UNIVERSIDAD NACIONAL DE LA MATANZA

Departamento de Ingeniería e Investigaciones Tecnológicas

Seguridad y Calidad en Aplicaciones Web

Unidad N° 3: Anexo PGP

Fuente: "Criptografía y Seguridad en Computadoras", Manuel José Lucena López. Capítulo 18, PGP

Capítulo 18

PGP

El nombre PGP responde a las siglas *pretty good privacy* (privacidad bastante buena), y se trata de un proyecto iniciado a principios de los 90 por Phil Zimmermann. La total ausencia por aquel entonces de herramientas sencillas, potentes y baratas que acercaran la criptografía *seria* al usuario movió a su autor a desarrollar una aplicación que llenara este hueco.

Con el paso de los años, PGP se ha convertido en uno de los mecanismos más populares y fiables para mantener la seguridad y privacidad en las comunicaciones, especialmente a través del correo electrónico, tanto para pequeños usuarios como para grandes empresas.

Actualmente PGP se ha convertido en un estándar internacional (*RFC 2440*), lo cual está dando lugar a la aparición de múltiples productos PGP, que permiten desde cifrar correo electrónico hasta codificar particiones enteras del disco duro (PGPDisk), pasando por la codificación automática y transparente de todo el tráfico TCP/IP (PGPnet).

18.1. Fundamentos e Historia de PGP

PGP trabaja con criptografía asimétrica, y por ello tal vez su punto más fuerte sea precisamente la gran facilidad que ofrece al usuario a la hora de gestionar sus claves públicas y privadas. Si uno emplea algoritmos asimétricos, debe poseer las claves públicas de todos sus interlocutores, además de la clave privada propia. Con PGP surge el concepto de *anillo de claves* (o llavero), que no es ni más ni menos que el *lugar* que este programa proporciona para que el usuario guarde todas las claves que posee. El anillo de claves es un único fichero en el que se pueden efectuar operaciones

de extracción e inserción de claves de manera sencilla, y que además proporciona un mecanismo de identificación y autentificación de llaves completo y simple de utilizar. Esta facilidad en la gestión de claves es una de las causas fundamentales que han hecho a PGP tan popular.

La historia de PGP se remonta a comienzos de los años 90. La primera versión era completamente diferente a los PGP posteriores, además de ser incompatible con éstos. La familia de versiones 2.x.x fue la que alcanzó una mayor popularidad, y sigue siendo utilizada por mucha gente en la actualidad. Los PGP 2.x.x emplean únicamente los algoritmos IDEA, RSA y MD5.

En algún momento una versión de PGP atravesó las fronteras de EE.UU. y nació la primera versión internacional de PGP, denominada PGPi, lo que le supuso a Phil Zimmermann una investigación de más de tres años por parte del FBI, ya que supuestamente se habían violado las restrictivas leyes de exportación de material criptográfico que poseen los Estados Unidos. Para la versión 5 de PGP se subsanó este problema exportando una versión impresa del código fuente, que luego era reconstruida y compilada en Europa (más información en `http://www.pgpi.com`).

Hasta principios de 2001 la política de distribución de PGP consistió en permitir su uso gratuito para usos no comerciales y en publicar el código fuente en su integridad, con el objetivo de satisfacer a los desconfiados y a los curiosos. Sin embargo, con el abandono de la empresa por parte de Zimmermann, en febrero de 2001, el código fuente dejó de publicarse. En la actualidad (finales de 2004), la empresa que gestiona los productos PGP (PGP Corporation), ha vuelto a publicar el código fuente de los mismos.

Paralelamente a la azarosa existencia empresarial de PGP, el proyecto GNU ha estado desarrollando su propia aplicación de código abierto compatible con el *RFC 2440*, denominada GnuPG, y que solo emplea algoritmos libres de patentes.

18.2. Estructura de PGP

18.2.1. Codificación de Mensajes

Como el lector ya sabe, los algoritmos simétricos de cifrado son considerablemente más rápidos que los asimétricos. Por esta razón PGP cifra primero el mensaje empleando un algoritmo simétrico (ver figura 18.1) con una clave generada aleatoriamente (*clave de sesión*) y posteriormente codifica la clave haciendo uso de la llave pública del destinatario. Dicha clave es extraída convenientemente del anillo de claves públicas a partir del identificador suministrado por el usuario, todo ello de forma

transparente, por lo que únicamente debemos preocuparnos de indicar el mensaje a codificar y la lista de identificadores de los destinatarios. Nótese que para que el mensaje pueda ser leído por múltiples destinatarios basta con que se incluya en la cabecera la clave de sesión codificada con cada una de las claves públicas correspondientes.

Cuando se trata de decodificar el mensaje, PGP simplemente busca en la cabecera las claves públicas con las que está codificado y nos pide una contraseña. La contraseña servirá para que PGP abra nuestro anillo de claves privadas y compruebe si tenemos una clave que permita decodificar el mensaje. En caso afirmativo, PGP descifrará el mensaje. Nótese que siempre que queramos hacer uso de una clave privada, habremos de suministrar a PGP la contraseña correspondiente, por lo que si el anillo de claves privadas quedara comprometido, un atacante aún tendría que averiguar nuestra contraseña para descifrar nuestros mensajes. No obstante, si nuestro archivo de claves privadas cayera en malas manos, lo mejor será *revocar* todas las claves que tuviera almacenadas y generar otras nuevas.

Como puede comprenderse, gran parte de la seguridad de PGP reside en la calidad del generador aleatorio que se emplea para calcular las claves de sesión, puesto que si alguien logra predecir la secuencia de claves que estamos usando, podrá descifrar todos nuestros mensajes independientemente de los destinatarios a los que vayan dirigidos. Afortunadamente, PGP utiliza un método de generación de números pseudoaleatorios muy seguro —una secuencia aleatoria pura es imposible de conseguir, como se dijo en el capítulo 8—, y protege criptográficamente la semilla que necesita[1]. No obstante, consideraremos *sensible* al fichero que contiene dicha semilla —normalmente `RANDSEED.BIN`—, y por lo tanto habremos de evitar que quede expuesto.

18.2.2. Firma Digital

En lo que se refiere a la firma digital, las primeras versiones de PGP obtienen en primer lugar la signatura MD5 (ver sección 13.4), que posteriormente se codifica empleando la clave privada RSA correspondiente. Versiones más modernas implementan el algoritmo DSS, que emplea la función resumen SHA-1 y el algoritmo asimétrico DSA (secciones 12.4.4 y 13.5).

La firma digital o signatura puede ser añadida al fichero u obtenida en otro fichero aparte. Esta opción es muy útil si queremos *firmar* un fichero ejecutable, por ejemplo.

[1] Algunas implementaciones de PGP emplean otras fuentes de aleatoriedad, como ocurre con GnuPG, por lo que no necesitan almacenar una semilla aleatoria.

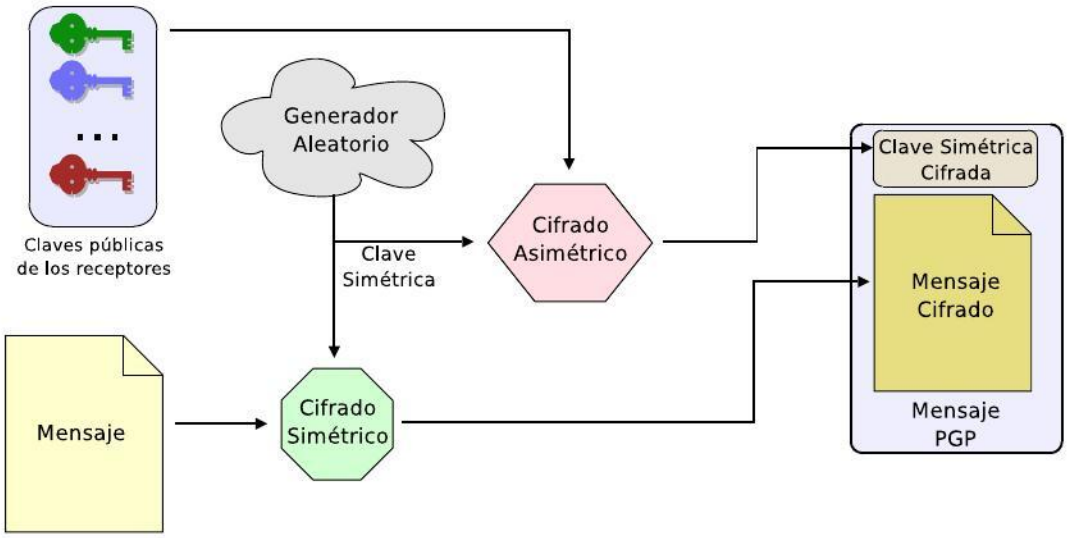

Figura 18.1: Codificación de un mensaje PGP

18.2.3. Armaduras ASCII

Una de las funcionalidades más útiles de PGP consiste en la posibilidad de generar una *armadura ASCII* para cualquiera de sus salidas. Obviamente, todas las salidas de PGP (mensajes codificados, claves públicas extraídas de algún anillo, firmas digitales, etc.) consisten en secuencias binarias, que pueden ser almacenadas en archivos. Sin embargo, en la mayoría de los casos puede interesarnos enviar la información mediante correo electrónico, o almacenarla en archivos de texto.

Recordemos que el código ASCII original emplea 7 bits para codificar cada letra, lo cual quiere decir que los caracteres situados por encima del valor ASCII 127 no están definidos, y de hecho diferentes computadoras y sistemas operativos los interpretan de manera distinta. También hay que tener en cuenta que entre los 128 caracteres ASCII se encuentran muchos que representan códigos de control, como el retorno de carro, el fin de fichero, el tabulador, etc. La idea es elegir 64 caracteres *imprimibles* (que no sean de control) dentro de esos 128. Con este conjunto de códigos ASCII podremos representar exactamente 6 bits, por lo que una secuencia de tres bytes (24 bits) podrá codificarse mediante cuatro de estos caracteres. Esta cadena de símbolos resultante se *trocea* colocando en cada línea un número razonable de símbolos, por ejemplo 72. El resultado es una secuencia de caracteres que pueden ser tratados como texto estándar, además de ser manipulados en cualquier editor. Existe la ventaja adicional de que esta representación es apropiada para ser enviada por correo electrónico, ya

que muchas pasarelas de correo no admiten caracteres por encima de 127, y además truncan las líneas demasiado largas, por lo que podrían alterar los mensajes si *viajaran* en otro formato.

Como ejemplo incluyo mi clave pública PGP —firmada con la de Kriptópolis— en formato ASCII:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.4 (GNU/Linux)

mQGiBDRkk6kRBADKYHrNnFeXlggr14IVGy6FudLG2Cd1wb3yKOaNnodyjZa0a5oi
Ls9jDfDfEdq8K+W6QBLv06w7oVFPNMYsU+ufb0pa/bHWq6IrHxKkTVH4o4PUYTmH
W0jfGjoXEtAUZ0vp9wYR0Yqi7wXO3L/N5KuVNjLj7rXOT7rOmHsOjmY1cQCg//2w
OcyAnkaDCODFNif/VdowntcD/j5midszzU6M7BWmeDJoqEEGzSuxfmRSNyNZe6/6
5k8TFXIVpB0vnxwsZSh0POSlNgz1cmX6VbEmmUXoYsMRfq7iXHSAZ3DLB333yR2b
QUbkrH5WZF75G2vvTO7rKS5KtmROJ8E+vX/py6PGz1f3tBZJ94KwM787g6j43F4X
IYTAA/9L5GZzClHOGt01BtZkioH5YoHnDGHKC8mMXcykXA5KdJvl+9jGz3InUHiG
04StaMxMcDcWLzL5FVLz3LBzlOXGs7jikgH3BYBI3p7dIExfRADucDHyKL/CpIl5
zqHBI+5bxY3Tysu3UlA1UkQloJMsSInlkkjQhwihNYsj8Avr9LYAAAAmTWFudWVs
IEx1Y2VuYSBMb3BleiA8bWx1Y2VuYUB1amFlbi5lcz6IVgQTEQIAFgUCOHyzZAQL
CgQDAxUDAgMWAgECF4AACgkQSLJRYWmrV4TqngCgsDk/ysnBdpPwp/r2dL0Lzcq0
1J8AnRxUUiS3SoVb3WfnaSQmdb6eaJ3qiEsEEBECAAsFAjTa4FoECwMBAgAKCRBI
slFhaatXhO9yAJ9vI1QWihIKMUa4g3S8t3EZZ9SXxgCaAjfnHx8Kayylm6XXjjsC
6iJKBmaIPwMFEDTa5h2buAet57tpPxEC8K4AoOTP5I1fJFN6KtZdmLtENKSRrKfx
AJ4gwl5R1MzpeTFiysWKab/PsU5GwohGBBARAgAGBQI3eQrfAAoJEPi4YmyN8qnz
A1sAniVQF6V/6gBVPq0Idt1Yrtuy4+aQAKDTuyVvfU1tRNy/U89FhzMmBVRL44ht
BBERAgAtBQI+JnRPBYMB4TOAIBpodHRwOi8vd3d3LnRvZWhvbGQuY29tL3JvYm90
Y2EvAAoJEBBYFoXFIQl+g80An1lb7UmR7euGyIwluvc4n84w3opTAJwKLudIa08d
6eOKeSmDMwMYsmHCZrkCDQQ0ZJRfEAgAw/iGbTW9OaTyfV4RNZdg1HRDGEyasZdE
PCM9ihPkfvQyK44nH13OseaikIYoyoA/BFiWeTNcHvb/4KOuCK2GnO/p/6ohFcAO
K5anEygGrhUUttUw8kYZ0rUBFIJnurtDcxwawugbPFv3qA+sn756q7XUxjnTtpou
+lWyj6VkN/EvrZDf9E7ikPUqRuIsHzJ5PUwypWtXaKg2HfClKkZlYFqzdPDCssrX
OfjZDx2q6GSek6Sgj5Ph3X4opoXIx6Cfmp4ELYmvdmnDu4oe6A6l/XIQ8NNhj+Gx
dtOgTq8QKDRWl2f6M3pQgPnYzBHoDIqnr/ie8jK4seDezRPtL1/TlQACAgf+JXw0
3Q1opLBAaO/WZlcs2SiEzqv+gCkFW9vk2bJbSY4PQHwiLc0HwcPEDi7jIu9QxJfZ
cHkax8XgXkCvfFJFFmqgqarIOzXp/BgiYyma6GVAmXcI6lI9ZSgzPvvaNFGe0/7R
6Yroee7nJ/9RyxF89SI++5tZY+/bpLuKAbnX9SA3PEnUWiHD2ah3cC3VXNrus3ls
KA7MEh3q9xnoF/8Z7vwldrKUyLZdaDqSM7isyI5Fe0PWn/mtW4+7/rjboaY7PGJC
Aqtn8cHDvByRYCZ8kLRlobQHzL8XN1fsdfBv6WDNeS9IqBCXcPME7R21wytsi2WM
DnYL7rQWU/CgLqFx2Ig/AwUYNGSUX0iyUWFpq1eEEQL3JACfTfvh6A70A9N2SbnR
BmktuRBp9NsAn2ZQbpg0eaeVRuzejA2QM7ldrz53mQENAzRkY3EAAAEIAOy6UGjP
ly4OJtrPookV6kxUjmL83LY6jI+ZRH5/ZkHeLcQ+Qufmme+bOms5XHv+KTOkKV5R
UdJwUXTQtHe+yX7XcvnLlxnxE/dmhgeNHcLH0XfQ9rBjIvREcKtRhUBP1t0d+QTn
Uro7Jrg8ZQSTupLTb5LO7683FfF/2eBSMsn1QZx6ODSir4t05EqMOzlHc53jV8y2
```

```
bYDRDMhQ5r1C1ap0vZS6tp85Wb64+aun0et1yee4voeUwNubnr1FBXzfBwQUy4e4
IdkJbXYb1f8Iy7+t5B3WviU1BgGQOP29fObjg7eMtXUgaF6eYK88Byu7tHMuFYQR
Oeq37dWwHELi6LEABRO2AAAAJE1hbnVlbCBMdWNlbmEgUlNBIDxtbHVjZW5hQHVq
YWVuLmVzPokBFQMFEDY7Tx/t1bAcQuLosQEBsCEIAOvHEw9fuHTqWpRMxtvqYZnf
oslqg27vNC4fE4QGc/KhyxwCeqm/fUh51geVMna7QwLubbHcmd4IPAZT614LdAhw
zDzv99o+iHDwL3fv+LJWqdmkxCZYHJs7vKMSShaVCd9JXPe5FT6iXIukIy3oCU5T
kjumRZNzr40MgQZzYW5rxCfYX+feoLaX8SBR7EU2mdX1LaMy+RJ3cG7a76btqdKn
Lx+E5USlIDWC26sk7Y+Dutp987F3ZHW9TVO5IUHn5TqirxnL7a6ZGn2c2oq4V27l
oCuFxo6/KJ1m92tddM35SHzoijA6hWWH9OsaMiA6d4Qu1LbK28NElnAo+FZUitOJ
ARUDBRA3eQqZGIrd0JPxO/EBAcE1B/0WG4aU63s6E5lLvlUSsZ0yTtUBPETH0K2y
l5scacX5+uF/7+hHGAoN0/yTbpB0ODoxVNxgctkRDf82QHtDr4HkGdvdux/Grwuu
FigKQoeBEpPZ1yAcLX/zcWKPjveysNhywngFHmzdnvCXJeegWsila1BEoWT4OeMp
elY7U9zH7RLY/u66yKM5TIitFYd5uOWO+SSWnkvD8KvbAv2UYYXbK9aXieIKnrt/
F8SB8nCQzxAUy2A1JBJoQt07B4AyevHhjOZ1zAsUdwVk0zcfx1AsI0N9hWE9NUh/
3WLyKSe4IF35wYZEdcy7+i0SwBinZ4dls8wwvcISF3JwqJmDhL+c
=2wgK
-----END PGP PUBLIC KEY BLOCK-----
```

Como puede verse, los únicos símbolos empleados son las letras mayúsculas y minúsculas, los números, y los signos '/'y '+'; el resto de símbolos y caracteres de control simplemente será ignorado. Cualquiera podría copiar esta clave pública a mano (¡!) o emplear una aplicación OCR[2] para introducirla en su anillo de claves correspondiente, aunque es mejor descargarla a través de Internet.

18.2.4. Gestión de Claves

PGP, como ya se ha dicho, almacena las claves en unas estructuras denominadas *anillos*. Un anillo no es más que una colección de claves, almacenadas en un fichero. Cada usuario tendrá dos anillos, uno para las claves públicas (`PUBRING.PKR`) y otro para las privadas (`SECRING.SKR`).

Cada una de las claves, además de la secuencia binaria correspondiente para el algoritmo concreto donde se emplee, posee una serie de datos, como son el identificador del usuario que la emitió, la fecha de expiración, la versión de PGP con que fue generada, y la denominada huella digital (*fingerprint*). Este último campo es bastante útil, pues se trata de una secuencia hexadecimal lo suficientemente larga como para que sea única, y lo suficientemente corta como para que pueda ser escrita en un papel, o leída de viva voz. La huella digital se emplea para asegurar la autenticidad de una clave. Por ejemplo, la huella digital de la clave pública anterior es:

[2]OCR: *Optical Character Recognition*, reconocimiento óptico de caracteres. Permite convertir texto escrito a formato electrónico.

```
9E2B 9D14 CBCE FE12 16A8  C103 48B2 5161 69AB 5784
```

Si alguien quisiera asegurarse de la autenticidad de dicha clave, bastaría con que llamara por teléfono al autor, y le pidiera que le leyera su huella digital. Afortunadamente, las últimas implementaciones de PGP permiten convertir esta cadena hexadecimal en una secuencia de palabras fácilmente legibles por teléfono.

18.2.5. Distribución de Claves y Redes de Confianza

PGP, como cualquier sistema basado en clave pública, es susceptible a ataques de intermediario (sección 12.2). Esto nos obliga a establecer mecanismos para asegurarnos de que una clave procede realmente de quien nosotros creemos. Una de las cosas que permite esto, aunque no la única, es la *huella digital*.

PGP permite a un usuario firmar claves, y de esta forma podremos confiar en la autenticidad de una clave siempre que ésta venga firmada por una persona de confianza. Hay que distinguir entonces dos tipos de confianza: aquella que nos permite creer en la validez de una clave, y aquella que nos permite fiarnos de una persona como certificador de claves. La primera se puede calcular automáticamente, en función de que las firmas que contenga una clave pertenezcan a personas de confianza, pero la segunda ha de ser establecida manualmente. No olvidemos que el hecho de que una clave sea auténtica no nos dice nada acerca de la persona que la emitió. Por ejemplo, yo puedo tener la seguridad de que una clave pertenece a una persona, pero esa persona puede dedicarse a firmar todas las claves que le llegan, sin asegurarse de su autenticidad, por lo que en ningún caso merecerá nuestra confianza.

Cuando una clave queda comprometida, puede ser revocada por su autor. Para ello basta con generar y distribuir un *certificado de revocación* que informará a todos los usuarios de que esa clave ya no es válida. Para generarlo es necesaria la clave privada, por lo que en muchos casos se recomienda generar con cada clave su certificado de revocación y guardarlo en lugar seguro, de forma que si perdemos la clave privada podamos revocarla de todas formas. Afortunadamente, las últimas versiones de PGP permiten nombrar revocadores de claves, que son usuarios capaces de invalidar nuestra propia clave, sin hacer uso de la llave privada.

18.2.6. *Otros* PGP

La rápida popularización de PGP entre ciertos sectores de la comunidad de Internet, y el desarrollo del estándar público *Open PGP*, han hecho posible la proliferación de variantes más o menos complejas del programa de Zimmermann. Muchas de ellas

son desarrolladas por los propios usuarios, para mejorar alguna característica, como manejar claves de mayor longitud (PGPg), y otras corresponden a aplicaciones de tipo comercial.

Especial mención merece la implementación de *Open PGP* que está llevando a cabo el proyecto GNU: GnuPG (*GNU Privacy Guard*), que funciona en múltiples plataformas, y emplea únicamente algoritmos de libre distribución —entre ellos AES—, aunque presenta una estructura que la hace fácilmente extensible. De hecho, hoy por hoy, podríamos decir que es la implementación de PGP más completa, segura y útil para cualquier usuario.

18.3. Vulnerabilidades de PGP

Según todo lo dicho hasta ahora, parece claro que PGP proporciona un nivel de seguridad que nada tiene que envidiar a cualquier otro sistema criptográfico jamás desarrollado. ¿Qué sentido tiene, pues, hablar de sus *vulnerabilidades*, si éstas parecen no existir?

Como cualquier herramienta, PGP proporcionará un gran rendimiento si se emplea correctamente, pero su uso inadecuado podría convertirlo en una protección totalmente inútil. Es por ello que parece interesante llevar a cabo una pequeña recapitulación acerca de las *buenas costumbres* que harán de PGP nuestro mejor aliado.

- *Escoger contraseñas adecuadas*. Todo lo comentado en la sección 17.5.1 es válido para PGP.
- *Proteger adecuadamente los archivos sensibles*. Estos archivos serán, lógicamente, nuestros llaveros (anillos de claves) y el fichero que alberga la semilla aleatoria. Esta protección debe llevarse a cabo tanto frente al acceso de posibles curiosos, como frente a una posible pérdida de los datos (¡recuerde que si pierde el archivo con su clave privada no podrá descifrar jamás ningún mensaje!).
- *Emitir revocaciones de nuestras claves al generarlas* y guardarlas en lugar seguro. Serán el único mecanismo válido para revocar una clave en caso de pérdida del anillo privado. Afortunadamente, la versión 6 de PGP permite nombrar *revocadores* para nuestras claves, de forma que éstos podrán invalidarla en cualquier momento sin necesidad de nuestra clave privada.
- *Firmar sólo las claves de cuya autenticidad estemos seguros*. Es la única manera de que las redes de confianza puedan funcionar, ya que si todos firmáramos las claves alegremente, podríamos estar certificando claves falsas.

Al margen de un uso correcto, que es fundamental, debemos mencionar que últimamente han sido detectados algunos fallos en las diversas implementaciones de PGP. Clasificaremos dichas vulnerabilidades en dos grupos claramente diferenciados:

- *Debidas a la implementación*: Estos agujeros de seguridad son provocados por una implementación defectuosa de PGP, y corresponden a versiones concretas del programa. Por ejemplo, el fallo descubierto en la versión 5.0 de PGP para UNIX, que hacía que las claves de sesión no fueran completamente aleatorias, o el encontrado en todas las versiones para Windows, desde la 5.0 a la 7.0.4, en la que un inadecuado procesamiento de las armaduras ASCII permitía a un atacante introducir ficheros en la computadora de la víctima.
- *Intrínsecas al protocolo*: En este apartado habría que reseñar aquellos agujeros de seguridad que son inherentes a la definición del estándar *Open PGP*. En este sentido, a principios de 2001 se hizo pública una técnica que permitía a un atacante falsificar firmas digitales. En cualquier caso, se necesita acceso físico a la computadora de la víctima para manipular su clave privada, por lo que el fallo carece de interés práctico.