Uso de la criptografía simétrica para la comunicación de mensajes cortos en dispositivos móviles

Adolfo Di Mare Hering
José Pablo Noguera Espinoza

Universidad de Costa Rica
Escuela de las Ciencias de la Computación e Informática

{adolfo.dimare,jose.nogueraespinoza}@ucr.ac.cr

Resumo: *Presentamos un nuevo enfoque para usar cifrado simétrico de mensajes cortos (SMS). Los archivos de fotos o de música se usan como llave de encripción pues tienen alta entropía y están disponibles en los dispositivos móviles a las que muchas personas tienen acceso. Se analizan las fortalezas y debilidades del protocolo propuesto, que resulta en la comunicación segura entre un grupo pequeño de personas.*

Abstract: *We present a new approach to use symmetric encryption for short message service (SMS). Photo or music files are used as the encryption key because they have high entropy and are available on mobile devices to which many people have access. The strengths and weaknesses of the proposed protocol are analyzed, resulting in secure communication among a small group of people.*

Palabras claves: Móvil, Seguridad, Criptografía.

1 Introducción

En la actualidad, el uso extendido de la telefonía móvil a nivel mundial a roto barreras de comunicación y ha creado nuevos formas de interactuar de las personas, creando una dependencia en su uso [Muñoz-2009]. Uno de los servicios importantes que brinda esta tecnología es el servicio de mensajes cortos de texto SMS (*Service Message Short*), el que ya es percibido como parte fundamental de la telefonía móvil. Desafortunadamente, la confidencialidad de la información, que es uno de los tres pilares de la seguridad [Bishop-2002], ha sido severamente comprometido debido al control que ejercen a nivel mundial organizaciones como es la Agencia de Seguridad Nacional de los E.E.U.U. (NSA) [Bishop-2002]. En particular, la NSA ha sido desenmascarada por sus actividades de espionaje de todas la comunicaciones, a nivel global, hecho que fue revelado por uno de los ex-agentes de esa organización: Edward Snowden [Lara-2014]. Estos hechos de espionaje han resultado en críticas de todos los países, en particular de Alemania [Müller-2014].

Figura 1: Beware of NSA watchdog [Ohman-2013]

Estas intromisiones inesperadas han causado revuelo entre los ciudadanos estadounidense quienes han sentido que sus derechos han sido violentados, como lo muestra la caricatura de la Figura 1.

A nivel local, en Costa Rica , el 20 de enero del 2014 el diario La Extra acusó al Organismo de Investigación Judicial (OIJ) por espionaje, como lo describe el reportaje del noticiero de Televisora de Costa Rica, canal 7:

> El Diario Extra publicó una noticia en la que denuncian que "algunas de las más altas autoridades del Poder Judicial (OIJ) han espiado a periodistas de este medio, para conocer cuáles son sus fuentes informativas" [Quesada-2014].

Esto hecho creó un antecedente que violenta la libre expresión y comunicación en el país. La Sala Constitucional, mediante la resolución No.2014-4035, sentenció el recurso de amparo presentado por diario La Extra y las coadyudancias de varios medios de comunicación, que pretendían declarar ilegal el rastreo telefónico del cual fue víctima el periodista Manuel Estrada y poner, ante todo, derechos como la libertad de expresión, reserva de la fuente y derecho a informar a la población [Aguilar-2014].

En el mercado de aplicaciones móviles existen algunas empresas que han creado aplicaciones para el envío de mensajes simples de forma segura, como por ejemplo "*SMS Encrypt*" [FMontano-2012]. Estas organizaciones utilizan algoritmos conocidos que se han convertido en un estándar en seguridad y en requisito para el trasiego de cualquier tipo de datos. Las funcionalidades básicas ya han sido identificadas y están descritas en muchas investigaciones: por ejemplo, "*SafeSlinger*" [FLKMP-2013] describe que la utilización de esquemas de llave pública pueden ser comprometidos pues en Internet es posible capturar esas llaves. "*SafeSlinger*" es un sistema diseñado para evitar que las claves criptográficas sean comprometidas, que es uno de los problemas recurrentes en la comunicación de datos.

2 Criptografía

Un buen sistema criptográfico debe tener la cualidad de que si ya se conoce el texto en claro y el texto cifrado debe resultar más caro en tiempo o recursos descifrar la clave que el valor posible de la información obtenida por terceros intercambiar de forma segura y privada sus claves públicas en Internet.

Figura 2: Encriptación Simétrica [Moreno-2004]

Los algoritmos de criptografía se dividen en dos grandes familias. Si se usa la misma clave para encriptar y para desencriptar el algoritmo se llama simétrico, mientras que si se usan 2 claves al método criptográfico se le llama asimétrico. Los primeros algoritmos fueron simétricos, pues son más fáciles de formular. La Figura 2 muestra su funcionamiento.

El algoritmo de encriptación usado siempre es conocido: su fortaleza es producto de la longitud de la clave empleada. Una de las formas de criptoanálisis primario de cualquier tipo de sistema es la de “prueba y ensayo”, que consiste en probar diferentes claves hasta encontrar la correcta. Los algoritmos simétricos encriptan bloques de texto del documento original, y son más sencillos que los sistemas de clave pública.

El sistema criptográfico propuesto en este trabajo usa criptografía simétrica, por lo que tanto el emisor como el receptor deben conocer y mantener una misma clave: si la llave cayera en manos de terceros ya no habría comunicación segura. Un algoritmo simétrico es fiable si cumple estos requisitos [Moreno-2004]:

1. Conocido el criptograma (texto cifrado) no se pueden obtener de él ni el texto en claro ni la clave.
2. Conocidos el texto en claro y el texto cifrado debe resultar más caro en tiempo o recursos descifrar la clave que el valor posible de la información obtenida por terceros.

Las principales desventajas de los métodos simétricos son la distribución y almacenamiento de las claves. Si muchas personas deben conocer la misma clave es difícil proteger esa clave que todos deben compartir [Moreno-2004].

La criptografía asimétrica usa 2 claves, que usualmente se llaman clave pública y clave privada. Estas claves tienen cualidades matemáticas especiales, de tal forma que se generan siempre a la vez, por parejas, de manera que ambas están intrínsecamente ligadas. En particular, si dos llaves públicas son diferentes, sus claves privadas asociadas también lo son (y viceversa). Los algoritmos asimétricos están basados en funciones matemáticas fáciles de resolver en un sentido, pero muy complicadas de realizar en sentido inverso. Por eso, deber ser muy difícil obtener la llave privada a partir de la pública.

Debido a las calidades matemáticas que deben tener las llaves cuando se usa criptografía asimétrica, lo usual es que ambas claves sean generadas por computadora sin que las elija el usuario.

3 Contexto actual de la mensajería SMS

Existen diferentes técnicas de infiltración en dispositivos móviles y teléfonos inteligentes. Una técnica de amplio uso es incluir código malicioso en los archivos en que se distribuyen aplicaciones Android, que usan el formato APK. De esta forma, quien instala una aplicación desde el archivo APK también instala el virus que permite infiltrar el dispositivo, como lo reportan en [KKC-2012]. Por eso, se ha usado criptografía para evitar este tipo de ataques que comprometen la privacidad de comunicación con dispositivos móviles.

Talvez no se ha explorado mucho cómo incorporar criptografía para trasegar mensajes de texto SMS porque ese servicio es muy simple y se usa de manera informal. Sin embargo, es claro que este tipo de comunicación es esencial, como lo demuestra el crecimiento exponencial de la cantidad de usuarios de servicios como "*WhatsApp*" o "*Telegram*" [Kerr-2014].

El cifrado fuerte demanda uso de claves de alta seguridad, donde la distribución de la clave se convierte en un gran problema a resolver [FLKMP-2013]. Aquí proponemos un forma de abordar este inconveniente, para lo cual se plantea una solución sencilla de distribución de llaves por proximidad en el contexto de la comunicación SMS entre dos personas.

Muchas personas disfrutan hoy en día de la telefonía móvil, y poseen aparatos que son computadores avanzados y de gran potencia, de manera que pueden utilizar aplicaciones de mensajería instantánea (como "*WhatsApp*" o "*Telegram*"), las que usan como transporte Internet en lugar del servicio SMS de la compañía telefónica. Todavía existen pocas aplicaciones que utilizan criptografía asimétrica para este tipo de comunicación (como por ejemplo "*TextSecure Private Messenger*"), pero no se ha explorado el uso de criptografía simétrica en el contexto de la mensajería inmediata entre pocas personas.

Nuestra propuesta es muy simple, pues presume que cuando 2 personas quieren tener comunicación segura vía SMS solo se ponen de acuerdo en usar un archivo de llave, que ambos ya hayan almacenado en su dispositivo móvil, y que cumpla un requisito muy simple: tener alta entropía de información. Este requisito lo cumplen las fotos o piezas musicales almacenadas en nuestros dispositivos, pues generalmente los formatos que se usan resultan en archivos de alta entropía porque están empacados para reducir su tamaño. Este es el caso de los formatos "*PNG*" o "*JPG*" de imágenes, o "*MP3*" de música y "*FLV*" de video (también sirve un archivo "*ZIP*" o "*RAR*").

No importa si esta llave viaja vía Internet o es copiada con "*BlueTooth*" entre las personas que quieran establecer comunicación SMS encriptada, pues el protocolo que presentamos no busca atender a muchas personas, como ocurre cuando un banco acude a métodos de encriptamiento para atender a miles de clientes. Por eso, cuando dos personas quieren usar mensajes cifrados, simplemente se ponen de acuerdo en utilizar algún archivo como llave.

La protección de la llave también pasa a un segundo plano, pues se almacena en cada dispositivo móvil sin ninguna seguridad adicional. Esto implica que si alguien se apodera del dispositivo, podrá tener acceso a toda la clave. Sin embargo, los mensajes SMS no están encriptados en cada dispositivo, sino que se almacenan de manera legible pues lo que se busca es lograr que el trasiego de mensajes sea seguro, pero cuando cada mensaje llega a su destinatario toda forma de encriptamiento o codificación desaparece.

Este protocolo busca preservar la privacidad de las personas y supone que el atacante es un ente grande y potente, con gran presupuesto, como lo es el gobierno de EE.UU. a través de su agencia NSA, que utiliza enormes computadores para almacenar y clasificar los mensajes de todas las personas.

Para apreciar la magnitud del problema de privacidad actual, en que la mensajería instantánea viaja por las redes sin ninguna forma de encripción, se puede usar el siguiente cálculo resumido. Supongamos que en EE.UU. hay alrededor de 350 millones de personas quienes envían no más de 50 mensajes por hora. Si la mayoría de los mensajes de texto son de 150 caracteres de longitud, el número de caracteres trasegados por año vía SMS es la siguiente:

$$22{,}995{,}000{,}000{,}000 = 350{,}000{,}000 * 50 * (24 * 365) * 150$$

Tan grande como aparece este número, cuando se expresa en megabytes se convierte en 22,995,000,000, que es lo mismo que 22,995,000 gigabytes o 22,995 terabytes. Si nos fijamos en una tienda de computación (como Amazon.com) podemos constatar que cada terabyte de almacenamiento cuesta alrededor de $50 dólares, lo que significa que cualquier persona con un presupuesto de $1.149.750 dólares pueda comprar suficiente espacio en disco para almacenar todos los mensajes SMS de los EE.UU. Para el gobierno de EE.UU, el gobierno chino o el gobierno alemán, etc., un millón de dólares es una cifra muy baja con la que puede extraer mucha información valiosa: todos estos gobiernos se pueden beneficiar de las escuchas de las comunicaciones de sus ciudadanos. Debido a que ya la NSA demostró con sus acciones que trata de invadir diariamente la privacidad de todos, es fácil concluir que este es problema es enorme; por eso, hay sobrada justificación para crear programas que permitan cifrar mensajes SMS con el fin de prevenir y neutralizar la intromisión que practican los gobiernos. Hoy en día no existe privacidad en los SMS.

La aplicación Android OS [Bort-2013] que se ha desarrollado permite enviar mensajes SMS encriptados. Ayuda a que los usuarios de teléfonos celulares recuperen su intimidad, que han perdido porque las empresas telefónicas y el gobierno puedan guardar y realizar minería de datos en cada mensaje SMS enviado por cualquier ciudadano o extranjero, sin importar su pasado o su intención.

Posteriormente se incorporará esta ampliación de manera que sea parte intrínseca del sistema operativo, de forma que cualquier mensaje pueda ser fácilmente encriptado.

El método aquí propuesto usa un archivo muy grande como llave (foto, música, vídeo, etc.), lo que permite usar una llave diferente de encriptación para cada mensaje. Por ejemplo, si la longitud promedio de los mensajes enviados es de 200 caracteres, el archivo de una canción de 5 megabytes permitiría encriptar más de 20,000 mensajes sin usar la misma secuencia de bits de encripción, de manera que el mismo mensaje sería codificado de forma diferente en cada ocasión, lo que acerca este método a ser un método de criptografía perfecto [Stinson-1995]. La fortaleza de este método está en el uso de una llave enorme cuya alta entropía garantiza que el mensaje transmitido quede siempre bien cifrado.

4 Descripción del protocolo

El protocolo para el intercambio de mensajes SMS cifrado es el siguiente:

1. Cada pareja de personas elige un archivo de alta entropía como su clave de la comunicación: la mayoría de las canciones o archivos de imágenes se pueden utilizar. Una alta entropía significa que la densidad de la información del archivo es alta. La mayoría de las canciones y fotografías se comprimen lo que significa que su entropía es suficientemente buena.

2. La clave puede ser compartida en cualquier forma adecuada, ya que un atacante puede no ahora que utilizarla para cifrar mensajes. Un archivo de imagen de Internet también puede ser utilizado, o el intercambio se puede hacer a través de "*Bluetooth*", FTP o por correo electrónico. Tanto el emisor como el receptor deben almacenar localmente esta clave de cifrado.

3. Una base de datos que contiene el número de teléfono y la ruta de acceso al archivo cifrado deberá ser conservada por la aplicación, tanto para el emisor como para el receptor. Además, una indicación de desplazamiento permitirá utilizar una parte diferente del archivo para el cifrado, lo que significa que el mismo mensaje se encripta de manera diferente al paso del tiempo.

4. Al momento de instalar la aplicación, quedará se adjunta al programa de SMS actual, y se asegurará de que cualquier mensaje enviado a cualquier número de teléfono en la base de datos de la aplicación estará cifrada. Esto automatiza el proceso de cifrado y evita los errores producidos por el manejo de mensajes por las personas.

Este protocolo es muy sencillo y permite a las partes que necesitan intercambiar mensajes cifrados utilizando una matriz de bits muy grande (la fotografía o canción) para cifrar mensajes. Como la clave de cifrado es tan grande en comparación el tamaño de cada mensaje de texto, el mismo mensaje se encripta de manera diferente muchas veces, lo que evita que el atacante pueda adquirir suficiente texto cifrado para deducir la clave de cifrado.

A pesar de que un examen detallado de la mayoría de las fotografías "*JPG*" o "*PNG*" muestra que sus bits no representan una buena secuencia aleatoria, su densidad de

información hace que sean adecuadas para ser utilizado como una clave de cifrado secreto perfecto porque el tamaño del archivo grande permite diferentes patrones de bits para ser utilizado como claves de cifrado. El algoritmo de cifrado es simple; por ejemplo:

1. Cifrar el texto SMS utilizando la cadena siguiente bits tomado del archivo cifrado (que está siempre disponible para el remitente y el receptor). El algoritmo de cifrado sigue la cadena de bits de la clave de cifrado: '1'significa cambiar el bit y '0' significa dejarlo como está. Esta cadena de bits cifrados contendrá varios caracteres no válidos ver Figura 3 (por legibilidad, en este ejemplo se muestra basura revuelta en lugar del valor XOR real).

```
                  0xF0F0F0F0F0F0
   123456789.12      123456789.12
  "Hello World!" ==> "$f&(A%;6-z)?]"
```

Figura 3

2. La cadena cifrada no se puede transmitir porque contiene muchos caracteres especiales, los que serían cambiados o eliminador por el sistema de entrega de mensajes SMS. Para evitar esta corrupción del mensaje, la cadena debe ser recodificada usando un método que la convierta en otra hilera que no contenga ninguno de esos caracteres extraños o impropios. El método de codificación Base64 convierte todos los caracteres en caracteres ASCII, números y { + / = } (en promedio, Base64 aumenta en 37 por ciento la longitud del mensaje) [Wiki-2014a].

```
  123456789.123  -->  123456789.123456789.
 "$f&(A%;6-z)?]" ==> "JGYmKEElOzYteik/XQ=="
```

Figura 4

3. Al momento de recepción, el receptor sabrá que el mensaje entrante es cifrado porque cada mensaje de este receptor y emisor siempre será cifrado, pues en la base de datos local aparece la identificación del emisor (por ejemplo, su número de teléfono). El archivo de clave acordado será usado para decodificar el mensaje, después de transformarlo de vuelta desde su representación en Base64.

4. El mensaje original será almacenado en su forma legilbe, tanto en el remitente como el final de receptor. Este protocolo está diseñado para proteger la transmisión del mensaje, no su almacenaje. Una fotografía de 2 megabytes permitirá que el emisor envíe más de 10,000 mensajes antes de que el final de la foto sea utilizada como clave de cifrado:

$$10{,}000 < (2{,}000{,}000/150)$$

Si los mensajes son más cortos de 150 caracteres, incluso más mensajes pueden enviarse. Cuando una cantidad razonable de tiempo haya pasado, lo que puede ser medido en "meses" y o en "número de mensajes enviados", la aplicación pedirá el cambio de llave, tanto del emisor como del receptor.

5 Resincronización

Habrá momentos en que emisor y receptor pierden la sincronización, pues la plataforma SMS puede retrasar, borrar o duplicar los mensajes. Se requiere un procedimiento de resincronización en estos casos. Aquí "I" inicia la resincronización y "R" responde a la misma.

1. Una operación de resincronización es iniciado por "I" enviando un mensaje que contiene sólo un caracter que no sea ni una letra ni un número. Ejemplos (en comillas) son: `'#' '+' '.' '/' '%'` (para los caracteres **`#+./%`**).

2. El programa del dispositivo "I" anexará el caracter de resincronización al principio y final de un número. Este número es el desplazamiento actual en el archivo de clave. Por ejemplo, si el desplazamiento actual del archivo de clave es 123,456,789 la cadena del mensaje que "I" enviará vía SMS es "**`+123456789+`**" (el carácter de iniciación en este caso es **`'+'`**).

3. Cuando el receptor "R" recibe la iniciativa de sincronización verifica el número de desplazamiento y lo compara con su desplazamiento actual. Si el número recibido es menor que su desplazamiento actual (`I.displacement < R.displacement`), "R" construye un nuevo mensaje que contiene las siguientes partes:

a) El carácter de iniciación utilizado para iniciar la sincronización.
b) El desplazamiento original enviado por "I".
c) El carácter de iniciación utilizado para iniciar la sincronización.
d) El nuevo valor para el desplazamiento, que siempre es más grande.
e) El carácter de iniciación utilizado para iniciar la sincronización.
f) El valor MD5 calculado para el archivo de llave [Wiki-2014b] (sirve para verificar que tanto el emisor como el receptor están usando el mismo archivo de clave).
g) El carácter de iniciación utilizado para iniciar la sincronización.

Un ejemplo de una de estas hileras de sincronización es la siguiente (el MD5 se muestra en hexagesimal):

`+1234+55555+D3226666AB00AFF0FD326677266D3FFA+`

Este mensaje se envía cifrado usando el desplazamiento originalmente enviado por "I".

`encrypt(NEW.displacement using I.displacement)`

4. Cuando el receptor "R" recibe la iniciativa de sincronización comprueba el número de desplazamiento y lo compara con su desplazamiento actual. Si el número recibido es igual o más grande, actualizar en silencio su desplazamiento y no envía nada de vuelta.
5. Inmediatamente después de iniciar la sincronización, "I" esperará recibir un mensaje de desplazamiento, pero también asumirá que su desplazamiento actual es el correcto. Si "I" recibe un nuevo desplazamiento antes de que cualquier otro mensaje SMS, se actualizará su desplazamiento actual (con el nuevo valor, que es más grande). De lo contrario, mantendrá en uso el que ya tiene.

6. El síntoma que muestra cuándo la aplicación salió de sincronización va a ser detectado cuando se produce la basura. En esta situación, el usuario sólo tiene que responder al mensaje confuso usando cualquier carácter especial. Esto iniciará el proceso de resincronización.

La base de datos de la aplicación también puede contener otra información útil. Por ejemplo, podría ser valioso almacenar un número de versión del protocolo junto al número de teléfono, lo que permitiría mejorar el protocolo si surge una nueva versión. El esquema relacional de los datos almacenados en cada base de datos local es este:

smsBD(phone, keyfile path, displacement, version)

Las mejoras a este protocolo pueden ser cualquiera de las siguientes:

- Ocultamiento de la longitud del mensaje añadiendo algunos caracteres al texto original.
- Girar el texto inicial para complicar aún más el descifrado.
- Comprobar la entropía en la siguiente cadena de bits de desplazamiento para asegurarse de que se utilizan sólo partes de alta entropía del archivo de cifrado.
- Instruir al remitente para que utilice un archivo de clave diferente cuando el desplazamiento está llegando al final del archivo de clave.
- Sugerir cambiar el archivo de clave después de algunos meses han pasado, o el lugar en que muchas operaciones de resincronización han tomado lugar.
- En lugar de Base64, se puede utilizar un método diferente para convertir el texto cifrado.
- Incluir el sumas de control ("*check-sum*") de datos para proteger el texto original.
- Permitir que un grupo de personas que comparten el mismo archivo de la clave de cifrado (esto requeriría el manejo de múltiples llaves en la base de datos).

6 Conclusiones y Trabajo Futuro

El intercambio de llaves es una de los mayores problemas que presentan los algoritmos de cifrado simétrico; además, de la mecanismo de almacenamiento que se debe dar a la llave en el emisor y el receptor: estos problemas se pueden mitigar con la idea que planteamos, ampliando la implementación a otros sistemas de intercambio de información, como podría ocurrir con mensajes de correo electrónico o con las llamadas telefónicas.

El hecho de dar al usuario el poder de encriptación dificulta la labor que realizan de manera indebida los "*hackers*", sistemas de espionaje y organizaciones a nivel mundial. El poder debe estar en las personas y no en los sistemas que los controlan.

Referencias bibliográficas

[Aguilar-2014] Aguilar, Mauricio: "Espionaje a diario extra fue ilegal". Diario Extra, Marzo 2014.
http://www.diarioextra.com/Dnew/noticiaDetalle/227937

[Bishop-2002] Bishop , Matt: "Computer Security: Art and Science". Addison Wesley, Noviembre 2002.

[Bort-2013] Bort, Julie: "Android Officially Owns More Than 80% Of The World Smartphone Market", Business Insider, 2013.
http://www.businessinsider.com/android-owns-over-80-of-world-market-2013-11

[FLKMP-2013] Farb, Michael; Lin, Yue-Hsun; Kim, Tiffany Hyun-Jin; McCune, Jonathan; Perrig, Adrian: "Safeslinger: Easy-to-use and secure public-key exchange". In Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, MobiCom '13, pp 417–428, New York, NY, USA, 2013. ACM.

[FMontano-2012] FMontano: "SMS Encrypt". Marzo 2012.
https://play.google.com/store/apps/details?id=edu.udc.smsencrypt

[Kerr-2014] Kerr, Dara: "WhatsApp hits 600 million monthly active users". C|net, Agosto 2014.
http://www.cnet.com/news/whatsapp-hits-600-million-monthly-active-users/

[KKC-2012] Kim, Sungmin; Kim, Eunhoe; Choi, Jaeyoung: A method for detecting illegally copied apk files on the network. In Proceedings of the 2012 ACM Research in Applied Computation Symposium, RACS '12, pp 253–256, New York, NY, USA, 2012. ACM.

[Lara-2014] Lara, Juan: "Ante el espionaje, el blackphone promete privacidad al usuario", la Nación, Marzo 2014.
http://www.nacion.com/tecnologia/celulares/espionaje-Blackphone-promete-privacidad-usuario_0_1402659769.html

[Moreno-2004] Moreno , Luciano: "Criptografía (iii)". HTML, Setiembre 2004.
http://usuaris.tinet.cat/acl/html_web/seguridad/cripto/cripto_3.html

[Muñoz-2009] Muñoz, Ramón: "El móvil omnipotente". El País, Diciembre 2009.
http://elpais.com/diario/2009/12/29/sociedad/1262041201_850215.html

[Müller-2014] Müller, Enrique: "Alemania pide a las embajadas que identifiquen a sus agentes secretos". El País, Agosto 2014.
http://internacional.elpais.com/internacional/2014/08/08/actualidad/1407526365_794832.html

[Ohman-2013] Ohman, Jack: "Beware of NSA watchdog" (Imagen). The Sacramento Bee, 2013.
http://blogs.sacbee.com/capitol-alert-insider-edition/2013/06/jack-ohman-beware-of-nsa-watchdog.html

[Quesada-2014] Quesada, Daniel: "Grupo extra denuncia espionaje del poder judicial a sus periodistas". Teletica, Enero 2014.
http://www.teletica.com/Noticias/39678-Grupo-Extra-denuncia-espionaje-del-Poder-Judicial-a-sus-periodistas.note.aspx

[Stinson-1995] Stinson, Douglas: "Cryptography: Theory and Practice", CRC Press, CRC Press LLC, ISBN:0849385210, 1995.

[Wiki-2014a] "Base64", Wikipedia, 2014.
http://en.wikipedia.org/wiki/Base64

[Wiki-2014b] "MD5", Wikipedia, 2014.
http://en.wikipedia.org/wiki/MD5