Protocolo para la certificación de llaves públicas para OpenPGP libre del envenenamiento de certificados

Gunnar Wolf
Instituto de Investigaciones Económicas, UNAM
Facultad de Ingeniería, UNAM
Ciudad de México, México
gwolf@gwolf.org

Jorge Luis Ortega Arjona
Facultad de Ciencas, UNAM
Ciudad de México, México
jloa@ciencias.unam.mx

***Abstract*—Los mecanismos de cifrado basados en llave pública en Internet deben ir de la mano de un modelo de confianza para la validación de identidades. Si bien el modelo más utilizado en Internet es el de Infraestructura de Llave Pública con Autoridades Certificadoras (PKI-CA), es un modelo absolutamente centralizado. Para los casos de uso en los que esto es indeseable, es posible también utilizar un modelo completamente distribuido: la malla de confianza (Web of Trust, WoT). El sistema más ampliamente conocido que implementa un modelo WoT es OpenPGP, derivado de PGP, que fue publicado inicialmente en 1991. Para ser de utilidad para un grupo geográficamente disperso de participantes, la WoT requiere de una *red de servidores de llaves* para la búsqueda y descubrimiento de llaves — que, a su vez, opere también de forma libre de centralización.**

En el último decenio han aparecido diversas vulnerabilidades de alto perfil sobre la red de servidores de llaves. Estas vulnerabilidades no únicamente afectan a las implementaciones específicas, sino que demuestran una distancia cada vez mayor entre las suposiciones y modelos de amenaza iniciales, que llevaron a la implementación de protocolos posiblemente ya no aptos para la realidad de la red. Como consecuencia, la red de servidores de llaves ha sufrido una drástica reducción en su número de nodos, y puede afirmarse que se enfrenta con una crisis existencial.

Este trabajo delinea la un trabajo en proceso, enfocado a resolver esta problemática, a partir de la necesidad de mantener una solución para la viabilidad de un modelo de confianza transitiva libre de centralización, partiendo de una propuesta que modifica los criterios de admisión para los nuevos certificados de las llaves.

***Keywords*— Modelos de confianza transitiva, Malla de confianza, OpenPGP, Modelo distribuido, Identidad**

I. Introducción

El estándar OpenPGP [3] es conocido principalmente por ser el principal mecanismo de cifrado de correo electrónico en uso. Para poder hacerlo, debe ir de la mano con el uso de capacidades de distribuidas de autenticación de identidad y de confianza transitiva, conocido como *malla de confianza* (*Web of Trust*, WoT; véase la Sección II). La WoT requiere que la información de identidad se haga disponible mediante una red de *servidores de llaves*. La red de servidores de llaves está diseñada para operar de forma distribuida; los servidores evitan la centralización usando un protocolo *de chisme* (*gossip*) epidémico para sincronizar sus respectivas bases de datos de llaves públicas.

La red de servidores de llaves, y por tanto, toda la infraestructura para la distribución de llaves públicas, está en peligro existencial debido a una serie de debilidades y ataques (particularmente el ataque denominado *envenenamiento de certificados*; véase la Sección III para una discusión general y, en particular, la Subsección III-D respecto al envenenamiento de certificados). Distintas implementaciones se han enfocado en alternativas para la distribución de llaves, pero al hacerlo, caen a un modelo que requiere mayor centralización, o debilitan el modelo de confianza transitiva WoT.

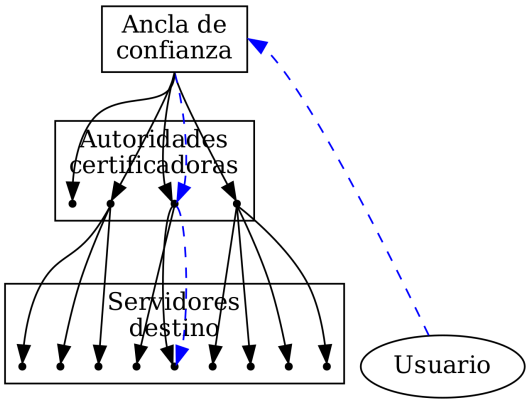
Este trabajo presenta un protocolo para la certificación de llaves que contrarresta al ataque de envenenamiento de certificados, requiriendo que todas las modificaciones a una llave pública sean *atestiguadas* por la llave destino.

II. Modelos de confianza transitiva

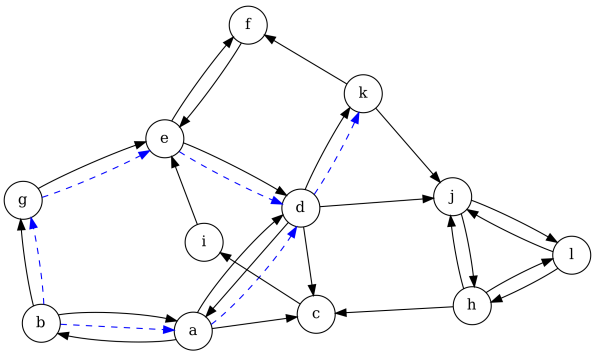
Los *modelos de confianza* permiten al usuario estar seguro acerca de la *identidad* de la contraparte en una comunicación cifrada. Los modelos de confianza más comunes son *transitivos*, esto es, un equipo A confía *directamente* en un grupo reducido de identidades TA_1, TA_2, TA_3, las cuales pueden *certificar* a otras.

La mayor parte del tráfico cifrado de Internet utiliza al protocolo TLS (Seguridad en Capa de Transporte, Transport Layer Security). Este protocolo, así como su predecesor, SSL (Capa Segura de Conexiones, Secure Socket Layer) basan la autenticación en el modelo PKI-CA (Infraestructura de Llave Pública con Autoridades Certificadoras, Public Key Infrastructure — Certification Authority, véase la Figura 1(a)). En este modelo, los usuarios confían plenamente en un conjunto de servidores llamados *anclas de confianza* (*Trust Anchors*). Estos servidores *delegan* su capacidad de certificación a las *Autoridades Certificadoras*, las cuales emiten los certificados de identidad para los servidores. Siguiendo la estructura presentada en la Figura 1(a), cuando un usuario establece comunicación con un servidor W, éste le envía una *cadena de certificados* que incluye las siguientes aseveraciones:

978-1-6654-8014-7/22/$31.00



(a) Modelo centralizado: PKI-CA. El usuario verifica que haya un camino válido desde una *raiz de confianza* hasta el servidor destino.



(b) Modelo distribuido: WoT. El usuario *b* construye caminos de confianza hacia el objetivo *k*, siguiendo los vértices existentes en el grafo.

Fig. 1. TTMs. Black lines denote all trust relationships in the network, and blue dashed lines mean trust paths followed from a user to their target.

$$TA_2 \rightarrow CA_3; CA_3 \rightarrow W$$

El modelo PKI-CA es adecuado para una muy gran cantidad de casos de uso, de lo cual es prueba su amplio uso. Hay, sin embargo, un modelo de distribución de confianza distinto, basado en un modelo distribuido y sin autoridades centrales designadas: La *Malla de Confianza* (*Web of Trust*, WoT). Para este modelo, en vez de que todos los usuarios de una red confíen en un conjunto predefinido de raíces de confianza, cada uno de los usuarios centra la confianza *en sí mismo*: las relaciones de confianza se expresan como la malla presentada en la Figura 1(b). Para este caso, si el usuario *Beto* (b) busca establecer comunicación segura verificando la identidad de *Karen* (k), debe construir *caminos de confianza* tales que permitan $b \rightarrow k$. Recorriendo acíclicamente el grafo presentado, se presentan los dos siguientes caminos:

$$b \rightarrow a \rightarrow d \rightarrow k$$

$$b \rightarrow g \rightarrow e \rightarrow d \rightarrow k$$

La implementación más conocida de WoT es OpenPGP [3]. Este protocolo es principalmente utilizado para comunicación *asíncrona*: cifrado de correos electrónicos, firma de documentos (frecuentemente para su autenticación ante posteriores descargas por terceros en Internet), y respaldo de documentos; por tal motivo, encontrar un camino de confianza entre b y k requiere de infraestructura adicional: al ser asíncronos, no se realiza una primer comunicación de ida y vuelta entre ellos; para no requerir que b envíe un mensaje sin cifrar para inicializar una comunicación (lo cual podría acarrear demoras impredecibles) se requiere de un mecanismo para la *distribución de llaves públicas*. Para OpenPGP, se cuenta con los *servidores de llaves* [7, 15]. Los servidores de llaves independientes se sincronizan entre sí utilizando un protocolo *chisme* o *epidémico* de reconciliación entre conjuntos grandes [12]; esta sincronización permite que, si un usuario envía su llave pública a cualquier servidor que participe en dicha red, ésta llegará a todos los demás servidores en poco tiempo.

III. Debilidades y ataques de la malla de confianza (WoT)

Se han presentado debilidades en el esquema descrito de la WoT a muy distintos niveles; en la presente sección abordaremos algunas vulnerabilidades que no son específicas a la implementación, y por tanto no pueden ser corregidas mediante una corrección al software (popularmente conocido como un *parche*), sino que vulnerabilidades de las premisas base y protocolos sobre los cuales se construyó el ecosistema OpenPGP. Por tal motivo, estas vulnerabilidades son más difíciles de corregir: no basta con "sencillamente" corregir un error en el software, sino que hace falta adoptar una visión más amplia. Además, dado que la interacción entre las siguientes vulnerabilidades constituye un problema mucho más grave de lo que puede parecer si se les analiza por separado.

A. Falta de uso y comprensión del modelo

El modelo WoT asume que los usuarios de OpenPGP se tomarán la molestia de vincular sus llaves públicas (y, por tanto, validar sus identidades) hacia el *conjunto fuerte* de la malla (el conjunto de llaves más grande interconectado bidireccionalmente por certificaciones).

Sin embargo, a lo largo de los años, los usuarios han adoptado el uso de las diversas implementaciones de OpenPGP principalmente por su capacidad de realizar cifrado fuerte, buscando la posibilidad de ocultar el contenido de sus comunicaciones de atacantes muchas vces identificados con los proveedores de comunicaciones o los gobiernos nacionales, como puede verse en guías de usuarios tanto históricas como actuales [6, 13]; puede incluso encontrarse esta orientación en guías corporativas acerca de este tema [5]:

> PGP se usa sobre todo para cifrar mensajes de correo. Inicialmente fue utilizado por personas interesadas en compartir información sensible, como activistas y periodistas. Pero su popularidad se ha incrementado significativamente al revelarse la recolección de información personal que realizan organizaciones y agencias de gobierno, conforme la gente busca mantener privada su información personal sensible.

Sin embargo, como muchas veces esto se presenta como una característica para "usuarios avanzados", muchos usuarios no aprovechan –ni siquiera consideran relevante– el uso de la capacidad de autenticación que brinda la WoT. Esto ha llevado a que una amplia mayoría de las llaves que forman parte de los servidores no estén vinculados con la WoT: De 5 217 474 llaves que conformaban la base de datos de los servidores de llaves en 2020, 84% están aisladas (no cuentan con ninguna certificación que permita a algún usuario establecer la confianza en sus comunicaciones). 8.14% de las llaves tienen una única certificación, 2.53% tienen dos, y únicamente 1.71% de las llaves cuentan con 10 o más certificaciones. Del conjunto de llaves que sí han sido certificadas, únicamente del orden de 60 000 conforman el *conjunto fuerte*. Esto significa que, para fines prácticos, la WoT resulta útil apenas para cerca del 1% de las llaves que conforman la red [21].

B. Inmutabilidad de información no autenticada

El modelo de amenazas al que responde el diseño de la red de servidores de llaves está explícitamente diseñado para resistir a la interferencia o censura de gobiernos nacionales. Los datos que se envían a los servidores de llaves no se autentica (cualquiera puede subir datos OpenPGP arbitrarios) [20].

Dado que la red de servidores de llaves está construida sobre los protocolos *chisme* de sincronización ya mencionados, la red aceptará tanto nuevas llaves como actualizaciones de llaves existentes, y una vez que la información sea recibida y comience a esparcirse a otros servidores, para fines prácticos no puede ser ya eliminada: cualquier servidor que tenga un paquete de datos que no tenga alguno de sus pares lo inyectará de vuelta a la red cuando se vuelva a sincronizar. El protocolo es, pues, *eficientemente descentralizado*, dado que no hay ninguna instancia de coordinación central entre los servidores de llaves [4].

Dado que la resistencia a la censura fue un criterio de diseño, el protocolo no provee ninguna provisión para eliminar información no deseada. Este problema ha traído problemas legales a algunos operadores de servidores de llaves, que no pueden cumplir con las órdenes de remoción de material derivadas de las leyes de privacidad más nuevas que el diseño de este mecanismo [14]. La adopción de la Regulación General de Protección de Datos (GDPR) por la Unión Europea en 2016 eleva la relevancia de este asunto, y ha llevado a que muchos operadores de servidores de llave dejen de ofrecer sus servicios [14, 18]

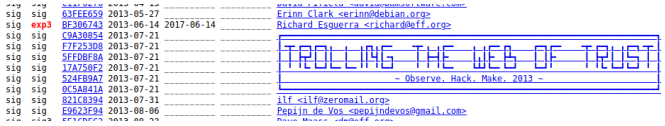


Fig. 2. Ejemplo del ataque *Trolling the Web of Trust*: Cuando un servidor de llaves muestra la información de la WoT para la llave afectada, cada línea incluye la identidad del firmante. Creando *identidades descartables*, Lee usa a la WoT como un *lienzo para graffiti* [10]

C. Uso de la WoT como un lienzo para graffiti

El formato estándar de mensajes de OpenPGP incluye diversos campos visibles al usuario, estructurados en *paquetes*. Uno de estos campos es el identificador de usuario, que "consiste de texto UTF-8, cuya intención es representar el nombre y dirección de correo electrónico del dueño de la llave. Por convención, está compuesto por nombre y dirección de correo siguiendo el RFC 2822, pero no hay restricciones respecto a su contenido" [3, p. 48]. Dado que no se impone ningún formato sobre dichas cadenas, Micah F. Lee presentó en el encuentro *Observe. Hack. Make.* –con la intención de que se viera como una *broma divertida* sobre del protocolo– una manera de crear y subir a los servidores masivamente conjuntos de llaves sin uso ni verdadero significado, y emplearlas para certificar a otras llaves, utilizando a la WoT como un *lienzo para graffiti*, como lo muestra la Figura 2 [10].

Puede sonar excesivo llamar a esta broma un ataque. El impacto sobre la WoT es de un incremento marginal en el tamaño, y el crecimiento de un poco de ruido. Sin embargo, esto trae a la luz una importante omisión en el diseño de la WoT de OpenPGP, que demostró a la postre ser de una mucho mayor severidad.

Un ataque derivado del anterior es el abuso de la red de servidores de llaves para la *distribución de contenido arbitrario*: un programa que permite codificar información arbitraria, haciéndola pasar por una llave y su material de certificación relacionado, permitiendo abusar de la red de servidores públicos de llaves para el almacenamiento de datos arbitrarios [19, 20]. Dado que, como se describió en la Sección anterior, la WoT es para fines prácticos un *medio distribuido, anónimo, que únicamente permite el agregado, y sin provisión para la eliminación de información*, este abuso hace efectivamente ilegal operar nodos de servidor de llaves, dado que tras la entrada en vigor de la ley de privacidad GDPR, un agente del gobierno puede exigir que los operadores de un sitio retiren de la vista pública determinada información — y no hay manera de cumplir con dicha solicitud [19].

D. Envenenamiento de certificados

La conjunción de varias de las debilidades mencionadas lleva a un ataque en toda forma: el envenenamiento de certificados.

Si *Alicia* tiene la llave k_A y quiere comunicarse con el usuario *Beto*, que tiene la llave k_B y la cadena de certificados C_{k_B}, *Alicia* se conecta a un servidor de llaves públicas y busca por *bob@example.org*. Verifica los resultados, e importa la

llave a su llavero local. *Alicia* verifica cuidadosamente para asegurarse de que k_B está certificado por sus amigos mutuos, *Carlos* (C) y *Diana* (D). *Las llaves OpenPGP llevan también una /auto-certificación*, una firma del mismo usuario que incluye información como el periodo de validez de la llave. *Alicia* se presenta ante *Beto*. Hasta este momento tenemos:

$$C_{k_B} = k_B, cert_{k_B \to k_B}, cert_{k_C \to k_B}, cert_{k_D \to k_B}$$

Pero *Malva* quiere interponerse en la comunicación entre ellos, así que crea miles de llaves descartables, sin información significativa — probablemente, sin una cadena válida de identidad siquiera, y que no puedan ser rastreadas de vuelta hacia *Malva*. Entonces, *Malva* controla:

$$k_{M_1}, k_{M_2}, k_{M_3}, \ldots, k_{M_{9\,999}}, k_{M_{10\,000}}$$

Malva certifica la llave pública de *Alicia*, k_A, con todas sus llaves descartables, y sube el resultado a la red de servidores de llaves. Un usuario que busque a k_A ahora recibirá un resultado substancialmente más grande:

$$C_{k_A} = \begin{cases} k_A, certk_A \to k_A, cert_{k_C \to k_A}, cert_{k_D \to k_A}, \\ cert_{k_{M_1}} \to k_A, cert_{k_{M_2}} \to k_A, cert_{k_{M_3}} \to k_A, \\ \ldots \\ cert_{k_{M_{9\,999}}} \to k_A, cert_{k_{M_{10\,000}}} \to k_A \end{cases}$$

Las llaves públicas de OpenPGP típicamente miden del orden de algunos kilobytes; algunas llaves muy bien conectadas (esto es, que están certificadas por muchos otros usuarios) pueden llegar a algunos cientos de kilobytes. Pero después del ataque de *Malva*, k_A mide decenas o cientos de megabytes, y está *envenenada* — es inutilizable. Cuando *Beto* intenta obtener k_A desde el servidor de llaves para iniciar la comunicación con *Alicia*, su cliente OpenPGP recibirá información órdenes de magnitud mayor al que está diseñado para manejar. Se ha observado que las fallas derivadas de esta situación incluyen congelamiento del programa, e incluso la corrupción del *llavero local* de Beto.

En este punto, la llave de *Alicia* ya no puede ser utilizada, y ella tendrá que migrar a una nueva k'_A. Para hacerlo, tendrá que encontrarse cara a cara con otros participantes del llavero de confianza, solicitando que le firmen la nueva llave, para volver a vincularse al *conjunto fuerte*. Sin embargo, *Malva* puede fácilmente repetir su ataque.

Si bien no se han documentado muchos ataques como el descrito [8], son un riesgo que amenaza a todos los usuarios de la comunidad OpenPGP, y son parte de la razón por la cual se percive que la comunidad de servidores de llave pública está muriendo [11].

IV. Protocolo para la certificación de llaves

La certificación de llaves entre usuarios requiere que estos, como primer paso, realicen una verificación de identidad *fuera de banda*, en la cual intercambien las *huellas digitales* de sus llaves públicas para poder dar fé cada cual de la veracidad de la

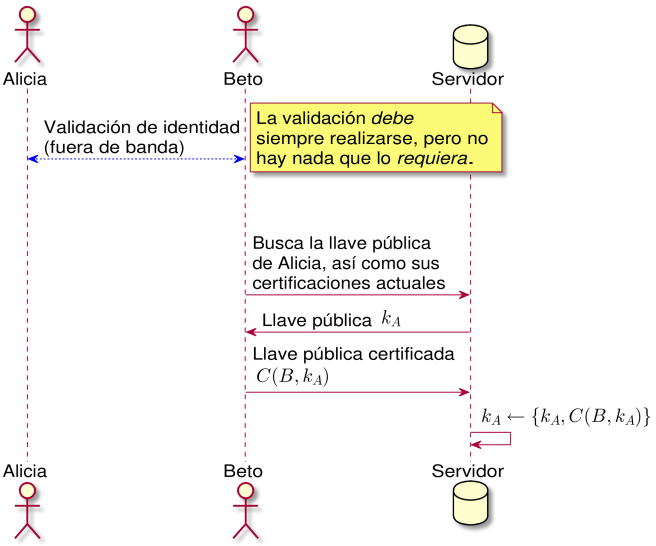


Fig. 3. Diagrama de secuencia que presenta cómo *Beto* certifica la llave pública de *Alicia* en la red de servidores de llaves preexistente; no se requiere de ninguna acción por parte de *Alicia* para agregar nuevos certificados a su llave.

identidad de su contraparte. Este proceso se ilustra en la Figura 3: Para *Beto* poder certificar la llave pública de *Alicia*, ambos deberían haberse encontrado, pero no hay manera en que un servidor de llaves imponga esta condición — esto significa que el servidor de llaves no requiere siquiera que *Alicia* esté al tanto de que se agregan certificaciones a su llave. ¡Este es precisamente el problema que permite el envenenamiento de certificados!

Nuestra propuesta modifica la lógica del servidor de llaves, de forma que cualquier modificación sobre la llave de *Alicia* debe ser aprobada por ella, como se ilustra en la Figura 4 shows. Cualquier paquete que modifica a la llave de *Alicia* debe estar *atestiguado* (firmado, certificado) por la misma llave de *Alicia*.

La lógica aquí descrita ha sido denominada *Certificaciones de tercera persona atestiguadas por primera persona* (*First-party-attested third-party certifications*, 1PA3PC), aunque no se ha propuesto aún ninguna implementación. Comprender por qué 1PA3PC hace imposible al ataque de envenenamiento de certificados resulta fácil de entender: volviendo al ejemplo presentado en la Subsección III-D, incluso si *Alicia* no fue cuidadosa y atestigua algunos de los certificados espurios craedos por alguna de las llaves descartables controladas por *Malva*, no aceptará decenas de miles de solicitudes para atestiguar nuevas firmas sobre su llave. Los servidores de llaves, por tanto, no distribuirán la mayor parte de los certificados $cert_{k_{M_1}}$ a $cert_{k_{M_{10\,000}}}$, y la llave de *Alicia* se mandendrá utilizable.

Debemos hacer notar que 1PA3PC no previene el uso *legí-*

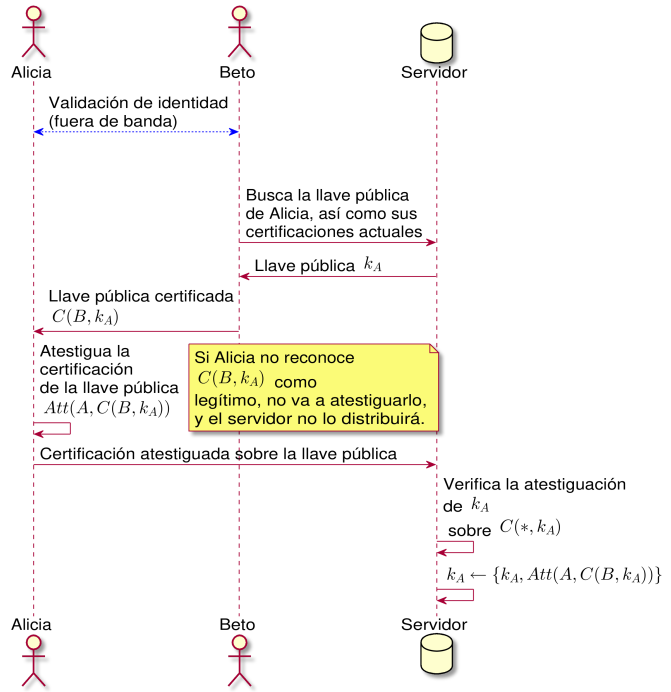


Fig. 4. Diagrama de secuencia para el protocolo propuesto, certificación de llaves atestiguada por la primera persona.

timo de la WoT como un *lienzo para graffiti*[1], como lo describe la Subsección III-C: Si *Alicia* quiere "decorar" su listado en la WoT con certificaciones provenientes de identidades descartables sin significados, ella podrá atestiguarlas, y los servidores de llaves los aceptarán.

A. Retos a la aplicabilidad

Los servidores OpenPGP aceptan todos los paquetes para agregar a llaves existentes sin requerir autenticación, independientemente de su proveniencia. Esto se explica parcialmente por la problemática presentada en III-A: El uso adecuado de la WoT *requiere de esfuerzo* por parte de los usuarios casuales; el modelo de operación delos servidores de llaves existentes actuales se conformó a partir de elegir la barrera más baja posible para la participación. Aún así, la participación en la WoT apenas alcanza al 16% del total de llaves, con apenas sobre 1% del total perteneciendo al conjunto fuerte. Reconocemos que agregar requisitos y pasos a la lógica actual probablemente lleve a una todavía menor adopción del modelo de confianza transitiva.

Un servidor de llaves que implemente en el protocolo aquí descrito no podrá participar en la red de servidores de llaves preexistente, pues la divergencia entre su base de datos y las del resto de los servidores en la red crecerá rápidamente, al presentar distintas políticas de admisibilidad para el material criptográfico. Puede realizarse una importación inicial de los datos exportados por los servidores de llaves, filtrando y excluyendo todo el material que resulte en firmas no recíprocas, pero si se une al intercambio de información por *chisme* probablemente falle. Por tanto, puede inicializarse una red de servidores de llaves públicas paralelo, utilizando como *semilla* al material existente en la red actual, pero rechazando todo el material que no cumpla con las políticas de la nueva red.

[1]Podría argumentarse que los *lienzos de graffiti* son necesariamente un abuso, dado que utilizan a la red de servidores de llaves de una manera distinta a como fue diseñada para operar, y la cargan con información inútil. Empleamos el calificativo *legítimo* dado que su impacto sobre la red es mínimo, y su afectación se limita a un renglón en el listado de las llaves atestiguadas firmadas por dichas identidades descartables.

V. Trabajo relacionado

La comunidad de usuarios de OpenPGP no se ha quedado de brazos cruzados ante los ataques y vulnerabilidades anteriormente mencionados. Hay diversas estrategias que pueden adoptarse hoy en día contra ellos, pero la mayor parte de las propuestas lo hacen comprometiéndose en la propiedad distribuida del modelo WoT. Esta Sección presenta algunas de dichas estrategias.

A. Mecanismos diferentes para el descubrimiento de llaves

Las siguientes estrategias identifican a la red de servidores de llaves como el eslabón más débil en la cadena, como la pieza que permite los ataques, y presentan diferentes maneras en que los usuarios pueden descubrir y asignar confianza a las llaves públicas de terceros.

1) Autenticación basada en DNS de entidades nominadas (DANE): Dado que las llaves se utilizan típicamente en relación estrecha con la dirección de correo electrónico del usuario, el protocolo DANE (DNS-based Authentication of Named Entities) codifica la llave pública de los usuarios como la respuesta extendida a una solicitud a un campo especial de DNS sobre el dominio del servidor de correo [17]. DANE requiere, sin embargo, que el proveedor de correos esté dispuesto a prestar esta funcionalidad centralizada, no sólo convirtiéndose en un punto único de falla, sino que también requiriendo que el dueño del dominio tenga interés en proveerlo; hoy en día, dada la cantidad de usuarios de correo electrónico empleando proveedores masivos (como GMail de Google), una gran proporción de los usuarios de correo efectivamente están imposibilitados para poder utilizar este mecanismo.

2) Directorio de llaves sobre Web (WKD): WKD (Web Key Directory) es un mecanismo alterno para hacer disponibles las llaves de los usuarios de un proveedor de correo bajo determinado dominio [9]. WKD presenta una menor *fricción* ante su adopción que DANE, dado que las páginas Web son más fáciles de administrar que los registros extendidos de zona DNS (y los proveedores están más dispuestos a delegar la administración de páginas específicas a su personal con menos centralidad técnica), pero sigue siendo un esquema altamente centralizado, y sigue dejando a los usuarios a expensas de la voluntad de los administradores para distribuir sus identidades.

3) Confiar al Primer Uso (TOFU): Partiendo de la baja adopción de la WoT presentado en la Subsección III-A, los impulsores de TOFU (Trust On First Use; también se le conoce como un esquema de distribución de confianza *Leap of Faith*,

de *Salto de fé*) sostienen que la WoT es *teóricamente* fuerte, pero *prácticamente* no resulta útil [16]. Para distribuir la llave pública, cada usuario la adjunta como un encabezado a todos sus mensajes de correo. Por razones de espacio,la llave se distribuye en su formato *minimizado*, sin certificaciones. El cliente de correo de cada usuarios se encarga de crear la lista de destinatarios cifrados posible a partir de las llaves que vaya encontrando en sus interacciones..

Los usuarios que quieran iniciar una conversación cifrada y hayan estado ya en contacto pueden comenzar a intercambiar mensajes cifrados, casi transparentemente. Si bien la primera comunicación sí se presenta vulnerable al engaño con una identidad falsa, sostienen, es altamente improbable que un atacante intercepte la primera comunicación entre dos individuos — el valor de la comunicación cifrada se presenta en mensajes subsecuentes, y crece conforme estos aumentan. La confianza en la identidad crecerá también del uso repetido del cifrado.

B. Esquemas diferentes de sincronización entre servidores

Si se reconoce que parte importante del problema es la imposibilidad de eliminar –o por lo menos dejar de entregar ante consultas– las llaves o certificados que sean necesarios puede deberse al uso de un protocolo *chisme* para la sincronización, las siguientes propuestas presentan diferentes esquemas de sincronización:

1) BlockPGP: Emplea una cadena de bloques (*blockchain*) derivada de Ethereum para la representación de cambios en las bases de datos de los servidores de llaves [21]. Esta solución presenta mejoría en el tiempo requerido para la propagación de las llaves y certificados nuevos, y una fuerte mejoría para los servidores de llaves que se han atrasado en su sincronización.

El modelo presentado por BlockPGP sigue manteniendo la operación general de la red de servidores descentralizada, sin embargo, introduce el papel de una *cuenta administrador* con capacidad de eliminar información tóxica de la cadena de bloques; si bien esto potencialmente resuelve la fricción con la GDPR y otras legislaciones similares, puede percibirse como antitético a la filosofía descentralizada de OpenPGP.

2) Hagrid: El proyecto Sequoia-PGP es una reimplementación de diversos aspectos de OpenPGP. Uno de los módulos que lo constituyen es el servidor de llaves con validación *Hagrid* [2]. Este servidor de llaves no es, por diseño, interoperable con la red de servidores preexistente, y puesto que descarta todas las firmas de las llaves que entrega, hace imposible la ocurrencia de los ataques antes mencionados: permite a los usuarios buscar llaves específicas, pero elimina la información necesaria para establecer una WoT.

Com lo menciona el sitio Web del proyecto, Hagrid no publica información relativa a la identidad sin el consentimiento del usuario en cuestión, y permite la eliminación de información personal. Al día de hoy no es *federable* (carece de un componente de sincronización entre servidores), aunque esta característica forma parte de su mapa de desarrollo futuro.

C. Dejar a OpenPGP

Otros autores caracterizan a OpenPGP como pertenecientes a una era distinta, reflejando un distinto nivel de confianza que se tenía en Internet hace 30 años, y proponen nuevos protocolos para reemplazarlo, junto con diferentes suposiciones iniciales y modelos de amenazas propios.

1) Off The Record (OTR): OTR busca facilitar las comunicaciones de baja latencia, orientadas a mensajes cortos, como las que se producen en la mensajería instantánea [1]. OTR implementa *secreto perfecto hacia adelante*, llaves de sesión efímeras, y repudiabilidad explícita. El modelo de confianza es TOFU (descrito en la Subsección V-A3), ofreciendo también la posibilidad de autenticar la llave de un contacto fuera de banda.

VI. Conclusion

El presente trabajo muestra una modificación menor al flujo de validación de identidades empleando la WoT de OpenPGP, requiriendo que el dueño de una llave atestigue (dé fé de haber aceptado) toda modificación hecha a su lleve pública. Esta modificación logra que un ataque devastador como el *envenenamiento de certificados* se vuelva imposible, y puede llevar a facilitar que el modelo distribuido transitivo de confianza basado en mallas de confianza siga teniendo viabilidad.

A. Trabajo futuro

El presente trabajo describe un *trabajo en proceso*. Si bien este artículo delínea la idea principal, es necesario aún sustentar con pruebas y con una implementación los dichos que presenta. Para tal fin, trabajaremos para adaptar programas servidor existentes para la implementación de servidores de llave pública, para que requieran la modificación al protocolo que introduce nuestra propuesta.

Además del problema de los certificados envenenados, es muy importante atacar el problema de la remoción de información de los servidores públicos de llaves, para poder cumplir con la GDPR (o solicitudes de redes comparables en distintas jurisdicciones); si bien abordar ese problema excede el ámbito del trabajo presente, la propuesta de una nueva infraestructura de servidores de llaves constituiría una gran oportunidad para trabajarlo.

Referencias

[1] N. Borisov, I. Goldberg, and E. Brewer, “Off-the-record communication, or, why not to use pgp,” in *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, Association for Computing Machinery, 2004, pp. 77–84.

[2] V. Breitmoser, J. Winter, K. Michaelis, and N. Widdecke, *Hagrid: A verifying keyserver*, 2022. (visited on 03/28/2022).

[3] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, “Openpgp message format,” *Internet Engineering Task Force (IETF)*, no. 4880, 2007.

[4] K. Fiskerstrand, “An update on the sks-keyservers.net services,” in *OpenPGP.conf*, 2016.

[5] Fortinet, *Pgp encryption*, 2022. (visited on 03/27/2022).

[6] D. Hamilton, *Pgp for absolute beginners*, 1998. (visited on 03/27/2022).
[7] M. Horowitz, "Pgp public key server," M.S. thesis, MIT, 1997.
[8] D. Kahn Gillmor, *Openpgp certificate flooding*, 2019.
[9] W. Koch, "OpenPGP Web Key Directory," Internet Engineering Task Force, Internet-Draft draft-koch-openpgp-webkey-service-12, May 2021, Work in Progress, 18 pp. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-koch-openpgp-webkey-service-12.
[10] M. F. Lee, *Trolling the web of trust*, 2014.
[11] M. F. Lee, *The death of sks pgp keyservers, and how first look media is handling it*, 2019.
[12] Y. Minsky and A. Trachtenberg, "Practical set reconciliation," in *40th Annual Allerton Conference on Communication, Control and Computing*, vol. 248, 2002.
[13] J. Petters, *What is pgp encryption and how does it work?* 2020. (visited on 03/27/2022).
[14] P. Pramberger, *Keyserver.pramberger.at terminating*, 2010.
[15] D. Shaw, "The openpgp http keyserver protocol (hkp)," Internet Engineering Task Force, Internet-Draft draft-shaw-openpgp-hkp-00, Mar. 2003, Work in Progress, 8 pp.
[16] N. H. Walfield and W. Koch, "Tofu for openpgp," in *EuroSec'16: Proceedings of the 9th European Workshop on System Security*, 2016, pp. 1–6.
[17] P. Wouters, "Dns-based authentication of named entities (dane) bindings for openpgp," *Internet Engineering Task Force (IETF)*, no. 7929, 2016.
[18] K. Yakamo, *Are pgp key-servers breaking the law under the gdpr?* 2018.
[19] K. Yakamo, *Are sks keyservers safe? do we need them?* 2018.
[20] K. Yakamo, *Using pgp keyservers for decentralised file storage*, 2019.
[21] A. Yakubov, W. Shbair, N. Khan, R. State, C. Medinger, and J. Hilger, "Blockpgp: A blockchain-based framework for pgp key servers," *International Journal of Networking and Computing*, vol. 10, no. 1, pp. 1–24, 2020.