El software criptográfico GnuPG

Jacobo Tarrío Barreiro
<jtarrio@gpul.org>

¿Quién necesita criptografía?

- Gobiernos
- Delincuentes
- El común de los mortales
  - Banca/comercio online
  - Mensajes privados
  - Datos sensibles
  - ...

Historia de GnuPG

- Basado en PGP (de Philip Zimmermann, 1990)
- Zimmermann tiene problemas con el gobierno USA
  - Artefactos criptográficos clasificados como armas
    - Exportación limitada
  - Se "exportan" copias de PGP
- La solución pasa por vender PGP a una empresa
- Se crea GnuPG (1999) para tener un programa criptográfico libre

Criptografía de clave privada

- Utiliza la misma clave para cifrar y descifrar
- La forma más antigua y conocida
- Problema del intercambio de claves
  - Alicia, Benito y Eva
  - Alicia y Benito se quieren comunicar la clave sin conocimiento de Eva
  - Si tienen un canal seguro, ¿por qué no usarlo para los mensajes?

Criptografía de clave pública (I)

- Algoritmo de intercambio de claves Diffie-Hellman
  - Alicia y Benito intercambian datos a la vista de Eva
  - Finalmente, ellos obtienen el mismo número (clave); Eva no
  - Inventado independientemente en el GCHQ

Criptografía de clave pública (II)

- Algoritmo RSA (Rivest-Shamir-Adleman)
  - Alicia genera un par de claves; una la publica y se queda con la otra
  - Benito cifra un mensaje con la clave pública de Alicia
  - Sólo se puede descifrar con la clave secreta de Alicia
  - Eva conoce el mensaje cifrado y la clave pública, pero no puede descifrar

Generación de claves

- Generación de un par:
  - gpg --gen-key
- Generación de un certificado de revocación:
  - gpg --gen-revoke UID
  - Útil por si se pierde la clave secreta o la contraseña de la clave

Cifrado de mensajes

- Cifrado:
  - gpg --encrypt FICHERO
- Cifrado con armadura ASCII:
  - gpg --encrypt --armor FICHERO
- Descifrado (salida por pantalla):
  - gpg --decrypt FICHERO
- Descifrado (salida en fichero):
  - gpg FICHERO

Importación/exportación de claves

- Importar una clave pública:
  - gpg --import FICHERO
- Exportar una clave pública:
  - gpg --export [--armor] UID > FICHERO
- Listar claves disponibles:
  - gpg --list-keys [UID]
- Recibir una clave de un servidor de claves:
  - gpg --keyserver SERVIDOR --recv-keys UID
- Enviar una clave a un servidor de claves:
  - gpg --keyserver SERVIDOR --send-keys UID
- Buscar una clave en un servidor:
  - gpg --keyserver SERVIDOR --search-keys UID

Firma de mensajes (I)

- Función hash
  - De un texto obtiene un número
  - Difícil conseguir dos textos con el mismo hash
  - Los más usados: MD5 y SHA
- Uso de un hash
  - Alicia envía un mensaje a Benito, y su hash
  - Benito calcula el hash del mensaje recibido y lo compara con el que Alicia le envió
  - Si son iguales, el texto es el mismo
  - Problema: Eva puede calcular el hash del texto modificado
  - Solución en la siguiente pantalla

Firma de mensajes (II)

- La solución
  - Criptografía de clave pública, pero al revés
  - Alicia cifra el hash con su clave secreta, y Benito lo descifra con la pública de Alicia
  - Si Eva modificó el mensaje, no puede cifrar el nuevo hash
  - Ahora Benito sabe que el hash que recibe de Alicia es real
  - Si coinciden, el mensaje no ha cambiado y sólo Alicia pudo enviarlo

Firma de mensajes (III)

- Firma de un texto:
  - gpg --sign [--armor] FICHERO
- Firma de texto claro:
  - gpg --clearsign [--armor] FICHERO
- Firma en un fichero separado:
  - gpg --detach-sign [--armor] FICHERO
- Comprobación de una firma:
  - gpg FICHERO

Validez de una clave pública

- Alicia publica su clave
- Eva interviene las comunicaciones de Benito
- Benito intenta descargar la clave de Alicia
- Eva proporciona a Benito una clave suya haci éndola pasar por una de Alicia
- Ataque del "hombre interpuesto"
- Solución: que alguien de confianza firme las claves buenas
- Aproximaciones: autoridades certificadoras y red de confianza

La red de confianza

- Alicia y Carlos se ven en persona
- Carlos firma la clave de Alicia
- Carlos sabe que la clave de Alicia es "buena" si su firma es correcta (Eva no puede falsificarla)
- Si Benito confía en las firmas de Carlos, puede confiar en la clave de Alicia
- El software GnuPG gestiona los niveles de confianza en las firmas

Firma de claves

- Obtener "fingerprint":
  - gpg --fingerprint UID
- (Comparar fingerprint)
- Firmar clave:
  - gpg --sign-key UID
- (Exportar clave o enviar a servidor de claves)
- Ver firmas de una o varias claves:
  - gpg --list-sigs [UID]
- Editar confianza:
  - gpg --edit-key UID
  - (comando trust)

Uso de GnuPG en Debian

- Voto en línea
  - Mensajes firmados y cifrados
- Autenticidad de paquetes
  - Ficheros firmados
- Avisos de seguridad
  - Mensajes firmados
- Red de firmas "Debian"
  - Excusa para ir a "botellones Linux" ;-)

Datos útiles

- Sitio web de GnuPG
  - http://www.gnupg.org/
- Compatible con OpenPGP
- Servidores de claves
  - pgpkeys.mit.edu
  - keyring.debian.org
- Mi ID de clave
  - 0x149cddb2
  - (para fingerprint, preguntar personalmente)