Correo Electrónico Seguro Empleando Infraestructura de Terceros

Juan Camilo Corena
Politécnico Grancolombiano
Email: jccorena@poligran.edu.co, investigacion@juancamilocorena.com

***Resumen*—El correo electrónico es un activo vital en las comunicaciones personales y empresariales hoy en día. Por una parte el uso de servicios web gratuitos como Gmail o Hotmail prima en usuarios personales, por el otro el uso de infraestructura propia prima en las empresas. Servicios como el de Gmail permiten el uso de correos con dominio propio empleando la infraestructura de Google, lo que permite delegar la manutención de los servicios informáticos que soportan las actividades de comunicaciones. No obstante las ventajas, los riesgos de esta modalidad en cuanto a privacidad de la información cobran relevancia significativa. Se presenta en este artículo un esquema criptográfico que permite el aprovechamiento de infraestructura de terceros para correo gratuito preservando la privacidad en la información de sus usuarios, aún ante el comprometimiento de los servidores por parte de un atacante o la entrega voluntaria de la información por parte del tercero administrador de la infraestructura.**

***Index Terms*—Correo seguro, privacidad.**

I. Introducción

Se presenta en este artículo un esquema criptográfico que garantice la privacidad del correo electrónico, aunque este se encuentre almacenado en infraestructura insegura. La motivación del artículo obedece a los grandes beneficios para los usuarios ya sea un individuo que desee mantener su correo electrónico cifrado o una compañía que desee mantener su información de negocios protegida mientras usa infraestructura provista por un tercero. Las ventajas se encuentran no solo en el ámbito de privacidad sino también en términos de transparencia en el manejo criptográfico de la información así como en el ahorro en infraestructura, costos de contratación de personal, y mayor disponibilidad de los servicios y la información para las empresas.

En años recientes la privacidad del correo electrónico ha sido producto de controversia por casos como el de la candidata vice presidencial estadounidense Sarah Palin [14] y pedidos por parte del gobierno chino a empresas del sector de internet para revelar información sobre sus usuarios [1]. Se espera que con este tipo de esquemas se logre mantener la seguridad de los usuarios a medida que tecnologías de outsourcing IT como esta y *cloud computing* ganen mayor terreno ante esquemas tradicionales.

El documento se encuentra organizado como sigue: en la sección II se definirán algunas primitivas criptográficas y la notación que se usará durante el resto del documento, luego en la sección III se dará una definición formal del problema, para luego en la sección IV mostrar la solución propuesta, en la sección V se hará una evaluación del esquema propuesto y por último en la sección VII se mostrarán algunas conclusiones y se discutirán los pasos a seguir en trabajos futuros.

II. Preliminares

Antes de continuar con el desarrollo del documento se procederá a definir la notación a usar. Como primera medida cuando se mencione el termino *seguro* en realidad se estará haciendo mención a *computacionalmente seguro* a menos que se indique lo contrario, esto para ser consecuentes con el hecho de que las funciones a usar son computacionalmente seguras, lo que quiere decir que algo es seguro ante ataques que emplean el poder de computo con limites[12].

Denotaremos el operador $[,]$ como concatenación con lo cual si se tienen dos datos a y b entonces (a, b) será igual a a seguido de b, se supondrá en todos los mensajes que a partir de la concatenación (a, b) es fácil recuperar las componentes individuales que generaron el texto concatenado sin ambigüedad, esto para no hacer énfasis en los temas de codificación del mensaje y centrar la atención en los detalles criptográficos. En una implementación práctica estos detalles no deben obviarse para el éxito en la implementación del sistema. A continuación definimos las primitivas básicas a usar durante el documento.

II-A. Función de Hash

Se asume la existencia de una función de hash $H(l) = h$ de una sola vía resistente a colisiones que asigna a entradas de longitud arbitrarias l una cadena de longitud fija h.

II-B. Función de Cifrado Simétrica

Se asume la existencia de $E_k(m) = c$ el cual es un algoritmo de cifrado simétrico E tal que para una llave k y un texto plano m produce un texto cifrado c y cuya función inversa estará dada por $E_k^{-1}(c) = m$.

II-C. Función de Cifrado Asimétrica

Se asume la existencia de $E_{P_x}(m) = c$ el cual es un algoritmo de cifrado asimétrico E tal que para una pareja de llaves asimétricas $\{P_x, S_x\}$ para el usuario x, genera un texto cifrado c y cuya función inversa estará dada por $E_{S_x}^{-1}(c) = m$. En el caso en el que E se use para firmar digitalmente usaremos la siguiente notación $E_{S_x}(h) = s$ y en el caso de la comprobación $E_{P_x}(s) = h$

II-D. MIME Extensiones multipropósito de correo de internet

Mime es un estándar de internet para transmisión de contenidos de mensajes, en formatos distintos al código *ASCII* y para representar datos que no son textos[2], en nuestro caso la funcionalidad que usaremos será la de codificar un archivo adjunto, suponemos entonces que existe la función $A(m) = x$ que transforma un conjunto de datos m en una cadena MIME compatible con SMTP y su respectiva inversa $A^{-1}(x) = m$ para revertir la transformación.

III. Descripción del Problema

El problema a tratar se define de la siguiente manera: Se tiene un sistema propiedad de un tercero para correo electrónico representado en un conjunto de servicios de comunicaciones (SMTP [11], IMAP [3], POP3 [10]) y unos servicios de almacenamiento que pueden albergar correo electrónico perteneciente a dominios de internet de los usuarios, los cuales pueden ser individuos u organizaciones que desean delegar el manejo de su infraestructura tecnológica. El servicio de correo realiza todo el proceso de recibir y enviar la información a petición de sus usuarios y lleva a cabo una política de "mejor esfuerzo"para mantener privada la información. Se desea entonces diseñar una solución que permita lo siguiente:

1. Mantener la confidencialidad de la información ante atacantes externos al tercero. Esto quiere decir que la información debe ser visible única y exclusivamente por los usuarios, nunca por alguien que ataque al tercero o con la potestad legal de pedirle a éste que ceda la información de sus usuarios.
2. Mantener la privacidad de la información ante atacantes internos. Con este requerimiento se busca que la información no sea conocida por nadie perteneciente al tercero de forma directa o indirecta. Donde directa se refiere a conocer el contenido explícito de la información e indirecta al uso que pueda darse a ésta como medio para realizar cálculo de alguna índole; lo anterior obedece al hecho de que varios proveedores de infraestructura de correo electrónico se encuentran en el negocio de motores de búsqueda.
3. Ser compatible con la infraestructura existente de correo electrónico. Por lo cual no debe existir ninguna diferencia en el procedimiento para enviar y recibir el correo electrónico por parte de los usuarios salvo la configuración inicial inherente a configurar un cliente de correo electrónico.
4. El uso de funciones criptográficas debe ser transparente para el usuario final dado que la mayoría de los usuarios no se encuentran familiarizados con el tema.
5. El sistema debe permitir movilidad entre distintos tipos de proveedores de servicios de correo web para mantener una alta disponibilidad.

IV. Esquema Propuesto

Para el sistema propuesto usaremos la siguiente notación:

- s la persona que desea enviar el correo electrónico. Esta persona puede pertenecer a cualquier dominio.
- r la persona que va a recibir el correo electrónico. Esta persona pertenece al dominio sobre el cual opera el sistema propuesto.
- p un proxy que llevará a cabo todas las tareas relacionadas con el esquema criptográfico y su interacción con los servidores de correo, este proxy pertenece al mismo dominio de r.
- $T = t_1, t_2, \ldots, t_n$ un conjunto de cuentas de correo provistas por terceros no necesariamente conocidos entre sí, que serán las encargadas de almacenar los correos pertenecientes a r, t_i puede pertenecer a cualquier dominio. Aprovechando esta notación, cuando se mencione al mensaje j dentro de la cuenta de correo t_i usaremos $t_{i,j}$.
- m mensaje de correo electrónico que s envía a r; adicionalmente denotaremos el encabezado del mensaje mediante $m.header$ y a los datos de éste con $m.data$ este último incluye la información de los archivos adjuntos también.

Dados estos actores, la única diferencia con el esquema tradicional es la aparición del proxy, el cual va a realizar todas las labores asociadas al cifrado, envío de las operaciones a las cuentas, descifrado y descarga de la información almacenada. Ahora describimos los procedimientos que se deben llevar a cabo para envío, recepción y administración de correo electrónico usando el sistema.

IV-A. Envío de correo

s realiza una petición *DNS* preguntando por los servidores *MX* del dominio al que pertenece la cuenta de r, la respuesta a esta petición es p. A continuación s envía m a p ya que s cree que p es el servidor que va a almacenar los mensajes. p Al recibir m se ejecuta los siguientes pasos:

- Encontrar P_r dentro del conjunto de parejas de llaves que p posee (una pareja por cada usuario en el dominio).
- Generar una llave de sesión k.
- Para todo $t_i \in T$ cambiar el destinatario en $m.header$ a t_i, llamaremos a este nuevo encabezado $m_i.header$ y calcular

$$m_i.data = A(E_{P_r}(k), E_k(m.data, E_{S_p}(H(m.data))))$$

por último se envía $(m_i.header, m_i.data)$ a t_i. Lo que básicamente implica transmitir el mensaje original a modo de archivo adjunto cifrado, lo que lo hace compatible con SMTP.

Cabe destacar que $m_i.data$ va firmado digitalmente con la llave privada del proxy por lo que garantizamos que el mensaje pasó por p. Esto por varias razones, primero, al usar un protocolo de envío seguro como *SMTP seguro [7]* de manera adicional a la firma digital, garantiza que la versión en texto plano de m solo es conocida por s y p; segundo, evita envíos directos hacia las cuentas de almacenamiento, lo que brinda transparencia a la ejecución del sistema y previene errores de coherencia al replicar la información.

IV-B. Recepción de correo

En el momento en que r desee leer sus correos, éste se conecta a p el cual ejecuta los siguientes pasos con todos los $t_i \in T$

- Traer el correo $t_{i,j}$ mediante algún protocolo soportado por el servidor que contiene a la cuenta t_i (IMAP,POP).
- Modificar $m_i.header$ de manera acorde para que sea recibido por r, llamaremos a este header m.header.
- Comprobar que el mensaje no haya sido extraído de un servidor $t_k \neq t_i$ en caso de haberlo sido no se retransmite.
- Obtener $m.data$ a partir de

$$A(E_{P_r}(k), E_k(m.data, E_{S_p}(H(m.data))))$$

esto es posible gracias a que p posee S_r. En caso de que la firma del mensaje sea válida se transmite $(m.header, m.data)$ a r.

El hecho de que las llaves para cada cuenta se encuentren en p, puede despertar un aire de controversia acerca de la seguridad del protocolo propuesto; sin embargo esto brinda varias ventajas como son permitir la revocación de privilegios de acceso a los usuarios en caso de ser necesario y la interoperabilidad del sistema con la base instalada, estas dos razones son más importantes para el problema planteado, que el hecho de tener un punto centralizado de falla, tema sobre el cual se proponen soluciones de implementación en la sección V.

IV-C. Autenticación ante el proxy

Para hacer posible la autenticación ante los distintos t_i, se deben poseer las contraseñas respectivas; de la misma manera debe existir una contraseña que protege a la cuenta administrada por p. Cabe destacar que las contraseñas para las cuentas de almacenamiento y esta última no son necesariamente iguales, con esto se pretende mantener el control de acceso en manos de p y no del usuario final de la cuenta, lo que permite mantener controles similares a los de un sistema de correo con infraestructura propia.

IV-D. Manejo de los mensajes en la cuenta

De la misma forma en que se realizaron los comandos para envío y recepción de correo electrónico, los mensajes para manejo de la cuenta deben incluir a p a manera de hombre en el medio para realizar la interacción con cada uno de los $t_i \in T$, estas operaciones incluyen acciones como borrar mensajes, administración de carpetas, búsquedas sobre los encabezados de los correos almacenados entre otras. Todas las operaciones que puedan realizarse sin necesidad de conocer el contenido de los mensajes pueden realizarse usando las *APIs* y mensajes correspondientes de los protocolos implementados por los servidores; sin embargo las búsquedas sobre el contenido de los mensajes son poco prácticas debido a que implican descifrar todos los mensajes para poder verificar su contenido, tal aproximación posee una complejidad $O(nm)$ donde n representa el numero de mensajes de correo y m la longitud del mensaje más largo, dado que no existe una estructura previa que organice las cadenas, algoritmos con menores complejidades como árboles de sufijos no mejoran la complejidad del algoritmo.

IV-E. Manejo de errores

Debido a la naturaleza replicada del protocolo y en particular el hecho de que cada cuenta de correo se encuentra replicada mediante servidores sin ningún tipo de relación, cada una de las operaciones es susceptible de generar problemas de coherencia, para esto definimos que el sistema se encuentra en un estado coherente cuando se cumplen las siguientes propiedades:

- Por cada correo $t_{i,j}$ existe un correo igual $t_{k,l}$ para $1 \leq i, k \leq n$ y $1 \leq j \leq |t_i|, 1 \leq l \leq |t_k|$ donde $|t_i|$ es el numero de mensajes en la cuenta $|t_i|$ e igualdad entre mensajes significa que ese par de mensajes en distintas cuentas fueron replicados a partir del mismo mensaje original.
- Si $t_{i,j} = t_{k,l}$ entonces $k = l$.

Con estas propiedades se garantizan que las operaciones de inserción, borrado y organización de mensajes generen el mismo resultado en todas las cuentas, Durante una ejecución de una operación que modifique la estructura de las cuentas de correo, p ante cualquier error en la comunicación toma como primera medida reintentar la operación, en caso de no poder inmediatamente, posterga dicha operación hasta que todas las cuentas posean la información al día. Para evitar el crecimiento en la complejidad del proxy la información de los estados parciales de las operaciones será la que se encuentra almacenada en las cuentas por ejemplo, dado el caso en que existan servidores cuya fiabilidad no sea muy alta y las operaciones de borrado no siempre puedan llevarse a cabo, mientras exista una copia de un mensaje en alguna cuenta de correo t_i el mensaje se tomara como no borrado, por otra parte si solo una cuenta logró recibir el mensaje este mensaje se tomara como presente; ciertamente los dos estados descritos son incoherentes bajo nuestras propiedades por lo que p debe replicar los mensajes no presentes en todas las cuentas hasta lograr las propiedades de coherencia deseadas.

V. Análisis de seguridad de la solución

El análisis se llevará a cabo en términos de lo planteado en la sección III. En cuanto a las propiedades 1 y 2 que se refieren a la confidencialidad de la información ante atacantes internos y externos respectivamente, el sistema protege los datos usando algoritmos de cifrado que mantienen secretos los datos sobre los que fueron aplicados siempre y cuando se tomen las precauciones necesarias al generar las llaves criptográficas, lo que como resultado da la propiedad de *forward-secrecy* al sistema. En el caso de un atacante activo, el sistema propuesto depende del uso de un protocolo seguro de envío de correo electrónico para garantizar que la comunicación entre s y p se mantenga secreta, esto se hizo así ya que protocolos que cumplen el objetivo de proteger un canal de comunicaciones han sido diseñados anteriormente [4].

Con respecto a la propiedad 3 que se refiere a compatibilidad, el sistema no cambia la esencia de los protocolos para envío, recepción y administración de correo por lo que es compatible con la base existente; en cuanto a 4 que se refiere a transparencia para el usuario final, éste jamás se entera de que su correo ha sido cifrado y replicado en distintos servidores

alrededor de la red, por último el administrador de p puede cambiar sus proveedores de servicio con lo que se da por satisfecho el requisito 5.

Con respecto a detalles de implementación, el primer punto de ataque al sistema es el proxy dada su naturaleza centralizada, los datos relevantes que este posee son:

- Parejas de llaves $\{P_x, S_x\}$ para todos los usuarios. Esto presenta varios retos, el primero es mantener replicadas las llaves criptográficas sin comprometerlas, esto se logra mediante un esquema de umbral [13], por lo que no representa un problema relevante mantener una copia segura en caso de que estas se pierdan; por otra parte el problema de delegar la firma y cifrado de la llave de sesión k a una sola entidad dentro del sistema es peligroso ya que al romperse la seguridad de p todos las llaves quedarían comprometidas, una manera de lidiar con este problema es empleando funciones criptográficas de umbral [8], en las cuales los cálculos de las funciones criptográficas se realizan de manera compartida, esquemas de este tipo existen para algoritmos como ElGamal y RSA.
- Direcciones reales donde se almacena la información. Esta información aunque se mantiene secreta en p su conocimiento no representa ningún riesgo para el usuario, porque en primer lugar no habría sistema de correo electrónico seguro si solo el nombre de la cuenta de correo fuese suficiente para lograr un ataque exitoso, por lo que se puede realizar replicación exhaustiva para mantenerla disponible.

Tomando en cuenta las soluciones a los problemas anteriores vemos que el proxy es altamente replicable y por lo tanto no constituye un punto único de falla, por lo tanto no afecta la seguridad del sistema propuesto en términos prácticos.

Para culminar, vale la pena mencionar que la firma digital insertada por el proxy a cada mensaje, garantiza que el tercero no pueda modificar el contenido de los mensajes sin que el usuario se entere por lo cual también se garantiza la integridad de los mensajes, este mismo razonamiento aplica para un ataque de hombre en el medio durante la etapa de transmisión desde p hacia los t_i.

VI. Compatibilidad con Servicios Existentes

Dadas las características técnicas del sistema propuesto, su implementación no representa ningún cambio para los proveedores del servicio a nivel de infraestructura. Sin embargo las políticas de uso de estos sistemas podrían representar un impedimento por lo cual se consideró relevante leer detenidamente las políticas de uso de dos de los sistemas de correo más populares: Hotmail [9] y Gmail [5] . En ambos proveedores los términos de uso no hacen referencia explícita a la prohibición del uso de criptografía en los archivos adjuntos, sin embargo son bastante enfáticos en resaltar en que hacen lo posible para mantener confidencial la información de sus clientes, pero también en que pueden acceder a dar los datos antes pedidos de terceros que sean justificados como se muestra en la sección de preguntas frecuentes del servicio de Google [6]: *How does Google handle law enforcement requests? Google complies with valid legal process. It is Google's policy to notify users before turning over their data whenever possible and legally permissible.*. Otro punto importante a resaltar es el hecho de que los proveedores se reservan el derecho de cambiar sus políticas de uso y privacidad de la información en el momento en que lo crean conveniente por lo cual vale la pena emplear los mecanismos de duplicación de datos del sistema para evitar contratiempos en la prestación del servicio. Para corroborar empíricamente que no hubiese restricciones al envío de archivos adjuntos cifrados, se realizó una prueba que consistió en enviar un archivo cifrado a 3 cuentas de correo:

- Cuenta Gmail gratuita.
- Cuenta Hotmail gratuita.
- Cuenta perteneciente a un dominio externo a Gmail pero administrado por este ultimo

Para el cifrado se usó AES en modo CBC con una llave de 128 bits, adicionalmente el archivo fue codificado en Base 64; este proceso se realizó en Java 1.6 usando el proveedor criptográfico por defecto. Los resultados del experimento fueron alentadores ya que ninguna de las cuentas de correo presentó contratiempo alguno para recibir y mostrar la información asociada a los archivos adjuntos cifrados. Por lo que de implementarse el sistema podría usarse con estos servicios.

VII. Conclusiones y Trabajo Futuro

Se ha presentado un esquema para uso de seguro de correo electrónico que salvaguarda la confidencialidad, integridad y disponibilidad de la información incluso sobre infraestructura provista por terceros. Adicionalmente el sistema es compatible con la base instalada para correos electrónicos.

En general el sistema presenta grandes ventajas para el usuario sin afectar de manera significativa la funcionalidad a la que se encuentra acostumbrado. El siguiente paso consiste en realizar una implementación funcional del esquema propuesto y verificar de manera experimental el rendimiento del sistema que puede verse afectado por su estructura y demoras en la conexión con distintos proveedores del servicio de correo electrónico.

Referencias

[1] *Yahoo 'helped jail China writer'*, BBC News, Septiembre 2005. [En linea]. Disponible: http://news.bbc.co.uk/2/hi/asia-pacific/4221538.stm [Accedida: 20 de Marzo 2009]

[2] N. Borenstein, N. Freed. *RFC 1521 - MIME (Multipurpose Internet Mail Extensions)*. 1993 [En línea]. Disponible: http://www.faqs.org/rfcs/rfc1521.html [Accedida: 20 de Marzo de 2009].

[3] M. Crispin. *RFC 3501: Internet Message Access Protocol - Version 4rev1* 2003. [En línea]. Disponible: http://www.faqs.org/rfcs/rfc3501.html [Accedida: 20 de Marzo de 2009].

[4] T. Dierks. *RFC 2246 - The TLS Protocol Version 1.0*. 1999 [En línea]. Disponible: http://www.ietf.org/rfc/rfc2246.txt [Accedida: 20 de Marzo de 2009].

[5] Google. *GOOGLE APPS PREMIER EDITION AGREEMENT*. [En línea]. Disponible: http://www.google.com/apps/intl/en/terms/premier_terms.html [Accedida: 25 de Mayo de 2009].

[6] Google. *How does Google handle law enforcement requests?* [En línea]. Disponible: http://www.google.com/support/a/bin/answer.py?answer=107818 [Accedida: 25 de Mayo de 2009].

[7] P. Hoffman. *RFC2487 - SMTP Service Extension for Secure SMTP over TLS*. 1999 [En línea]. Disponible: http://www.faqs.org/rfcs/rfc2487.html [Accedida: 20 de Marzo de 2009].
[8] H.Lipmaa. *Cryptography and Data Security, 24.03.2004 Lecture 9: Secret Sharing, Threshold Cryptography*. 2004
[9] Microsoft. *Microsoft Service Agreement Last Updated: April 2009*. [En línea]. Disponible: http://help.live.com/help.aspx?project=tou&mkt=en-us [Accedida: 25 de Mayo de 2009].
[10] J. Myers, M. Rose . *RFC 1939: Post Office Protocol - Versión 3*. 1996 [En línea]. Disponible: http://www.rfc-es.org/rfc/rfc1939-es.txt [Accedida: 20 de Marzo de 2009].
[11] J B. Postel. *RFC 821: Simple Mail Transfer Protocol*. 1982 [En línea]. Disponible: http://james.apache.org/server/rfclist/smtp/rfc0821.txt [Accedida: 20 de Marzo de 2009].
[12] A. Russell, H. Wang. *How to fool an unbounded adversary with a short key*. Eurocrypt 2002.
[13] A. Shamir: *How to Share a Secret.*
[14] M.J. Stephey. *Sarah Palin's E-Mail Hacked*,Time magazine, Septiembre 2008. [En línea]. Disponible: http://www.time.com/time/politics/article/0,8599,1842097,00.html [Accedida: 20 de Marzo de 2009].