Un nouveau logiciel post-Snowden dans ma logithèque, TextSecure

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 12 octobre 2014. Dernière mise à jour le 1 novembre 2014

https://www.bortzmeyer.org/textsecure.html

Je suis désormais, comme beaucoup de gens, l'heureux utilisateur de TextSecure qui a complètement remplacé, sur mon "*smartphone*", l'application SMS par défaut.

À quoi sert TextSecure ? C'est un logiciel de messagerie instantanée, conçu pour être très simple d'usage, pour servir de remplaçant « tel quel » aux logiciels SMS actuels, tout en offrant une meilleure protection de la vie privée, dans certains cas. Remplaçant « tel quel », car l'idée de base est que les utilisateurs normaux (le célèbre M. Michu) ne devraient pas avoir à renoncer à quoi que ce soit pour avoir une meilleure sécurité, dans un monde post-Snowden. En effet, M. Michu peut mettre TextSecure, gérer tous ses SMS avec, et il ne verra guère de différence. Contrairement à bien d'autres outils de sécurisation, qui sont, au minimum, moins faciles à utiliser que les outils habituels, au pire, franchement hostiles aux utilisateurs, TextSecure peut être recommandé aux parents, aux ministres, et autres populations non-"*geeks*", sans se dire qu'on leur joue un mauvais tour. Sans doute n'offre-t-il pas de sécurité en béton, mais il ne fait rien perdre à l'utilisateur non plus. On peut résumer TextSecure en disant qu'il offre un très bon compromis entre facilité d'usage et sécurité.

Bon, qu'est-ce que voit l'utilisateur, avec TextSecure ? Justement, rien de particulier : il s'en sert comme de l'outil SMS par défaut. Le principal changement est que certains messages vont être marqués d'un joli cadenas : si le correspondant utilise également TextSecure, les messages sont chiffrés automatiquement avec sa clé. Si on est simple utilisateur, on a donc un peu plus de vie privée, et on en aura de plus en plus au fur et à mesure que TextSecure se répand.

TextSecure peut utiliser deux transports différents (mais on peut débrayer l'un ou l'autre), le SMS et la liaison « données » du "*smartphone*" (ce second transport se nomme "*PUSH*" dans la terminologie TextSecure, et est utilisé par défaut pour les correspondants qui ont également TextSecure). Le mode "*PUSH*" peut être intéressant si on a un quota SMS limité mais pas en Internet, et le mode SMS dans le cas contraire. Les messages apparaissent en vert traditionnel quand ils ont été transportés en SMS et en bleu autrement (curieux choix de couleur : les messages en vert ne sont pas forcément les plus sûrs,

puisqu'ils ne sont en général pas chiffrés, c'est peut-être une copie aveugle de ce que fait Apple <http://www.ianswerguy.com/green-blue-iphone-messages/>, suggère Jean Champémont).

TextSecure peut aussi chiffrer toute la base des SMS reçus et stockés. Ainsi, même si on vole votre téléphone, vos communications resteront sûres. Attention : si vous oubliez la phrase de passe, tout est fichu. Pensez à sauvegarder.

Si maintenant on est un utilisateur pas simple, qui veut savoir comment ça marche, on se pose quelques questions. Voici celles que je me posais et les réponses que j'ai pu trouver. Notez que la société qui développe TextSecure, OpenWhisper, a aussi une bonne FAQ <http://support.whispersystems.org/>.

D'abord, on a dit que TextSecure trouvait automatiquement les contacts de l'utilisateur qui étaient également utilisateurs de TextSecure. Ouh là là, se dit le paranoïaque moyen, c'est pas bon pour la vie privée. Comment il fait ? Il envoie tout mon carnet d'adresses à la société qui gère TextSecure, pour comparaison ? Cela serait catastrophique pour la sécurité. Le problème est difficile : si on veut de la sécurité maximale, il faudrait envoyer **zéro** bit d'information à l'extérieur. Mais le cahier des charges de TextSecure n'est pas « la sécurité maximale et rien d'autre », c'est « la meilleure sécurité du monde ne sert à rien si elle n'est pas déployée, car trop complexe ou trop pénible ». Pas question, donc, de se passer d'un système de découverte automatique. Il existe des algorithmes complexes et très astucieux (décrits dans ce remarquable article <https://whispersystems.org/blog/contact-discovery/>), mais le problème reste largement ouvert.

L'expert en sécurité qui sommeille dans chacun de mes lecteurs a déjà compris que les métadonnées de la communication, elles, sont en clair et que des tiers peuvent donc savoir qui écrit à qui et quand, même si le contenu des messages est chiffré. Comme le note Alda Marteau-Hardi <https://twitter.com/Alda> « les "*metadata*" sont connues d'OpenWhisper [si on utilise "*PUSH*"] ou de ton opérateur mobile [si on utilise le SMS traditionnel], au final ».

Au fait, toujours si je suis expert en sécurité paranoïaque, pourquoi ferais-je confiance à OpenWhisper plus qu'à la NSA ou à Orange ? La principale raison est que TextSecure est un logiciel libre : le code est disponible <https://github.com/WhisperSystems/TextSecure/>, et des experts en sécurité l'ont déjà lu et personne n'a encore trouvé de défaut significatif. Notez que le code du serveur utilisé par OpenWhisper est également disponible en ligne mais c'est moins utile puisque, de toute façon, on ne peut pas vérifier que le serveur effectif exécute bien ce code.

Et si je lis dix RFC le matin avant mon petit déjeuner et que je veux connaître le protocole exact utilisé ? Pas de problème, ce protocole est documenté <https://github.com/WhisperSystems/TextSecure/wiki/ProtocolV2>.

OK, mais il reste **le** gros problème de la cryptographie, authentifier le type en face. C'est bien joli de tout chiffrer mais, si Alice croit parler à Bob alors qu'elle parle en fait à Mallory, le chiffrement ne sert pas à grand'chose. Comment TextSecure traite-t-il ce problème ? À la première communication, TextSecure fait confiance. C'est risqué, si un Homme du Milieu se trouvait là, mais ce système, dit TOFU ("*Trust On First Use*") a l'avantage d'être trivial d'utilisation. Il correspond donc bien au cahier des charges de TextSecure, qui veut que le logiciel soit effectivement utilisé et, pour cela, qu'il ne rebute pas ses utilisateurs. Ce modèle est également utilisé par SSH et, comme le note le RFC 5218 [1], est une des principales causes

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5218.txt

de son succès. Je n'ai pas encore testé ce qui se passait si le correspondant réinitialise sa clé mais, d'après la documentation, TextSecure vous prévient (puisque cela peut indiquer une attaque de l'Homme du Milieu).

Si on veut, on peut toujours vérifier, lors d'une rencontre AFK <http://www.catb.org/jargon/html/A/AFK.html>, la clé de son correspondant, soit en la lisant à l'écran (la mienne est BdY73Lfk12kv9iTVUTGIL6VZr en Base64 ou bien 05 d6 3b dc b7 e4 d7 69 2f f6 24 d5 51 31 88 2f a5 59 ae 96 e0 fb a5 75 ab e6 6c 64 ca e9 bb 6a 77 en suite de chiffres hexadécimaux), soit via des "*codes QR*". Voici d'ailleurs le mien :

Comme beaucoup de points de TextSecure, c'est bien documenté en ligne <http://support.whispersystems.org/customer/portal/articles/1470045-how-do-i-verify-the-person-i-m-sendin Mais, et c'est à mon avis une des plus grosses faiblesses de TextSecure, je n'ai trouvé aucun moyen d'enregistrer le fait qu'on a vérifié, afin, par exemple, d'afficher les messages ultérieurs d'une couleur différente si le correspondant a ainsi été solidement authentifié. (La question est discutée dans la bogue #314 <https://github.com/WhisperSystems/TextSecure/issues/314> et la #689 <https://github.com/WhisperSystems/TextSecure/issues/689>.)

X_cli <https://twitter.com/X_Cli> (qui met aussi son code QR TextSecure sur sa page d'accueil <https://x-cli.eu/>) m'a suggéré une technique rigolote et, à ma connaissance, jamais utilisée : mettre la clé TextSecure dans une identité PGP, que les gens qui ont vérifié pourront signer. C'est fait pour ma clé CCC66677 :

```
% gpg --list-keys CCC66677
...
uid  TextSecure fingerprint (05 d6 3b dc b7 e4 d7 69 2f f6 24 d5 51 31 88 2f a5 59 ae 96 e0 \
                fb a5 75 ab e6 6c 64 ca e9 bb 6a 77) \
                <BdY73Lfk12kv9iTVUTGIL6VZrpbg+6V1q+ZsZMrpu2p3@base64>
...
```

Au passage, pour exporter la clé publique, le plus simple est d'afficher le code QR, de partager par courrier électronique, et le message contient la clé en Base64. Si on la veut en liste de chiffres hexadécimaux, on peut se servir de ce petit script Python (en ligne sur https://www.bortzmeyer.org/files/base64-to-hexas.py).

Autre défaut, et une exception à la règle comme quoi installer TextSecure ne fait rien perdre par rapport à l'application par défaut, envoyer un message à plusieurs personnes n'est pas possible simplement (il faut d'abord créer un groupe statique, qui, si un membre n'a pas TextSecure, est converti en groupe MMS dont l'usage est franchement pénible - et donne de drôles de résultats sur les vieux téléphones). Il y a également un intéressant article sur les groupes <https://whispersystems.org/blog/private-groups/>. Autre problème, TextSecure affiche obligatoirement l'heure de réception du message et pas son heure de départ. Si le téléphone a été éteint longtemps, c'est pénible (on a accès à l'heure de départ en sélectionnant le message puis en choisissant l'option I comme Information, en haut).

À noter qu'il existe des logiciels concurrents. Le plus proche me semble être Telegram. Une discussion de comparaison entre TextSecure et Telegram a eu lieu sur Ycombinator <https://news.ycombinator.com/item?id=6913456>.

Merci à Amaelle Guiton et Adrienne Charmet pour avoir servi de cobayes et pour les intéressantes discussions.
