Recepción: 21 de octubre de 2017**Aceptación:** 30 de noviembre de 2017**Publicación:** 14 de diciembre de 2017

PROPUESTA DE PROCEDIMIENTO DE SELLADO DE TIEMPO INDEPENDIENTE DE UNA AUTORIDAD CERTIFICADORA

PROPOSAL FOR A TIMESTAMPING METHOD THAT DOES NOT DEPEND ON A CERTIFICATION AUTHORITY

Romel Vera Cadena¹Rayner Durango Espinoza²

1. Consultor de Seguridad Informática. Maestría en Auditoría de Tecnologías de la Información. Universidad de Especialidades Espíritu Santo (Ecuador). E-mail: romel.vera.cadena@gmail.com
2. Magíster en Sistemas de Información Gerencial (Ecuador). E-mail: rdurango@uees.edu.ec

Citación sugerida:

Vera Cadena, R. y Durango Espinoza, R. (2017). Propuesta de procedimiento de sellado de tiempo independiente de una autoridad certificadora. *3C Tecnología: glosas de innovación aplicadas a la pyme*, 6(4), 24-45. DOI: <http://dx.doi.org/10.17993/3ctecno.2017.v6n4e24.24-45>.

RESUMEN

El sellado de tiempo confiable se emplea para certificar que un documento informático se encuentre íntegro desde la fecha en que se registró, sin embargo, en el modelo actual de sellado de tiempo en su procedimiento éstos dependen de una autoridad certificadora las cuales están sujetas a la confianza. Actualmente, existen casos en los que la confianza ha sido vulnerada por diversos motivos que van desde un mal manejo de certificados digitales hasta descuidos por parte de las autoridades certificadoras. Por lo tanto, existe exposición a suplantación de identidad, fraude en documentos, interceptación de comunicaciones, difusión de malware, y sabotaje. Por este motivo, se propone un procedimiento de sellado de tiempo que no dependa de una autoridad certificadora.

ABSTRACT

Trusted timestamping is the process used to certify that a computer document is not altered from the date it was registered; however, the current timestamping procedure depends on a certifying authority, which is subject to trust. In addition, there are currently cases where the trust has been lost for various reasons ranging from mismanagement of digital certificates to oversight by certification authorities. Therefore, there is exposure to identity theft, document fraud, interception of communications, malware spreading, and sabotage. For this reason, a new timestamping procedure is proposed that does not depend on a certification authority.

PALABRAS CLAVE

Sellado de Tiempo, Seguridad, Confianza, Certificado Digital, Autoridad Certificadora.

KEY WORDS

Timestamping, Security, Trust, Digital Certificates, Certificate Authority.

1. INTRODUCCIÓN

El procesamiento de datos digitales ha traído muchos beneficios a la forma en que se crea y se organiza la información, en forma de imágenes, documentos, archivos de audio o video. La conversión de dicha información digitalizada proporciona capacidades como el almacenamiento y recuperación, procesos de búsqueda eficientes y transmisión de datos (Cassany, 2000).

Sin embargo, los datos digitales también traen consigo un problema importante como es la facilidad con la que la información se puede copiar y manipular, sin dejar evidencia alguna. Y ha surgido la necesidad de contar con tecnologías capaces de salvaguardar la integridad y la autenticidad de los registros digitales, en particular si tales registros están sujetos a pruebas jurídicas y/o éticas. La preservación de la integridad de los registros digitales es particularmente importante cuando están sujetos a un posible ataque criminal (Voutssas, 2010).

En consecuencia, en las organizaciones es necesario contar con la veracidad sobre la integridad de documentos informáticos, es decir, que deben tener la capacidad de determinar si un documento fue modificado sin autorización previa con la finalidad de evitar fraudes y por efecto perjuicios en la información y evitar pérdidas económicas (Trejo, Domenech, & Ortíz, 2016).

En el caso de los documentos gubernamentales que se manejan con herramientas de editor de textos u hojas de cálculo, hay aplicaciones que ayudan a cifrarlos, pero esto puede complicar el manejo de los archivos, cuando solamente se desea que no sean adulterados (Stone, 2016).

2. SITUACIÓN ACTUAL EN LA INTEGRIDAD DE DOCUMENTOS Y SU PROBLEMÁTICA

Actualmente existe un modelo estándar de sellado de tiempo confiable para la verificación de integridad en los documentos informáticos, sin embargo, estos requieren del uso de certificados digitales (Une, 2001).

El sellado de tiempo confiable hace uso de certificados digitales sin embargo los certificados digitales están sujetos a la confianza, y la confianza se basa en dos principios:

1. La Autoridad Certificadora (CA) no debería emitir certificados a usuarios inapropiados.
2. Los usuarios no deben agregar ningún CA inapropiado a su lista de CAs de confianza.

Este concepto de confianza se encuentra en debate debido a que existen grandes organizaciones e instituciones que no necesariamente son de confianza (Eckersley, 2015) (Helme, 2017).

2.1. LOS CERTIFICADOS EMITIDOS Y LOS PROBLEMAS EN ORGANIZACIONES

Es posible conseguir certificados digitales usando cualquier nombre y en casos hasta el mismo dominio. Por ejemplo: LetsEncrypt emitió 15,270 certificados que contienen la palabra "Paypal" (Brenner, 2017).

También, existen casos que empresas conocidas a nivel mundial han tenido este tipo de problema:

Tabla 1. Lista parcial de empresas que han tenido problemas de confianza.

Empresa	Por medio de	Caso
Comodo	Diginotar	Paypal
Google	China Internet Network Information Center (CNNIC)	Google - CNNIC
Symantec	Varios CAs de Symantec	Google Chrome EV & Symantec
Kaspersky Lab	Foxconn	Stuxnet
Wosign, StartCom	Gobierno de China	Google vs China
Microsoft	Gobierno de Tailandia	Microsoft Trusted vs Privacy

Fuente: elaboración propia, 2017.

2.2. OBJETIVO GENERAL

El objetivo general del presente artículo es crear un nuevo procedimiento de verificación de integridad de documentos informáticos que no dependa de una autoridad certificadora.

3. MARCO TEÓRICO

3.1. FUNCIÓN HASH

La función hash conocida también como funciones de resumen o message digest en inglés, son algoritmos que a partir de una entrada de texto, contraseña o archivos crean una salida alfanumérica que representa un resumen de toda la información que se ha dado, es decir que solo puede volverse a crear con esos mismos datos y esta función tiene como objetivo principal asegurar que un determinado texto o archivo no se encuentre manipulado (NIST, 2017) (Johnson, Badger, Waltermire, Snyder, & Skorupka, 2016).

3.2. FIRMA DIGITAL / FIRMA ELECTRÓNICA

Las firmas digitales o electrónicas conocidas en inglés como Digital Signatures sirven para asegurar el origen del documento y la integridad de los datos del documento. La firma digital asegura que la persona que firma no pueda rechazar su propio documento a esto se le conoce como “No Repudio” (Barker & Roginsky, 2015).

Para garantizar la seguridad de las firmas digitales es necesario que éstas sean únicas, infalsificables, verificables, innegables, viables (Shirey, 2007).

- Únicas: Las firmas deben poder ser generadas solamente por el firmante y por lo tanto infalsificable.
- Infalsificables: Las firmas deben ser computacionalmente seguras.
- Verificables: Las firmas deben ser fácilmente verificables por los receptores de las mismas.
- Innegables, No repudiable: El firmante no debe ser capaz de negar su propia firma.
- Viables: Las firmas deben ser fáciles de generar por parte del firmante.

En el marco legislativo ecuatoriano la firma digital se define como el conjunto de datos en forma electrónica, consignados junto a otros o asociados con ellos, que pueden ser utilizados como medio de identificación del firmante (Banco Central del Ecuador, 2011).

3.3. CERTIFICADO DIGITAL

Un certificado digital es un archivo informático generado por una entidad de servicios de certificación que asocia unos datos de identidad a una persona física, organismo o empresa confirmando de esta manera su identidad digital en Internet. El certificado digital es válido principalmente para autenticar a un usuario o sitio web en Internet, por lo que es necesario, la colaboración de un tercero que sea de confianza para cualquiera de las partes que participe en la comunicación. El nombre asociado a esta entidad de confianza es Autoridad Certificadora pudiendo ser un organismo público o empresa reconocida en Internet (Kuhn, Hu, Polk, & Chang, 2001).

3.4. SELLADO DE TIEMPO CONFIABLE

El sellado de tiempo confiable o “trusted timestamping” en inglés, es un mecanismo en línea que permite demostrar que una serie de datos han existido y no han sido manipulados desde un instante específico en el tiempo. Este protocolo se describe en el RFC 3161 y está en el registro de estándares de Internet (Zuccherato, Cain, Adams, & Pinkas, 2001) (ANSI, 2016).

3.5. AUTORIDAD DE SELLADO DE TIEMPO

Una autoridad de sellado de tiempo (TSA) actúa como tercera parte de confianza testificando la existencia de una serie de datos en una fecha y hora concretos (Une, 2001).

3.6. SOFTWARE COMO UN SERVICIO “SCUS / SAAS”

Software como un Servicio, abreviadamente ScuS (del inglés: Software as a Service, SaaS), es un modelo de distribución de software donde el soporte lógico y los datos que maneja se alojan en servidores centralizados en una compañía (Turner, Budgen, & Brereton).

4. METODOLOGÍA

Se procederá a describir el proceso de sellado de tiempo confiable para posteriormente revisar los tres casos más populares de problemas de confianza en los certificados digitales. La selección de los casos se determinó por el índice de popularidad de los resultados de Google del día martes 28 de marzo del 2017. Seguidamente se graficará en qué parte del proceso del modelo de timestamping afecta el problema de la confianza, con el objetivo de reestructurar el procedimiento y poder quitar la dependencia externa de la Autoridad Certificadora.

Posteriormente, se desarrollará una aplicación SaaS/ScuS demo en la que se hace uso del nuevo procedimiento de timestamping.

Finalmente, se creará un cuadro comparativo de soluciones similares tomando en cuenta la verificación de la integridad, la verificación de la fecha, la dependencia de los CA, si es un servicio o no, y si es comercial o no.

Para la elaboración de la aplicación SaaS/ScuS demo se usará el sistema operativo Ubuntu 16.04 LTS, para el frontend se usará HTML5, CSS3, Javascript y los componentes jQuery, Datatables, Bootstrap, Responsive, y Select. Para el backend se usará Apache2, PHP7 y MySQL 5.

Se han escogido éstas plataformas de desarrollo porque sus licencias permisivas consienten desarrollar software de código abierto por lo tanto no se está sujeto a ninguna obligación con un tercero.

5. SELLADO DE TIEMPO CONFIABLE / TRUSTED TIMESTAMPING

Según el estándar RFC 3161 se puede obtener un sello de tiempo por medio de un tercero de confianza con el fin de probar la veracidad de datos tales como: contratos, investigaciones, bitácoras médicas, documentos financieros, etc.

5.1. PROCESO DE SELLADO DE TIEMPO CONFIABLE

En el proceso de sellado de tiempo confiable se describen los siguientes pasos:

1. Un usuario quiere obtener un sello de tiempo para un documento digital que él posee.
2. Un resumen digital “técnicamente un hash” se genera para el documento en el ordenador del usuario.
3. Este resumen forma la solicitud que se envía a la autoridad de sellado de tiempo (TSA).
4. La TSA genera un sello de tiempo con esta huella, la fecha y hora obtenida de una fuente fiable y la firma digital de la TSA.
5. El sello de tiempo se envía de vuelta al usuario.
6. La TSA mantiene un registro de los sellos emitidos para su futura verificación.

Figura 1. Setting a timestamp via a trusted third party.

Fuente: Bart Van den Bosch, 2004.

Y para la verificación:

1. Se calcula el hash de la información original.
2. Se adjunta el sellado de tiempo otorgado por la TSA.

3. Se calcula nuevamente un hash de la información anterior. Llamemos a este último hash A.
4. El hash A es comparado con el hash B dentro del mensaje firmado por la TSA para poder comprobar que son idénticos, evidenciando que el sellado de tiempo y el mensaje no fueron alterados y que éste fue realmente publicado por la TSA. Si esto no ocurre, entonces el sellado de tiempo fue alterado o bien el sellado de tiempo no fue emitido por la TSA.

Figura 2. Checking a timestamp.

Fuente: Tsuruya, 2012.

6. PROBLEMAS DE CONFIANZA - CASOS

Los certificados digitales son la base fundamental de la confianza en línea. Estos certificados digitales se comparan con las firmas, y normalmente se confía en un documento porque tienen una firma o lo respalda una autoridad de certificación en la que se confía. Es decir que los certificados digitales son una reproducción de un modelo simple que ocurre en la vida real (Lax, Buccafurri, & Caminiti, 2015).

Los incidentes que involucran certificados digitales y a las autoridades certificadoras (CA) no siempre son claros o notables para los usuarios finales. Sin embargo, los administradores de TI, los desarrolladores de software y otros profesionales de la seguridad de la información necesitan comprender estos problemas para que los riesgos puedan gestionarse adecuadamente (Laurie, 2014). Sin embargo, existen muchos casos en la que se pierde confianza en las Autoridades Certificadoras (CAs).

6.1. GOOGLE BLOQUEA CERTIFICADOS DE CAS DE CHINA WOSIGN Y STARTCOM

En agosto 17 del 2016 la compañía GitHub había notificado a Google que la autoridad certificadora WoSign había emitido un certificado base a nombre de GitHub sin previa autorización. Después de este reporte Google con la colaboración de Mozilla descubrieron algunos casos de WoSign en las que emitía certificados no autorizados.

Como resultado Google procedió a eliminar progresivamente de la lista de confianza del navegador Google Chrome a todos los elementos pertenecientes a WoSign y StartCom. Y según el calendario de actividades de Google los cambios finalizarán en septiembre del 2017.

En el año 2016 Apple y Mozilla ya habían dejado de confiar en WoSign y StartCom debido a dificultades técnicas que tenían con éstas compañías. Adicionalmente Mozilla había encontrado que WoSign había adquirido el CA StartCom y esto nunca fue notificado a ninguna compañía pese a las políticas establecidas entre ellas (Corser, 2015).

6.2. SYMANTEC PIERDE CONFIANZA

Desde octubre del año 2015 la empresa Symantec ha estado perdiendo gradualmente confianza, debido a problemas de seguridad por la forma en que emite sus certificados digitales y anteriormente Symantec había prometido solucionar el problema, sin embargo, Google reporta que no ha sido el caso y que va a dejar de confiar en los certificados del tipo EV “de alta seguridad” que genera Symantec. Esta medida afecta a otras marcas que contrala Symantec como son Geotrust, VeriSign, y Thawte.

Como contramedida frente a estos problemas Google va a asignar fechas de expiración temprana a los certificados de Symantec. En el navegador Google Chrome para su versión 62 que saldrá en octubre del 2017 va a fijar en los certificados digitales de Symantec una expiración de 15 meses a partir de su fecha de vigencia (Goodin, 2017).

6.3. VARIANTE DE STUXNET INFECTA KASPERSKY POR MEDIO DE CERTIFICADOS DIGITALES FOXCONN

En el año 2015 el malware Duqu 2.0 infectó la red corporativa de la empresa Kaspersky Lab creadora del anti-virus Kaspersky. El malware había pasado inadvertido debido a que estaba firmado con un certificado digital perteneciente a la compañía Foxconn que se encarga de manufacturar componentes electrónicos para iPhone, Xbox, Laptops Acer, Laptops Dell, entre otros productos conocidos (Goodin, 2015).

En el sistema operativo Windows para poder instalar controladores o drivers en inglés, es necesario que los drivers se encuentren firmados con un certificado digital. Foxconn usa drivers firmados para ciertos componentes que fabrica y de alguna forma los atacantes se adueñaron de los certificados digitales para luego firmar el malware como un driver legítimo de Foxconn. Duqu 2.0 se había instalado en firewalls, puertas de enlace y otros servidores que tenían enlace directo a Internet, nunca se llegó a conocer el verdadero propósito de los creadores del Duqu 2.0 aunque en los análisis de Kaspersky se refleja que los creadores tenían como objetivo principal las empresas que fabrican hardware (Kaspersky, 2015).

7. ANÁLISIS DEL PROBLEMA DE LOS CASOS

Los certificados digitales dependen mucho de la seguridad de la organización que los maneja y en el caso de los equipos de computación, laptops, PCs, servidores, celulares etc. En su operación están sujetos a cadenas de confianza, cada aplicación, servicio o librería confía en uno o más CAs, por ejemplo: En un celular Samsung Galaxy Note es posible encontrar aproximadamente 300+ aplicaciones y servicios que dependen de certificados digitales para diversas funciones que van desde una conexión segura hasta la comprobación de integridad de aplicaciones al revisar una firma digital (Blaich, 2015).

Figura 3. Círculo de confianza.

Fuente: Andrew Blaich, 2015.

Un mal manejo de un certificado digital puede tener varias consecuencias tales como la suplantación de identidad, fraude en documentos, interceptación de comunicaciones, difusión de malware, y sabotaje (Paganini, 2014).

8. GRÁFICO DEL PROBLEMA DE CONFIANZA EN EL PROCESO DE SELLADO DE TIEMPO CONFIABLE

En el proceso de sellado de tiempo confiable el problema de la confianza afectaría al paso número 4:

4. La TSA genera un sello de tiempo con esta huella, la fecha y hora obtenida de una fuente fiable y la firma digital de la TSA.

Figura 4. Gráfico del problema de confianza en el proceso de sellado de tiempo confiable.

Fuente: Bart Van den Bosch, 2004.

9. PROPUESTA DE NUEVO PROCEDIMIENTO DE SELLADO DE TIEMPO

El nuevo procedimiento que propongo se llama “Simple Hashing” que es una simplificación de la estructura del procedimiento actual de sellado de tiempo confiable que no requiere de un certificado digital ni genera un archivo de comprobación puesto que la comprobación se hace directamente desde el servidor interno de la organización.

9.1. BENEFICIOS DEL NUEVO PROCEDIMIENTO

- No requiere un certificado de una Autoridad Certificadora (no dependería de la confianza de una autoridad externa).
- El usuario no almacena o lleva archivos adicionales.
- Resultado inmediato en pantalla.
- Menos complejidad de uso al evitar el manejo de certificados digitales.
- Menos complejidad en la implementación debido a la simplificación del proceso.

Figura 5. Propuesta de nuevo procedimiento de sellado de tiempo confiable – Simple Hashing.

Fuente: elaboración propia, 2017.

En el proceso propuesto se describen los siguientes pasos:

1. Un usuario quiere obtener un sello de tiempo para un documento digital que él posee.
2. Un resumen digital “técnicamente un hash” se genera para el documento en el ordenador del usuario.
3. Este resumen forma la solicitud que se envía al servidor TSA.
4. La TSA genera un sello de tiempo con los datos del usuario, hash, la fecha y hora obtenida del servidor.
5. El sello de tiempo se guarda en la base de datos del servidor.
6. La TSA mantiene un registro de los sellos registrados para su futura verificación.

Y para la verificación:

1. Se calcula el hash del documento del usuario.

El hash del documento del usuario es comparado con el hash registrado en la base de datos del TSA para comprobar que son idénticos, evidenciando que el sellado de tiempo y el mensaje no fueron manipulados y que éste fue realmente registrado por la TSA. Si esto no ocurre, entonces el sellado de tiempo fue manipulado o bien el sellado de tiempo no fue emitido por la TSA.

10. IMPLEMENTACIÓN DE LA PROPUESTA SIMPLE HASHING

La implementación de la propuesta se desarrolla usando los siguientes elementos:

Tabla 2. Lista de elementos a usar para la implementación de la propuesta.

Elemento	Categoría	Función
Ubuntu 16.04 LTS	Sistema Operativo	Plataforma base
MySQL 5	Backend	Base de datos
PHP7	Backend	Motor dinámico web PHP
Apache2	Backend	Servidor HTTP
HTML5	Frontend	Lenguaje básico web
CSS3	Frontend	Lenguaje de diseño web
Javascript	Frontend	Lenguaje de contenido dinámico
jQuery	Frontend	Librería Javascript multiplataforma
Bootstrap	Frontend	Framework Javascript de interfaces
Datatables	Frontend	Librería Javascript para datos
Responsive	Frontend	Librería Javascript para contenidos
Select	Frontend	Librería Javascript de estilos de selección de datos

Fuente: elaboración propia, 2017.

El esquema de trabajo de los elementos es el siguiente:

Figura 6. Esquema de trabajo de los elementos de la propuesta - Simple Hashing

Fuente: Elaboración propia, 2017

10.1. DISEÑO DE LA BASE DE DATOS

Para el diseño de la base de datos se han asignado campos que permitan realizar una auditoría, es decir, para que sea posible cruzar información por email, fecha de creación, fecha de último ingreso, registros generados, registros borrados, registros inválidos, contraseñas duplicadas o débiles, etc.

En el campo de los hashes es necesario elegir un algoritmo de resumen de mensajes vigente, en este caso se hace uso de SHA256.

El diseño de la base de datos es el siguiente:

Figura 7. Diseño de la base de datos - Simple Hashing.

Fuente: elaboración propia, 2017.

Tabla 3. Descripción de los campos de las tablas de la base de datos – Simple Hashing.

Tabla user		
Campo	Descripción	Campo Auditable
iduser	Llave primaria	No
firstname	Nombre	No
lastname	Apellido	No
social_id	Cédula	Sí
email	E-mail	Sí
password	Contraseña en SHA256	Sí
creation_date	Fecha de creación	Sí
last_login_date	Fecha de último ingreso	Sí
is_active	El usuario está habilitado	Sí
role	Rol del usuario	Sí
admin_max_users	Límite de usuarios que puede crear un administrador	Sí
Tabla filename		
idfilename	Llave primaria	No
filename	Nombre del archivo	No
file_extension	Extensión del archivo	No
hash_primary	Hash primario (SHA256)	Sí
hash_secondary	Hash secundario (CRC32)	Sí
date_uploaded	Fecha de registro	Sí
user_email	Email del usuario que registró el archivo	Sí
file_status	Estado del documento, (Válido, Borrado, Bloqueado)	Sí

Fuente: elaboración propia, 2017.

10.2. ALGORITMO – INICIO DE REGISTRO DE SELLADO DE TIEMPO

Figura 8. Inicio de registro de sellado de tiempo - Simple Hashing.

Fuente: Elaboración propia, 2017.

10.3. ALGORITMO – VERIFICACIÓN DE REGISTRO DE SELLADO DE TIEMPO

Figura 9. Verificación de registro de sellado de tiempo - Simple Hashing.

Fuente: Elaboración propia, 2017.

10.4. ALGORITMO – CAMBIO DE ESTADO DEL REGISTRO DE SELLADO DE TIEMPO

Figura 10. Cambio de estado de registro de sellado de tiempo - Simple Hashing.

Fuente: elaboración propia, 2017.

10.5. ALGORITMO LÓGICA

α = Hash del documento

β = Existe en Base de datos

ζ = Graba Registro

δ = Muestra Mensaje

λ = Concatena datos

ϵ = Procesa estado de datos

str1 = Registro exitoso

str2 = El registro ya existe

str3 = El documento no es válido o no existe

Algoritmo de Registro

$$\alpha(x) \wedge \neg\beta(x) \rightarrow \lambda(x) \wedge \zeta(x) \wedge \delta(\text{str1}) \therefore 1$$

$$\alpha(x) \wedge \beta(x) \rightarrow \delta(\text{str2}) \therefore 0$$

Algoritmo de Verificación

$$\alpha(x) \wedge \neg\beta(x) \rightarrow \delta(\text{str3}) \therefore 0$$

$$\alpha(x) \wedge \beta(x) \rightarrow \lambda(x) \wedge \varepsilon(x) \wedge \delta(str[\varepsilon(x)]) \therefore 1$$

Algoritmo de Cambio de Estado

$$\alpha(x) \wedge \beta(x) \rightarrow \varepsilon(x) \wedge \zeta(x) \wedge \delta(str1) \therefore 1$$

$$\alpha(x) \wedge \neg\beta(x) \rightarrow \delta(str3) \therefore 0$$

Es posible obtener los códigos fuentes en Github: <https://github.com/RomelSan> o pedirlos directamente al email: romel.vera.cadena@gmail.com

11. TABLA COMPARATIVA DE SOLUCIONES SIMILARES

La información obtenida parte de los mismos datos que se encuentran en los sitios oficiales de los programas.

Tabla 4. Tabla comparativa de soluciones similares.

Software	Chequea Integridad	Verifica Fecha	Depende de CA	SaaS	Comercial	Link
Digistamp	Sí	Sí	Sí	Sí	Sí	https://www.digistamp.com
SignFiles	Sí	Sí	Sí	Sí	Sí	https://www.signfiles.com
GlobalSign	Sí	Sí	Sí	Sí	Sí	https://www.globalsign.com
SignServer	Sí	Sí	Sí	Sí	Sí*	https://www.signserver.org
FreeTSA	Sí	Sí	Sí	Sí	Gratis	https://www.freetsa.org
QuickSFV	Sí	No	No	No	Gratis	http://www.quicksfv.org
Hashtab	Sí	No	No	No	Sí*	http://implbits.com
MultiHasher	Sí	No	No	No	Gratis	http://www.abelhadigital.com
Hashtools	Sí	No	No	No	Gratis	https://www.binaryfortress.com
Propuesta: Simple Hash	Sí	Sí	No	Sí	Gratis	N/A

Nota: Sí* = ofrece una versión de costo de uso empresarial y una versión gratuita sin soporte y sin algunos componentes que se encuentran en la versión pagada.

Fuente: elaboración propia (mayo 2017).

Todos los softwares en la tabla 4 tienen la capacidad de verificar la integridad de documentos informáticos, sin embargo, no todos ellos dependen de CA o verifican fechas y esto se debe porque cada software está diseñado para resolver requerimientos diferentes.

12. CONCLUSIONES

Las personas y empresas deben velar por la integridad de sus documentos y no necesariamente para lograrlo deben estar sujetas a un CA, especialmente cuando existen problemas de confianza. Es por este motivo que existen varias soluciones que van desde un simple chequeo de integridad hasta operaciones más complejas en las que además de la integridad es posible también verificar fechas, firmas, origen de datos, autenticidad, etc.

El mal manejo de certificados digitales causa desconfianza y genera problemas de seguridad y es necesario que las personas y empresas estén conscientes de las entidades en las que confían, y adicionalmente deberán estar pendientes de las noticias de seguridad de la información con el fin de mitigar daños en el menor tiempo posible.

13. RECOMENDACIONES PARA FUTUROS TRABAJOS

Es posible adaptar este mismo procedimiento propuesto del tipo SaaS a una red descentralizada privada punto a punto como sería el caso de la cadena de bloques en la que también se hace uso del sellado de tiempo confiable. También es posible crear complementos o extensiones para soluciones existentes de manejo de contenidos, incluso se puede diseñar y analizar métricas de desempeño.

14. RECONOCIMIENTOS

La realización del presente artículo fue posible gracias a la guía del Profesor Rayner Stalyn Durango Espinoza, tutor de tesis de la Universidad de Especialidades Espíritu Santo, quien en su desempeño como tutor procuró por la claridad y simplicidad del presente artículo.

15. REFERENCIAS BIBLIOGRÁFICAS

- ANSI. (2016). Trusted Time Stamp Management and Security.
- Apache. (2016). *HTTP Server Project*. Obtenido de <http://httpd.apache.org/>
- Banco Central del Ecuador. (2011). Firma Electrónica. Obtenido de <https://www.eci.bce.ec/marco-normativo>
- Barker, E. B., & Roginsky, A. L. (nov de 2015). *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*. National Institute of Standards and Technology (NIST). doi:10.6028/nist.sp.800-131ar1
- Blaich, A. (2015). Decrease Your Circle of Trust: An Investigation of PKI CAs on Mobile Devices. *RSA Conference 2015*, (págs. 1-68). San Francisco.
- Brenner, B. (30 de March de 2017). *Let's Encrypt issues certs to 'PayPal' phishing sites: how to protect yourself*. Obtenido de Naked Security by Sophos: <https://nakedsecurity.sophos.com/2017/03/30/lets-encrypt-issues-certs-to-paypal-phishing-sites-how-to-protect-yourself/>
- Cassany, D. (2000). De lo analógico a lo digital. El futuro de la enseñanza de la composición. *Lectura y vida*, 6-15.
- Corser, C. (2015). Google's Distrust of Chinese Digital Certificates. Obtenido de <http://ncjolt.org/>
- Eckersley, P. (2015). *The Huge Web Security Loophole That Most People Don't Know About, And How It's Being Fixed*. Obtenido de Electronic Frontier Foundation: <https://www.eff.org/about/staff/peter-eckersley>
- Flanagan, D. (2006). *JavaScript: The Definitive Guide*. O'Reilly Media. Obtenido de <https://books.google.com.ec/books?id=2weL0iAfrEMC>
- Goodin, D. (2015). Stuxnet spawn infected Kaspersky using stolen Foxconn digital certificates.
- Goodin, D. (2017). Google takes Symantec to the woodshed for mis-issuing 30,000 HTTPS certs. Obtenido de <https://arstechnica.com/security/2017/03/google-takes-symantec-to-the-woodshed-for-mis-issuing-30000-https-certs/>
- Hars, A. (2002). Working for Free? Motivations for Participating in Open-Source Projects. *International Journal of Electronic Commerce*, 6, 25-39.
- Helme, S. (2017). HTTPS Certificate Revocation is broken, and it's time for some new tools. *Ars Technica*. Obtenido de <https://arstechnica.com/security/2017/07/https-certificate-revocation-is-broken-and-its-time-for-some-new-tools/>
- Johnson, C. S., Badger, M. L., Waltermire, D. A., Snyder, J., & Skorupka, C. (oct de 2016). *Guide to Cyber Threat Information Sharing*. National Institute of Standards and Technology (NIST). doi:10.6028/nist.sp.800-150
- jQuery Foundation. (2016). *What is jQuery?* Obtenido de <https://jquery.com/>
- Kaspersky, S. (2015). The Duqu 2.0 persistence module. Obtenido de <https://securelist.com/the-duqu-2-0-persistence-module/70641/>
- Kuhn, R., Hu, V., Polk, T., & Chang, S.-J. (2001). *Introduction to Public Key Technology and the Federal PKI Infrastructure*. NIST.

- Lakhani, K. R. (2003). How open source software works: “free” user-to-user assistance. *Research Policy*, 32, 923-943.
- Laurie, B. (2014). Certificate transparency. *Queue*, 10.
- Lax, G., Buccafurri, F., & Caminiti, G. (2015). Digital document signing: Vulnerabilities and solutions. *Information Security Journal: A Global Perspective*.
- MDN. (2016). *HTML developer guide*. Obtenido de <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML>
- NIST. (2017). Cryptographic Hash.
- OpenSource. (s.f.). *What is Linux?* Obtenido de <https://opensource.com/resources/what-is-linux>
- Paganini, P. (2014). *How Cybercrime Exploits Digital Certificates*. Obtenido de Infosec Institute: <http://resources.infosecinstitute.com/cybercrime-exploits-digital-certificates/>
- Shirey, R. (2007). Internet Security Glossary, Version 2. Obtenido de <https://tools.ietf.org/html/rfc4949>
- Sklar, D. (2016). *Learning PHP: A Gentle Introduction to the Web's Most Popular Language*. O'Reilly Media. Obtenido de <https://books.google.com.ec/books?id=-ZT4CwAAQBAJ>
- Spurlock, J. (2013). *Bootstrap*. O'Reilly Media. Obtenido de <https://books.google.com.ec/books?id=LZm7Cxgi3aQC>
- Stone, A. (2016). Protecting Data Means Balancing Security vs. Convenience. *GovTech Works*.
- Tahaghoghi, S. M., & Williams, H. E. (2006). *Learning MySQL*. O'Reilly Media. Obtenido de <https://books.google.com.ec/books?id=KQONxdHXS-wC>
- Trejo, C., Domenech, G., & Ortíz, K. (2016). LA SEGURIDAD JURÍDICA FRENTE A LOS DELITOS INFORMÁTICOS. *AVANCES*.
- Turner, M., Budgen, D., & Brereton, P. (s.f.). Turning Software into a Service.
- Une, M. (2001). The Security Evaluation of Time Stamping Schemes: The Present Situation and Studies. *IMES Discussion Papers Series 2001-E-18*, (págs. 100-8630).
- Voutssas, M. (2010). Preservación documental digital y seguridad informática. *Investigación bibliotecológica*, 127-155.
- W3C. (2015). *What is CSS?* Obtenido de <https://www.w3.org/Style/CSS/>
- Zuccherato, R., Cain, P., Adams, D. C., & Pinkas, D. (2001). Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC Editor. doi:10.17487/RFC3161

16. GLOSARIO

Código Abierto “Open Source”

El código abierto es el software desarrollado y distribuido libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre. Para muchos el término «libre» hace referencia al hecho de adquirir un software de manera gratuita, pero más que eso, la libertad se refiere al poder modificar la fuente del programa sin restricciones de licencia, ya que muchas empresas de software encierran su código, ocultándolo, y restringiéndose los derechos a sí misma. (Hars, 2002) (Lakhani, 2003)

Linux

Linux es un sistema operativo que sirve para que la computadora permita al usuario utilizar programas tales como: navegadores de internet, editores de texto, etc. Linux se origina a partir de Unix el cual apareció en los años sesenta y fue desarrollado por Dennis Ritchie y Ken Thompson del laboratorio telefónico de la empresa Bell. (OpenSource, n.d.)

HTML5

Es el lenguaje básico para la elaboración de páginas de internet, sus siglas HTML significan “HyperText Markup Language” en español “lenguaje de marcas de hipertexto” y el número cinco nos indica su revisión actual. (MDN, 2016)

CSS3

Es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento HTML, es muy usado para establecer el diseño visual de las páginas web o de cualquier otro lenguaje de marcado. (W3C, 2015)

Javascript

Javascript es un lenguaje de programación interpretado orientado a objetos. Se usa para mejorar la interfaz de usuario y para crear páginas web dinámicas del lado del cliente. Usa el dialecto ECMAScript estandarizado. (Flanagan, 2006)

Bootstrap

Bootstrap es un complemento de código abierto desarrollado por Twitter que tiene como objetivo facilitar el diseño web mediante un diseño adaptable, es decir que el diseño se ajusta a cualquier dispositivo y tamaño de pantalla. (Spurlock, 2013)

JQuery

JQuery es una librería multiplataforma de código abierto para javascript creada por John Resig en el año 2006, su objetivo principal es simplificar la interacción con los documentos HTML mediante la manipulación del árbol DOM, el manejo de eventos, la interacción con AJAX, etc. (jQuery Foundation, 2016)

Apache

Apache es un servidor web HTTP de código abierto creado por Robert McCool en 1995. Apache se usa principalmente para enviar páginas web estáticas o dinámicas por internet. (Apache, 2016)

PHP

PHP por sus siglas en inglés “Hypertext Preprocessor” es un lenguaje de programación creado por Rasmus Lerdorf en 1995 que funciona del lado del servidor y fue diseñado para la creación de contenidos web dinámicos. (Sklar, 2016)

MySQL

MySQL es un sistema de gestión de base de datos relacional de código abierto desarrollado por Oracle y se usa generalmente para entornos de desarrollo web. (Tahaghoghi & Williams, 2006)