L'option GnuPG qui permet de ne pas indiquer les ID...

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 19 mars 2014

http://www.bortzmeyer.org/gpg-option-no-keyid.html

La sécurité, ce n'est pas gratuit. Il y a parfois des techniques qui améliorent la sécurité mais qui sont tellement embêtantes à l'usage qu'on préfère les abandonner. C'est ainsi que je viens de supprimer l'option `throw-keyids` de ma configuration GnuPG.

Cette option est a priori bonne pour la sécurité : pour citer la documentation <http://www.gnupg.org/documentation/manuals/gnupg-devel/GPG-Esoteric-Options.html> elle permet de « ne pas mettre les identificateurs de clé dans le message ». Qu'est-ce que cela veut dire ? Par défaut, GnuPG, lorsqu'il chiffre un message, indique dans le message l'identificateur de la clé du (ou des destinataires). C'est décrit dans le RFC 4880 [1], section 5.1. On peut l'afficher avec `gpg` :

```
% gpg report.txt.gpg
gpg: encrypted with 2048-bit RSA key, ID 336525BB, created 2009-12-15
      "ISC Security Officer <security-officer@isc.org>"
...
```

Ici, tout le monde peut voir que ce document a été chiffré pour la clé publique `336525BB`, même si, en l'absence de la clé privée, on ne peut pas le déchiffrer. Ainsi, quelqu'un qui a accès au fichier a quand même une information utile, que j'écris des choses pour l'ISC. C'est une métadonnée qui peut être utile à un espion.

L'option `throw-keyids` permet de résoudre cette indiscrétion :

```
% gpg report.txt.gpg
gpg: anonymous recipient; ...
```

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4880.txt`

Et voilà, l'information a disparu. En mettant `throw-keyids` dans son `/.gnupg/gpg.conf`, on fabriquera toujours des messages anonymes. J'avais mis cette option dans ma configuration à l'occasion d'un durcissement de sécurité qui accompagnait ma nouvelle clé PGP <http://www.bortzmeyer.org/nouvelle-cle-pgp.html>.

Mais, car il y a un mais, le destinataire légitime ne saura donc pas si le message est bien pour lui, il devra essayer toutes ses clés secrètes. Cela prend du temps et, surtout, cela crée des messages ennuyeux :

```
gpg: anonymous recipient; trying secret key 1B217B95 ...
gpg: protection algorithm 1 (IDEA) is not supported
gpg: the IDEA cipher plugin is not present
gpg: please see http://www.gnupg.org/faq/why-not-idea.html for more information
gpg: anonymous recipient; trying secret key 1CE01D31 ...
gpg: protection algorithm 1 (IDEA) is not supported
gpg: anonymous recipient; trying secret key CCC66677 ...
gpg: anonymous recipient; trying secret key 96A4A254 ...
...
```

Notez que GnuPG a essayé toutes les clés secrètes de mon trousseau (y compris les révoquées : le message est peut-être un très ancien, fait avec une vieille clé). Comme certaines sont vraiment anciennes, elles utilisent même IDEA d'où l'avertissement anti-IDEA <http://www.gnupg.org/faq/why-not-idea.html>. D'expérience avec cette option, la plupart de mes correspondants ne comprennent pas ces messages, les interprètent mal, s'énervent, etc. Bref, j'ai fini par supprimer cette option de mon fichier de configuration.

Notez que, pour l'utilisation de PGP pour le courrier électronique, ce n'est pas forcément très grave. Dans le cas « normal » (pas celui du prudent qui change d'adresse de courrier tout le temps), l'espion éventuel a de toute façon tous les en-têtes RFC 5322 à sa disposition pour accéder à la même information. Pour des fichiers transmis par un autre moyen, il peut être toujours utile de se servir de `throw-keyids`, par exemple ponctuellement sur la ligne de commande :

```
% gpg --recipient 9EE8C47B --throw-keyids  --encrypt report.txt
```

Attention, il existe d'autres possibilités <http://security.stackexchange.com/a/22705> de fuite de l'information.

Merci à Ollivier Robert et Florian Maury pour des discussions intéressantes.