Defence (DoD). Prin această lege, software-ul criptografic este clasificat ca muniţie. Phil Zimmermann, prezentat în capitolul 0, a fost acuzat că a încălcat această lege, chiar dacă

[1] https://www.philzimmermann.com/EN/letters/index.html

guvernul admite că el nu 1-a exportat (dar 1-a dat unui prieten pentru a-1 pune pe Internet de unde putea fi obţinut de străini). Mulţi oameni au privit acest incident larg mediatizat ca o încălcare totală a drepturilor unui cetăţean american ce lucrează pentru îmbunătăţirea siguranţei oamenilor.

PGP-Pretiy Good Privacy (trad. confidenţialitate destui de bună)

Preety Good Privacy (PGP) este un program pentru criptarea şi semnarea digital a mesajelor de poşta electronică şi a fişierelor în general. Gnu Privacy Guard, abreviat GPG sau GnuPG, este o reimplementarea PGP, ca soft liber.

Primul nostru exemplu, PGP (Pretty Good Privacy) este în cea mai mare parte rodul gândiriiunei singure persoane, Phil Zimmerman (Zimmerman, 1995a, 1995b). Este un pachet complet de securitate a e-mailului, care furnizează confidenţialitate, autentificare, semnături digitale şicompresie, toate într-o formă uşor de utilizat. Pe deasupra, pachetul complet, incluzând toate" sursele, este distribuit gratuit prin Internet, sisteme de informare în reţea şi reţele comerciale. Datorită calităţii sale, preţului (zero) şi simplei disponibilităţi pe platformele MS-DOS/Windows, UNIX şi Macintosh, acesta este larg utilizat astăzi. Pentru acele companii care necesită sprijin, este disponibilă şi o variantă comercială.

RSA, IDEA şi MD5 sunt principalii algoritmi care au fost supuşi la analize amănunţite şi nu au fost controlaţi de agenţii guvernamentale pentru a-i modifica slăbindu-le proprietăţile de criptare.

PGP permite compresie de: text, mesaje şi semnături digitale şi furnizează de asemenea facilităţi de management extensiv al cheilor. Pentru a vedea cum lucrează PGP, să considerăm exemplul din figura 1. Aici, Alice vrea să-i trimită lui Bob, într-o manieră sigură, un mesaj text simplu, semnat, P. Alice şi Bob au cheile private *(Dx)* RSA publice *(Ex)*. Să presupunem că târziu de administrarea cheilor.

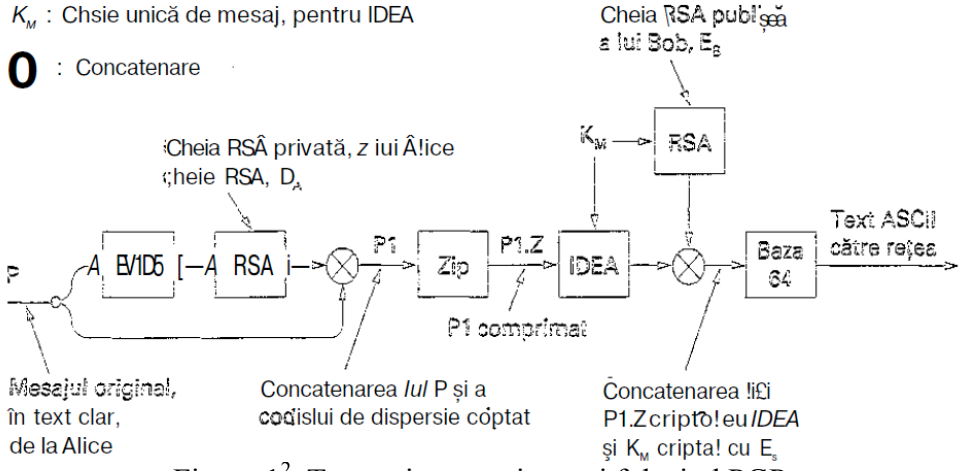


Figura 1[2]. Transmiterea uni mesaj folosind PGP

Alice:

1. invocă PGP pe calculatorul ei. PGP codifică prin dispersie (hash) mesajul P folosind MD5

2. criptează codul de dispersie rezultat folosind cheia sa RSA privată, D_A.

3. Când, la un moment dat, Bob primeşte mesajul, el poate decripta mesajul cu cheia

[2] Radu-Lucian Lupşa, Retele de calculatoare,2008

publică cunoscută a lui Alice şi poate testa corectitudinea mesajului codificat prin dispersie. Observaţie: Chiar dacă altcineva (de ex. Max) poate obţine codul în acest stadiu şi S poate decripta cu cheia publică a lui Alice, prin MD5 este nerealizabilă computaţional producerea unui alt mesaj care să aibă acelaşi cod MD5.

Codul de dispersie criptat şi mesajul original sunt acum concatenate într-un singur mesaj *PI* şi comprimate apoi cu programul ZIP, care utilizează algoritmul Ziv-Lempel (Ziv şi Lempel, 1977).

Numim ieşirea obţinută la acest pas *Pl.Z.* Apoi, PGP îi cere lui Alice introducerea unui şir de caractere oarecare. Atât conţinutul acestuia cât şi viteza de tastare sunt utilizate pentru a genera o cheie de mesaj de tip IDEA, de 128 de biţi, *KM* (numită cheie de sesiune în literatura PGP, dar numele este cu adevărat nepotrivit atâta timp cât nu există nici o sesiune). *KM* este acum utilizat pentru a cripta *Pl.Z* cu IDEA, prin metoda de tip reactive cifrată. în plus, *KM* este criptată cu cheia publică a lui Bob, *EB*- Aceste două componente sunt apoi concatenate şi convertite în bază 64, aşa cum s-a discutat în secţiunea despre MIME. Mesajul rezultat conţine numai litere, cifre şi simbolurile +, / şi =, ceea ce înseamnă că poate fi pus într-un corp de RFC 822 şi că ne putem aştepta să ajungă nemodificat la destinaţie.

Bob:

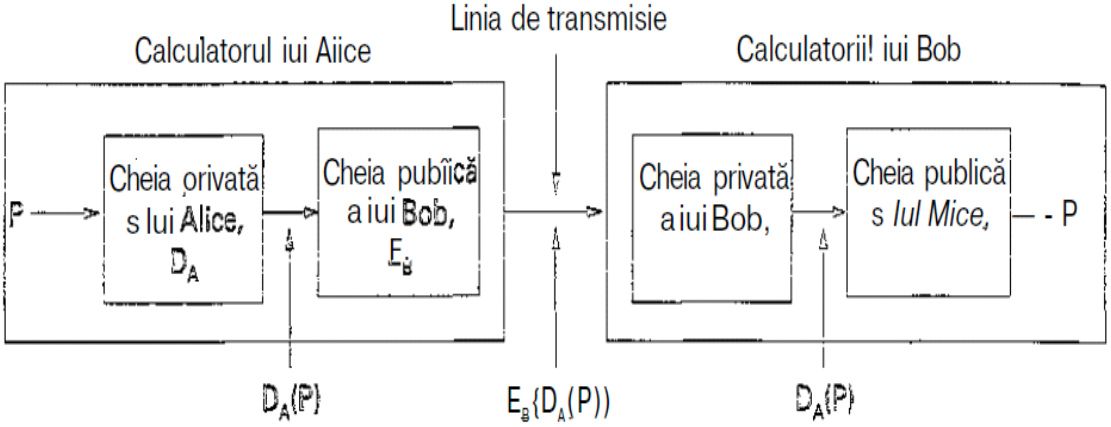
1. primeşte mesajul, îl reconverteşte din bază 64 şi decriptează cheia IDEA (**International Data Encryption Algorithm**) utilizând cheia sa RSA privată. Utilizând această cheie, decriptează mesajul pentru a obţine *Pl.Z.*

2. după decompresia acestuia, Bob separă textul simplu de codul cifrat şi decriptează codul de dispersie utilizând cheia publică a lui Alice.

3. dacă codul textului clar cade de acord cu calculul făcut de ei utilizând MD5, el ştie că P este mesajul corect şi că vine de la Alice.

RSA este utilizat în două locuri aici: pentru a cifra codul de dispersie de 128 de biţi generat de MD5 şi pentru a cifra cheia IDEA de 128 de biţi. Deşi RSA este lent, are de criptat doar 256 de biţi şi nu un volum mare de date. Mai mult, toţi cei 256 de biţi de text simplu sunt generaţi extrem de aleatoriu, astfel încât numai pentru a determina dacă o cheie ghicită este corectă, Trudy ar trebui să depună o cantitate însemnată de muncă.

Criptarea de mare putere este realizată de IDEA, care este cu câteva ordine de mărime mai rapidă decât RSA. Astfel, PGP asigură securitate, compresie şi o semnătură digitală şi face [3]acest lucru într-o manieră chiar mult mai eficientă decât schema ilustrată în figura 2.



[3] Radu-Lucian Lupşa, Retele de calculatoare,2008

Figura 2[4]. Semnaturi digitale folosind criptografia cu cheie publica

PGP acceptă trei lungimi de chei RSA. Lungimile disponibile sunt:

1. Obişnuită (384 biţi): poate fi spartă în ziua de azi de către cei cu bugete mari.
2. Comercială (512 biţi): ar putea fi spartă de organizaţii cu nume din trei litere (care se ocupă cu securitatea statului).
3. Militară (1024): Nici un pământean nu o poate sparge.

Formatul unui mesaj PGP este prezentat în figura 3. Mesajul are trei părţi, conţinând cheia IDEA, semnătura şi respectiv mesajul. Partea care conţine cheia mai include, de asemenea, un identificator de cheie, deoarece utilizatorilor li se permite să aibă mai multe chei publice.

Partea cu semnătura conţine un antet, de care nu ne vom ocupa aici. Antetul este urmat de o amprentă de timp, identificatorul pentru cheia publică a emiţătorului, care este folosit pentru a decripta codul semnăturii, unele informaţii care identifică algoritmul folosit (pentru a permite folosirea algoritmilor MD6 şi RSA2 atunci când aceştia vor fi inventaţi) precum şi codul de dispersie criptat.

Partea de mesaj conţine de asemenea un antet, numele implicit care va fi folosit pentru fişier în cazul în care utilizatorul doreşte să-1 salveze pe disc, o amprentă de timp pusă la crearea mesajului şi, în sfârşit, mesajul în sine.

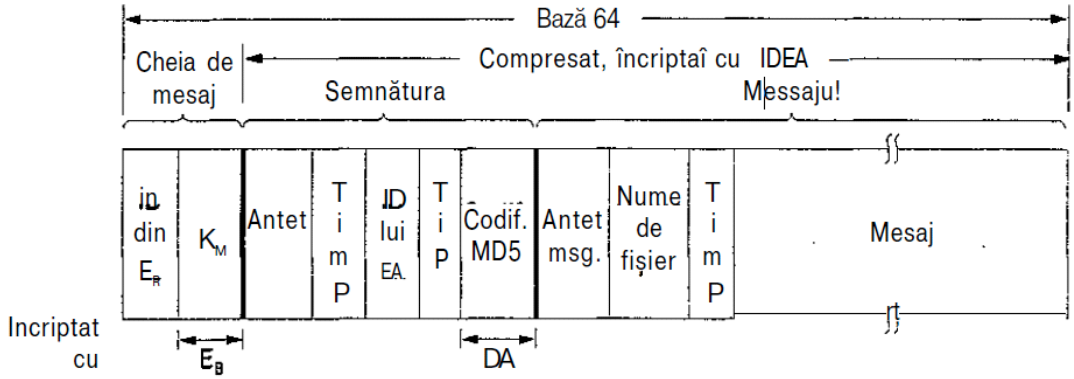


Figura 3[5]. Mesaj PGP

1.1. Chei primare şi subchei

Cheile GnuPG sunt de două tipuri: *chei primare* şi *subchei*. O cheie primară (care este formată defapt dintr-o pereche primară de chei) este o pereche de chei pentru semnătură digitală. O subcheie (care este formată defapt dintr-o sub-pereche de chei) este subordonată unei anumite perechi primare. Fiecare subcheie poate fi cheie de criptare sau cheie de semnătură.

Utilizatorul are o cheie primară şi, subordonate acesteia, zero sau mai multe subchei. În modul cel mai simplu de lucru, fiecare utilizator GnuPG are asociate, două perechi de chei: o pereche primară de chei pentru semnătură digitală şi o sub-pereche de chei pentru criptare asimetrică. Perechea de chei de criptare este folosită pentru a trimite mesaje secrete posesorului perechii de chei. Perechea de chei de semnătură este folosită atunci când posesorul trimite mesaje semnate.

[4] Radu-Lucian Lupşa, Retele de calculatoare,2008
[5] Radu-Lucian Lupşa, Retele de calculatoare,2008

Amprentă a cheii (engl. *key fingerprint*) = este un şir de biţi, calculaţi, printr-o funcţie de dispersie criptografică, din cheia publică respectivă.

Pentru a ne putea referi la o pereche de chei, fiecare pereche de chei (primară sau subcheie) are asociate doi *identificatori de cheie* (engl. *key ID*):

- *Identificatorul lung* conţine 16 cifre hexa. Şansele ca două chei distincte să aibă acelaşi identificator lung sunt extrem de mici, astfel încât identificatorul lung este suficient pentru a identifica unic orice cheie.
- *Identificatorul scurt* conţine 8 cifre hexa ale identificatorului lung. Dacă, într-un anumit context, nu există două chei cu acelaşi identificator scurt, el poate fi folosit pentru a ne referi la o cheie.

1.2. Utilizatori şi identităţi

Una sau mai multe *identităţi* îi sunt associate fiecărei chei primare. Identitatea este un nume complet de utilizator format din: 1.*numele real* (numele şi prenumele persoanei), 2.*adresa de poştă electronică* şi 3.*comentariu*, a treia componenta poate fi optională. Numele complet, adresa de poştă electronică se scrie între semne *mai mic* şi *mai mare*, iar comentariul se scrie între paranteze. Exemplu: Florin Benciu <benciuflorinvalentin@gmail.com>

Este posibil ca o cheie primară să aibă asociate mai multe identităţi.
Acest lucru este util dacă un utilizator are mai multe adrese de poştă electronică şi doreşte asocierea tuturor acestora cu aceeaşi cheie.

Reciproc, un acelaşi nume complet poate _ asociat mai multor chei primare. Acest lucru se întâmplă deoarece nu poate nimeni să împiedice doi utilizatori să genereze două chei şi să le asocieze acelaşi nume complet.

Această posibilitate este utilizată frecvent în situaţia în care cheia primară a unui utilizator expiră sau este revocată. În această situaţie, utilizatorul poate crea o nouă cheie primară căreia să-i asocieze acelaşi nume complet.

1.3. Generarea şi modificarea cheilor

Generarea unei chei primare se face cu comanda

gpg --gen-key

Comanda solicită utilizatorului următoarele informaţii: tipul cheilor, dimensiunea, durata de valabilitate a cheilor, numele complet al utilizatorului şi parola pentru criptarea cheii secrete.

La generarea cheilor, GnuPG afişează o amprentă a cheii primare generate. Este de preferat ca utilizatorul să noteze amprenta cheii generate. Acest lucru este util la transmiterea cheii publice către partenerii de comunicaţie.

Pentru o cheie primară dată, proprietarul ei poate crea (şi, eventual, şterge) subchei. Pentru acestea, se lansează comanda:

gpg --edit-key *cheie*

La lansarea comenzii de mai sus, GPG aşteaptă de la utilizatori subcomenzi pentru modificarea unor date privitoare la cheia primară identificată prin parametrul *cheie*. Ditribuirea seriei de subcomenzi se face dând, mai întâi, subcomanda **save** pentru a memora efectiv modificările efectuate, urmată de subcomanda **quit**.

2. Transmiterea cheilor publice

2.1. Transmiterea cheilor

În fișierul gestionat de GnuPG sunt memorate urmatoarele informații: cheile publice primare, identitățile asociate, semnăturile asupra identităților, subcheile publice subordonate cheilor primare și certificatele de revocare ale cheilor primare sau subcheilor.

Transmiterea acestor obiecte de la un utilizator la altul se face prin două metode:

1. prin fișiere (transmise, de exemplu, prin poștă electronică sau prin web);
2. prin *servere de chei*.

1. La transmiterea prin fișiere, utilizatorul A exportă una sau mai multe chei primare, împreună cu identitățile, semnăturile, subcheile și certificatele de revocare asociate acelor chei primare, într-un fișier. Utilizatorul B primește fișierul (transmis prin poștă electronică, web, pe o dischetă sau prin alte mijloace) și îi importă conținutul în GnuPG-ul local.

Exportul unei chei publice primare, împreună cu toate identitățile, subcheile publice, semnăturile și certificatele de revocare asociate, se face prin comanda:

gpg -a -o *fișier* **--export** *cheie*

unde parametrul *cheie* este identificatorul cheii sau a uneia dintre subchei sau numele utilizatorului căreia îi aparține, iar parametrul *fișier* reprezintă fișierul în care se vor scrie datele.

Importarea unei chei dintr-un fișier se face prin comanda:

gpg --import *fișier*

Cheile și obiectele din fișierul importat efectuând operația de import adăugămla cele locale sau acest lucru duce la modificarea acestora. Ștergerea unui tip de cheie indifferent dacă aceasta este primară, subcheie, identitate sau semnătură, nu poate fi transmisă asupra partenerilor de comunicație. Invalidarea orcărei tip de cheie descrise în această frază se poate face doar prin revocarea acesteia și apoi transmiterea certificatului de revocare.

2. La transmiterea prin servere de chei, primul utilizator încarcă, pe un *server de chei*, cheile și celelalte obiecte de transmis, iar celălalt utilizator le descarcă de pe serverul de chei.

Transmiterea unei chei primare și a obiectelor asociate către un server de chei se face prin comanda:

gpg --keyserver *server* --send-key *cheie*

Descărcarea unei chei și a obiectelor asociate de pe un server de chei se face prin comanda:

gpg --keyserver *server* **--recv-key** *cheie*,

unde *cheie* este identificatorul unei chei (nu poate _ numele posesorului cheii). Aarea identificatorului cheii unui utilizator se poate face prin comanda:

gpg --keyserver *server* **--search-key** *nume-utilizator*

2.2. Verificarea autenticității cheilor

GnuPG verifică automat, înainte de utilizarea unei chei publice, autenticitatea acesteia. Autenticitatea cheilor se verifică cu ajutorul certificatelor. În terminologia GnuPG, un certificat este numit *semnătură*
asupra unei chei.

O sub-cheie este în mod normal semnată cu cheia primară căreia îi este subordonată. O sub-cheie a cărei semnătură este validă este considerate autentică dacă și numai dacă cheia primară

coresunzătoare este considerate autentică. În consecință, dacă se importă noi sub-chei pentru o cheie primară declarată autentică, sub-cheile respective sunt imediat considerate autentice.

Restul paragrafului de față tratează doar cheile primare. Reamintim că o cheie primară este întotdeauna o cheie de semnătură.

O cheie publică pentru care GnuPG dispune de cheia secretă corespunzătoare este automat considețrată autentică. De asemenea, este considerată autentică orice cheie specificată printr-o opțiune:

--trusted-key *cheie*

fie la execuția comenzii gpg, fie în fișierul cu opțiunile implicite.

esențial, pentru securitatea sistemului, ca un utilizator să nu semneze un set de chei fără să-i verifice maiîntâi autenticitatea. Autenticitatea setului de chei se poate asigura în două moduri:

- Fișierul din care se importă setul de chei este adus pe o cale sigură, de exemplu printr-o dischetă dată personal de către proprietarul cheii sau este descărcat de pe un sait web securizat (*https*) și de încredere.
- Întâi, amprenta cheii primare este transmisă pe o cale sigură, de exem plu pe un bilet scris de către proprietarul cheii sau printr-o convorbire telefonică cu proprietarul cheii. Apoi, la importarea și semnarea setului de chei, utilizatorul verifică amprenta cheii primare din set.

2.3. Gestiunea cheilor secrete

GnuPG plasează cheile secrete într-un fișier gestionat de GnuPG. Acest fișier este creat cu drepturi de citire (în sistemul de operare) doar pentru utilizatorul curent.

Cheile sunt, în mod normal, criptate cu o parolă dată de utilizator. Parola de criptare poate fi schimbată cu comandaȘ

gpg --edit-key *cheie*

subcomanda **passwd**.

Cheile secrete pot fi exportate, prin comanda

gpg -a -o *fișier* **--export-secret-keys** *cheie*

Această comandă exportă cheia secretă primară identificată prin parametrul *cheie*, precum și subcheile sale secrete. Cheile secrete pot _ importate prin comanda:

gpg --import *fișier*

Acest lucru este util dacă utilizatorul lucrează pe mai multe calculatoare și dorește să utilizeze aceeași chei pe mai multe calculatoare. Cheia secretă este exportată în forma criptată.

Există posibilitatea de-a exporta doar subcheile secrete ale unei chei primare. Acest lucru se face prin comanda:

gpg -a -o *fișier* **--export-secret-subkeys** *cheie*

Cu ajutorul acestei comenzi, un utilizator poate ține cheia primară secretă pe un calculator sigur și poate transmite subcheile secrete către un calculator mai puțin sigur pe care îl utilizează frecvent. În acest mod, el poate utiliza calculatorul mai nesigur pentru transmite mesaje semnate și primi mesaje criptate utilizând subcheile, fără însă a risca compromiterea cheii primare în cazul în care cineva ar sparge acel calculator. Pentru o astfel de utilizare, subcheile se crează cu durată de valabilitate scurtă și se revocă la nevoie. Cheia primară este bine să aibă durată lungă de utilizare pentru a beneficia de semnăturile obținute de la alți utilizatori asupra ei.

3. Mesajele criptate/semnate și transmiterea acestora

Crearea unui mesaj criptat și semnat se face astfel:

gpg -o *fișieșire* **-r** *cheie-dest* **-se** *fiș-intrare*

sau

gpg -a -o *fiș-ieșire* **-r** *cheie-dest* **-se** *fiș-intrare*

unde *fiș-intrare* este fișierul ce trebuie semnat și criptat, *fiș-ieșire* este fișierul în care comanda gpg va pune datele criptate și semnate, iar *cheie-dest* reprezintă numele utilizatorului destinație sau identificatorul cheii de criptare de utilizate.

Cea de-a doua variantă produce un fișier text ASCII, prin recodificare în baza 64.

La criptarea unui mesaj, GnuPG generează aleator o cheie efemeră, criptează textul clar utilizând cheia efemeră, iar apoi criptează cheia efemeră utilizând cheia publică a destinatarului. Dacă sunt mai mulți destinatari, GnuPG criptează, pentru fiecare destinatar, câte o copie a cheii efemere, utilizând cheia publică a acelui destinatar.

Decriptarea unui mesaj se face cu, comanda:

gpg -o *fiș-ieșire* **--decrypt** *fiș-intrare*

unde *fiș-intrare* este fișierul semnat și criptat, iar *fiș-ieșire* este fișierul în care comanda gpg va pune rezultatul decriptării. Comanda verifică și semnătura și afișează pe ecran rezultatul verificării.

Se pot genera mesaje numai criptate sau numai semnate.

Pentru generarea unui mesaj semnat dar necriptat există trei posibilități: semnătură inclusă în mesaj, semnătură detașată și semnătură în clar.

Semnătura inclusă se generează similar cu generarea unui mesaj criptat și semnat, dar lipsesc destinatarii (opțiunile -r) și în loc de -se se dă doar -s. Fișierul generat conține datele originale și semnătura. Extragerea datelor și verificarea semnăturii se face exact ca în cazul unui mesaj criptat și semnat, adică prin comanda:

gpg -o *fiș-ieșire* **--decrypt** *fiș-intrare*

Semnătura detașată se generează prin comanda

gpg -a -o *fiș-sign* **--detach-sign** *fiș-date*

Rezultatul comenzii este scrierea în fișierul *fiș-sign* a semnăturii conținutului fișierului *fiș-date*. Fișierul produs, *fiș-sign*, este mic și conține doar semnătura; datele utile nu pot _ recuperate din el. Verificarea semnăturii se face prin comanda:

gpg --verify *fiș-sign fiș-date*

Semnătura detașată este utilă deoarece păstrează intact fișierul de date, nefiind nevoie de gpg pentru recuperarea datelor. De asemenea, permite mai multor utilizatori să semneze un același fișier de date.

În fine, semnătura în clar se poate utiliza doar dacă datele sunt text ASCII. Semnătura se generează prin comanda:

gpg -o *fiș-ieșire* **--clearsing** *fiș-intrare*

Fișierul astfel produs este un fișier text, care poate fi citit ușor de către utilizatorul uman. Textul original este pus între niște marcaje, iar semnătura este adăugată la sfârșit. Verificarea semnăturii se face prin comanda:

gpg --verify *fiș*

Semnătura în clar este utilă pentru semnarea documentelor text.

Acestea rămân ușor de citit de către om și, spre deosebire de semnătura detașată, datele utile și semnătura sunt puse într-un singur fișier.

Dacă GnuPG are mai multe chei secrete utilizabile pentru semnătură, se poate specifica ce cheie trebuie utilizată pentru crearea semnăturii. Specificarea cheii se face adăugând opțiunea

-u *cheie*

GnuPG se utilizează curent pentru autentificarea softului liber. În acest scop, alături de programul distribuit, se distribuie un fișier ce conține semnătura detașată a fișierului ce conține programul.

3. Concluzii

PGP satisface trei cerinţe esenţiale:

1. Caracterul privat al poştei electronice, ceea ce înseamnă că doar destinatarul desemnat al scrisorii poate citi conţinutul acesteia.
2. Autentificarea emiţătorului mesajului.
3. Autentificarea mesajelor - adică certitudinea că mesajele nu au fost modificate de alte persoane.

Agentiile inteligente au acces la tehnologii de criptare foarte avansate. Și la fel și cei care fac trafic de arme și trafic de droguri. Și la fel și companiile petroliere și alte corporatii gigantice. Dar oamenii de rand și organizațiile politice, majoritarea nu au acces la tehnologii de criptare de nivel militar. Pană acum.

PGP oferă puterea oamenilor, oferă dreptul mai bine spus, să faca ce doresc ei cu partea privata a mesajelor și informatiilor lor. Este o nevoie în crestere.(traducere dupa [6])

5. Bibliografie

1. https://rstforums.com/forum/31086-un-tutorial-foarte-simplu-despre-gpg.rst
2. https://www.gnupg.org/documentation/index.html
3. Retele de calculatoare editia a 5-a, Andrew S. Tanenbaum, 1998
4. V.V.Patriciu, Ene Pietroşanu, I.Bică, A.Cristea, Securitatea informaticăîn UNIX şi INTERNET
5. RFC 821 - SIMPLE MAIL TRANSFER PROTOCOL - Jonathan B. Postel – Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, California 90291, August 1982
6. Radu-Lucian Lupșa, Retele de calculatoare, Editura Casa Cartii de Stiinta, 2008

[6] http://www.pgpi.org/doc/whypgp/en/

6. Anexa

Scrisoare din partea creatorului PGP
Why do you need PGP?

by Phil Zimmermann

It's personal. It's private. And it's no one's business but yours. You may be planning a political campaign, discussing your taxes, or having an illicit affair. Or you may be doing something that you feel shouldn't be illegal, but is. Whatever it is, you don't want your private electronic mail (E-mail) or confidential documents read by anyone else. There's nothing wrong with asserting your privacy. Privacy is as apple-pie as the Constitution.

Perhaps you think your E-mail is legitimate enough that encryption is unwarranted. If you really are a law-abiding citizen with nothing to hide, then why don't you always send your paper mail on postcards? Why not submit to drug testing on demand? Why require a warrant for police searches of your house? Are you trying to hide something? You must be a subversive or a drug dealer if you hide your mail inside envelopes. Or maybe a paranoid nut. Do law-abiding citizens have any need to encrypt their E-mail?

What if everyone believed that law-abiding citizens should use postcards for their mail? If some brave soul tried to assert his privacy by using an envelope for his mail, it would draw suspicion. Perhaps the authorities would open his mail to see what he's hiding. Fortunately, we don't live in that kind of world, because everyone protects most of their mail with envelopes. So no one draws suspicion by asserting their privacy with an envelope. There's safety in numbers. Analogously, it would be nice if everyone routinely used encryption for all their E-mail, innocent or not, so that no one drew suspicion by asserting their E-mail privacy with encryption. Think of it as a form of solidarity.

Today, if the Government wants to violate the privacy of ordinary citizens, it has to expend a certain amount of expense and labor to intercept and steam open and read paper mail, and listen to and possibly transcribe spoken telephone conversation. This kind of labor-intensive monitoring is not practical on a large scale. This is only done in important cases when it seems worthwhile.

More and more of our private communications are being routed through electronic channels. Electronic mail is gradually replacing conventional paper mail. E-mail messages are just too easy to intercept and scan for interesting keywords. This can be done easily, routinely, automatically, and undetectably on a grand scale. International cablegrams are already scanned this way on a large scale by the NSA.

We are moving toward a future when the nation will be crisscrossed with high capacity fiber optic data networks linking together all our increasingly ubiquitous personal computers. E-

mail will be the norm for everyone, not the novelty it is today. The Government will protect our E-mail with Government-designed encryption protocols. Probably most people will acquiesce to that. But perhaps some people will prefer their own protective measures.

Senate Bill 266, a 1991 omnibus anti-crime bill, had an unsettling measure buried in it. If this non-binding resolution had become real law, it would have forced manufacturers of secure communications equipment to insert special trap doors in their products, so that the Government can read anyone's encrypted messages. It reads:

"*It is the sense of Congress that providers of electronic communications services and manufacturers of electronic communications service equipment shall insure that communications systems permit the Government to obtain the plain text contents of voice, data, and other communications when appropriately authorized by law*."

This measure was defeated after rigorous protest from civil libertarians and industry groups.

In 1992, the FBI Digital Telephony wiretap proposal was introduced to Congress. It would require all manufacturers of communications equipment to build in special remote wiretap ports that would enable the FBI to remotely wiretap all forms of electronic communication from FBI offices. Although it never attracted any sponsors in Congress in 1992 because of citizen opposition, it was reintroduced in 1994.

Most alarming of all is the White House's bold new encryption policy initiative, under development at NSA since the start of the Bush administration, and unveiled April 16th, 1993. The centerpiece of this initiative is a Government-built encryption device, called the Clipper chip, containing a new classified NSA encryption algorithm. The Government is encouraging private industry to design it into all their secure communication products, like secure phones, secure FAX, etc. AT&T is now putting the Clipper into their secure voice products. The catch: At the time of manufacture, each Clipper chip will be loaded with its own unique key, and the Government gets to keep a copy, placed in escrow. Not to worry, though -- the Government promises that they will use these keys to read your traffic only when duly authorized by law. Of course, to make Clipper completely effective, the next logical step would be to outlaw other forms of cryptography.

If privacy is outlawed, only outlaws will have privacy. Intelligence agencies have access to good cryptographic technology. So do the big arms and drug traffickers. So do defense contractors, oil companies, and other corporate giants. But ordinary people and grassroots political organizations mostly have not had access to affordable military grade public-key cryptographic technology. Until now.

PGP empowers people to take their privacy into their own hands. There's a growing social need for it. That's why I wrote it.