On the impact of the SHA-1 collider on Mexican digital signatures with legal binding

Luis J. Dominguez Perez[1], Laiphel M. Gómez-Trujillo[2], Nareli Cruz-Cortés[3] and Francisco Rodríguez-Henríquez[2]

[1] CONACyT, CIMAT, Zacatecas,
Mexico
luis.dominguez@cimat.mx

[2] CINVESTAV-IPN, Departamento de Computación, CDMX,
Mexico
francisco@cs.cinvestav.mx

[3] Instituto Politécnico Nacional, Mexico,
nareli@cic.ipn.mx

Abstract. Security warranties of the RSA-type digital signatures are based on two main hypothesis: First, in the assumption that factoring gigantic integer numbers is a computationally unfeasible problem. Second, in the assumption that hash functions produce a unique digest for any digital document. With these two hypothesis in mind, in the last decades in Mexico, and also in other countries, legislation has been enacted to legalize digital signatures. In Mexico, the combination of the RSA algorithm and the SHA-1 hash function can be used to legally validate digital contracts. This selection of algorithms is known as the RSA-SHA-1 digital signature. However, recently the SHA-1 hash function has suffered a falsification attack in which, given an arbitrary document for which the SHA-1 digest was produced, it is possible to generate a second document with the same digest. In other words, this attack permits to find arbitrary pairs of documents that share the same digest. This situation has provoked that the RSA-SHA-1 algorithm used to sign legal contracts is on risk to be broken. In this article, some of the repercussions in the information security of the legal documents signed with this protocol are discussed. We also discuss some countermeasures that can mitigate this vulnerability.

Keywords. SHA, digital signatures, RSA.

Sobre el impacto del colisionador SHA-1 en las firmas digitales Mexicanas con valor legal

Abstract. Las garantías de seguridad de las firmas digitales tipo RSA están basadas en dos hipótesis principales: La primera, en el supuesto de que la factorización de números enteros gigantescos es un problema computacionalmente difícil; La segunda, en el supuesto de que las funciones picadillo producen un digesto único para cualquier documento digital a firmar. Bajo estas hipótesis en las últimas décadas en México y en otros países, se han promulgado leyes que estipulan el uso de firmas digitales conformadas por la combinación del algoritmo RSA y la función picadillo SHA-1. Esta selección de algoritmos es conocida como la firma digital RSA-SHA-1. Dichas firmas digitales pueden utilizarse para validar contratos digitales de manera legal. Sin embargo, recientemente, la función picadillo SHA-1 ha sufrido un ataque de falsificación en el que dado un documento arbitrario y su digesto SHA-1, es posible generar un segundo documento con el mismo digesto. En otras palabras, el ataque permite encontrar parejas arbitrarias de documentos que comparten el mismo digesto. Esta situación ha provocado que el protocolo RSA-SHA-1, esté en riesgo. En este artículo, discutimos algunas de las repercusiones en la seguridad de la información de los documentos legales firmados mediante este esquema. También se presentan algunas contramedidas para mitigar esta vulnerabilidad.

Keywords. SHA, firmas digitales, RSA.

1. Introducción

El término “firma digital“ evoca para muchas personas el uso de algún medio electrónico para firmar un documento, por ejemplo el simple escaneo de una firma autógrafa y su inserción

como imagen en un documento digital podría ser equivocadamente interpretado como una firma digital. Otro ejemplo sería considerar que un usuario genera su firma digital cuando dibuja su firma autógrafa sobre una pantalla táctil con la ayuda de un lápiz electrónico o con su dedo. Este tipo de acciones no garantiza servicios de seguridad informática esenciales tales como la autenticación, integridad de datos y no-repudio. Grosso modo, el servicio de autenticación permite aseverar que un documento fue generado/enviado/firmado por una determinada entidad. Por su parte el servicio de integridad de datos garantiza que el documento digital recibido por la entidad destinataria es una copia genuina del documento original enviado por la entidad remitente. Finalmente, el servicio de no-repudio previene que un usuario malicioso intente repudiar o rechazar la autenticidad y/o validez de una firma digital que había sido previamente generada por éste de manera legítima.

En seguridad informática, una firma digital o firma electrónica es aquella que ha sido generada utilizando mecanismos criptográficos escogidos por una entidad a cargo y que se asocia a un documento para garantizar que el signatario o firmante autoriza o da fé del mismo. La presunta fortaleza del algoritmo criptográfico empleado permite determinar unívocamente a la entidad signataria. A diferencia de la firma autógrafa tradicional, cuyo único propósito es garantizar la autenticidad del signatario, la firma digital también brinda el servicio de integridad de datos, de manera tal que la mínima alteración en el contenido del documento digital anula automáticamente la validez de la firma.

En la práctica, las firmas digitales son implementadas mediante el uso de criptografía de llave pública. En este paradigma a cada usuario se le asigna un par de llaves, una de las cuales es su llave pública mientras que la otra es su llave privada. El RFC 8017 PKCS #1 versión 2.2, provee las especificaciones y primitivas para implementar criptografía de llave pública utilizando el algoritmo RSA [8], así como las propiedades matemáticas que deben exhibir ambas llaves. Además, el RFC 2313, PKCS #1 versión 1.5, especifica los algoritmos de firma con apéndice utilizados hoy en día.

Con el propósito de evitar ataques de usurpación de identidad, es necesario asociar rigurosamente la identidad del signatario a su llave pública. La forma de realizar este vínculo de una manera segura está especificada por el estándar RFC 3820 de infraestrutura de llave pública (PKI por sus siglas en inglés).[1] De acuerdo a dicho estándar, la llave pública de cualquier usuario debe ser avalada mediante un certificado digital emitido por una autoridad conocida como Autoridad Certificadora (AC).

Una vez que una entidad signataria ha firmado un documento con su llave privada (o también llamada llave secreta), cualquier tercero puede verificar la validez de la firma, que incluye tanto la identidad del signatario, como la integridad del documento correspondiente. Esta verificación se realiza utilizando la llave pública del signatario, que desde luego está a disposición de cualquiera.

2. Funcionamiento de las firmas digitales

La firma de documentos digitales se realiza mediante el uso del paradigma de criptografía de llave pública, que podríamos explicar de la siguiente manera: A cada ciudadano se le genera un par de llaves pública/privada y un certificado digital. Este último instrumento establece un vínculo entre el ciudadano y la llave pública que le fue asignada. El certificado digital es distribuido o publicado para el conocimiento de todos los usuarios del sistema, entonces bajo este escenario, el signatario puede firmar cualquier documento o contrato digital utilizando su llave privada. Una vez que los documentos firmados son recibidos por alguna entidad, ésta puede comprobar la validez de la firma utilizando la llave pública del firmante. A continuación se describen brevemente los principales bloques criptográficos necesarios para implementar un sistema de firma digital seguro.

[1] Public Key Infrastructure.

2.1. Huella digital

Las funciones de resumen o picadillo producen huellas digitales de longitud fija a partir de documentos de tamaño arbitrario. Cualquier variación en el documento original, por pequeña que ésta sea, produce una huella digital totalmente diferente a la que correspondería a una copia genuina de éste. De manera más formal, una función de resumen criptográficamente útil debe de ofrecer las siguientes características:

- Resistencia de pre-imagen.
- Segunda resistencia de pre-imagen.
- Resistencia a colisión.

La resistencia de pre-imagen implica que debe ser computacionalmente impráctico hallar la proyección inversa de una función de resumen. La segunda resistencia de pre-imagen se refiere a que dado un mensaje que se pretende falsificar, resulte impráctico obtener un segundo mensaje con el mismo resumen. Finalmente, la resistencia a colisión se refiere a la intratabilidad computacional de que dado un mensaje cualquiera, éste pueda ser modificado para obtener el mismo resumen. En el caso de la resistencia a colisión, un ataque exitoso sería el tomar cualquier documento y modificarlo para que tenga la huella digital del original. Por ejemplo, se podría intentar que variaciones sutiles en dos documentos diferentes produjesen la misma huella digital.

En seguida se describe el procedimiento de firma de documentos haciendo uso de las huellas digitales.

2.2. Firmas con cifrado de huellas digitales

En la Fig. 1 se ilustra un esquema de firma electrónica con cifrado de huella digital. Del lado izquierdo se muestra algún mensaje que se desea firmar, entonces utilizando una función resumen se obtiene su huella digital. Luego mediante el uso de algún cripto-esquema de llave pública, se cifra dicha huella digital utilizando la llave privada del signatario. Tanto el mensaje, como la firma (y la llave pública del signatario) se hacen llegar a una entidad verificadora que realizará un proceso de comprobación de la firma. La operación de

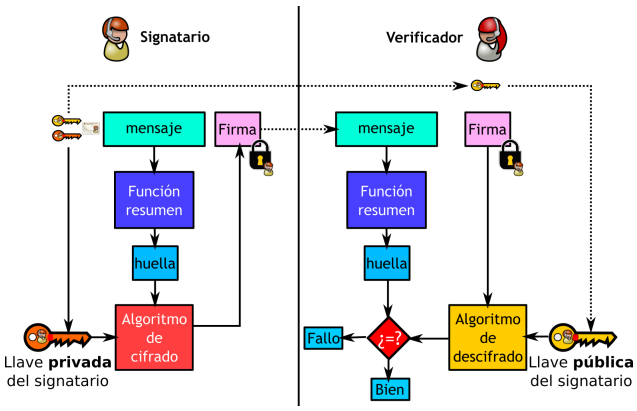

Fig. 1. Esquema general de firma electrónica tipo cifrado de huella digital.

verificación se muestra esquemáticamente en el lado derecho de la figura: Al documento recibido se le calcula su huella digital (tal y como se realizó en el proceso de firma), después se descifra la firma recibida del documento utilizando la llave pública del firmante. El resultado de esta última operación debería de coincidir con la huella digital del documento recibido, en cuyo caso el proceso de verificación es aceptado, de otra manera la firma se rechaza por considerarse falsa o alterada. Los procedimientos de firma electrónica y su verificación se describen en los Alg. 1, y Alg. 2, respectivamente.

Algoritmo 1 Firma por cifrado de huella digital

Entrada: Llave privada P y mensaje M a firmar
Salida: Mensaje firmado
{Se publica la llave pública P}
Huella$_H \leftarrow$ Resumen(Mensaje$_M$)
Firma$_F \leftarrow$ Cifrar$_P$(Huella$_H$)
return Mensaje$_M$ y la Firma$_F$

Algoritmo 2 Verificación de firma por cifrado de huella digital

Entrada: Llave pública Q, mensaje M y firma F a verificar
Salida: Mensaje verificado
  $\text{HuellaVerificación}_{H'} \leftarrow \text{Resumen}(\text{MensajeRecibido}_M)$
  $\text{HuellaRecibida}_{H''} \leftarrow \text{Descifrar}_Q(\text{FirmaRecibida})_F$
  if $\text{HuellaVerificación}_{H'} \equiv \text{HuellaRecibida}_{H''}$ then
    return Firma correcta, documento válido
  else
    return Firma incorrecta, documento no válido
  end if

Nótese que en el Algoritmo 2 se supone la posesión de la llave pública legítima del firmante. Para tener esa certeza de la identidad del firmante se hacen uso de los certificados digitales, como se explica a continuación.

2.3. Certificado digital

Un certificado digital es un instrumento que permite establecer un vínculo entre la identidad de un usuario o nodo del sistema y su llave pública. Dicho vínculo es construido a través de la firma digital de una entidad de confianza conocida como Autoridad Certificadora. La identidad de un usuario es descrita mediante un conjunto de datos que permiten identificar de manera unívoca al sujeto, que suele incluir el nombre del usuario, organización, dirección, entre otra información. En la práctica los certificados digitales se construyen de acuerdo al estándar RFC 6818 [5], en el que se especifica la estructura precisa que deberá tener un certificado digital X.509 versión 3.

En el contexto mexicano, el Instituto Nacional Electoral (INE) es el organismo encargado de asociar la identidad de un ciudadano con su correspondiente firma autógrafa a través de las credenciales para votar. De esa manera el INE da la certeza legal de que la firma impresa en la credencial para votar efectivamente pertenece al ciudadano. Para el caso de las firmas digitales, es la Secretaría de Hacienda y Crédito Público, mediante su organismo SAT, quien emite un Certificado Digital en donde se asocian los datos de identidad de los ciudadanos mexicanos con su llave pública correspondiente.

En el lado derecho de la Fig. 2 se ilustra cómo una persona o entidad envía su identificación (ID) y su llave pública a la Autoridad Certificadora, y cómo ésta emite un certificado digital que las asocia. Posteriormente el certificado está disponible en Internet para que cualquiera pueda descargarlo, como por ejemplo, una entidad verificadora de firmas de documentos. Note que la llave privada queda en posesión y resguardo del solicitante, y que el modelo de seguridad de este paradigma está basado en la suposición de que únicamente los dueños legítimos tienen posesión de sus llaves privadas.

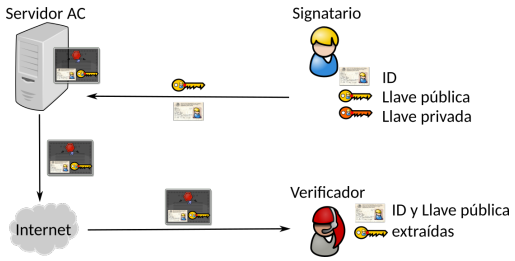

Fig. 2. Solicitud de certificado digital y Extracción de credenciales.

En el caso de utilizar certificados digitales, el Algoritmo 2 requeriría un proceso de obtención de la llave pública desde un certificado digital, como se muestra en la Fig. 2. Los certificados digitales se confeccionan de acuerdo a las directrices estipuladas en el estándar ANSI X.509 correspondiente al ya mencionado RFC 6818 [5]. A continuación se describe cómo el SAT utiliza los certificados digitales para proporcionar una validez legal a su proceso de firma/verificación de documentos.

3. Firmas con RSA en la práctica

En la práctica, considerando el estándar PKCS #1, versión 2.2 (RFC 8017) existen dos métodos de codificación para las firmas con apéndice: EMSA-PKCS1-v1_5 y EMSA-PSS.

El método EMSA-PKCS1-v1_5 es determinístico lo que significa que, dado un documento para firmar, su huella digital siempre será la misma, es decir dada la misma entrada la salida no cambiará. Esta firma tiene formato ASN.1 que incluye la huella del mensaje y un código de relleno al inicio formado por un número variable de ceros, al que se

conoce por el nombre de padding. Por otro lado, la estrategia EMSA-PSS añade un número aleatorio en su codificación, por lo que para dos (o más) ejecuciones sobre el mismo documento, la firma producirá salidas diferentes (el elemento aleatorio agregado es conocido como la "sal"). Es por ello que se considera un algoritmo probabilístico.

El codificado EMSA-PKCS1-v1_5 es mucho más sencillo de implementar que EMSA-PSS, sin embargo este útimo carece de una demostración formal de seguridad, esto es, no existe un análisis de seguridad matemático satisfactorio que lo califique como seguro, a pesar de que actualmente no existe ningún ataque documentado. Por otro lado, debe mencionarse que la demostración formal de la seguridad del esquema de codificado EMSA-PSS incluye el uso de la llave privada de manera exclusiva para la firma, esto es, que las claves RSA utilizadas para firmar, no se deberían de utilizar para cifrar documentos.

4. Firmas digitales en México

El organismo mexicano del Servicio de Administración Tributaria (SAT) ha promovido el uso de firmas electrónicas desde el 2005 como mecanismo de autenticación[2]. Inicialmente, la Secretaría de la Función Pública mexicana definió los requisitos que debería de cumplir dicha firma, sin embargo, en la práctica el SAT está a cargo de manejarla y administrarla ante el usuario final.

A principios del 2012, se emitió en México la Ley de Firma Electrónica Avanzada que dió certeza jurídica a la firma digital al hacerla equivalente con la firma autógrafa [2]. Este nuevo mecanismo digital permite simplificar los trámites administrativos que el ciudadano debe realizar, siempre y cuando cuente con los medios informáticos adecuados.

[2]Aunque ya desde el 2003 se habían hecho ajustes al Código de Comercio en México [3]

4.1. Firma con RSA, SHA-1 y el certificado del SAT

El procedimiento para firmar un documento utilizando el certificado del SAT con la combinación del cripto-esquema RSA y la función de resumen SHA-1, consiste de los siguientes pasos: Se supone que un ciudadano mexicano que realice sus declaraciones de impuestos tiene un certificado digital expedido por el SAT conocido como certificado FIEL. Este certificado digital utiliza el cripto-esquema RSA con llaves de longitud de 1024 o 2048 bits y contiene datos de la identidad del ciudadano, entre los que se encuentra el código del Registro Federal de Contribuyente (RFC), firma autógrafa, domicilio, huellas dactilares, etc. (estos datos son almacenados en los servidores del SAT). La asociación de las llaves con el ciudadano, se realiza utilizando la aplicación Certifica (antes conocida como Solcedi) o de forma personal acudiendo a las oficinas del SAT. Al final de este procedimiento, el contribuyente obtiene una llave privada y su certificado con la llave pública correspondiente, conocido como la e.firma.

Cada contribuyente puede utilizar su certificado para la firma de documentos con fines diferentes a la declaración de impuestos. Para ello, el contribuyente necesita el archivo de su llave privada (la cual es almacenada en un archivo con extensión ".key"), y la biblioteca OpenSSL, o alguna otra que incluya servicios criptográficos de llave pública. A continuación se muestra el uso de la biblioteca OpenSSL para RSA de 1024 bits. Para propósitos ilustrativos, en el siguiente ejemplo se supone que se desea firmar cierto documento que ha sido almacenado como "archivo.txt" (véase Listado 1).

```
1 openssl pkcs8 -inform DER -in
      Claveprivada_RFC_FIRMANTE.key -out
      Claveprivada_RFC_FIRMANTE.pem
2 openssl dgst -sha1 -sign
      Claveprivada_RFC_FIRMANTE.pem archivo.
      txt > firmabinaria.txt
3 rm Claveprivada_RFC_FIRMANTE.pem
4 openssl base64 -in firmabinaria.txt -out
      firma.txt
```

Listado 1. Firma RSA con SHA-1 con certificado

Después de ejecutar el Listado 1, la firma del documento archivo.txt es producida y almacenada

en el archivo firma.txt . En una fase posterior, el signatario deberá distribuir tanto el documento original como su firma e indicar que funge como la entidad signataria (véase Listado 2).

```
#Descargar Certificado FIEL del Signatario desde el SAT: https://portalsat.plataforma.sat.gob.mx/RecuperacionDeCertificados/
openssl x509 -inform DER -in RFC_FIRMANTE.cer -pubkey -noout > RFC_FIRMANTE.pem
openssl base64 -d -in firma.txt -out firmabinaria.txt
openssl dgst -sha1 -verify RFC_FIRMANTE.pem -signature firmabinaria.txt archivo.txt
```

Listado 2. Verificación de firma RSA con SHA-1 y certificado

Si la pareja archivo.txt/firmabinaria.txt corresponden al certificado del signatario, entonces aparecerá la leyenda "Verified OK" (o su equivalente en otro idioma). Enseguida se describen algunos problemas de seguridad informática que pueden llegar a presentarse con este esquema de firmado.

5. Requisitos de seguridad para las huellas digitales

Como se ha explicado anteriormente, la función de resumen recibe un documento de tamaño arbitrario para generar como salida una cadena de texto de tamaño fijo. Cualquier mínima variación en el documento de entrada produce digestos diferentes (cf. §2.1).

5.1. Función SHA

El Secure Hash Algorithm, o función SHA, es un algoritmo de una función de picadillo (o resumen) aprobado por el Instituto Nacional de Estándares y Tecnología (NIST) de los EE.UU., y la Agencia de Seguridad Nacional (NSA) de ese mismo país [6]. Fue utilizado como bloque básico del estándar para el Algoritmo de Firma Digital (DSA). Originalmente estaba basado en las funciones MD4 y MD5, y fue evolucionando a las funciones SHA-1, SHA-2 y recientemente a SHA-3 [7].

5.2. Función de resumen de dominio completo

El esquema de firma basado en RSA requiere la aplicación de la función de resumen al mensaje, seguida por la firma. Para ello el protocolo PKCS #1 (desde su versión 1.5) según el RFC 8017, recomienda que la salida de la función de resumen tenga una longitud en bits igual al tamaño del espacio de dominio de RSA. Esto es conocido como función de resumen de dominio completo (FDH por sus siglas en inglés). Por ejemplo, si utilizamos la función de reducción SHA-1 y RSA de 1024 bits, claramente no cumple con FDH, porque la salida de SHA-1 es de apenas 160 bits, mientras que el dominio de RSA es de 1024 bits. De hecho, aún para SHA-2 el esquema no cumple pues su salida es de sólo 256 bits.

Si el esquema de firma tiene dominio completo entonces existe una demostración que garantiza la seguridad teórica del esquema bajo ciertas condiciones[3].

6. Colisiones en las funciones de resumen

Las primeras versiones del estándar para la firma digital estaban basadas en la función de resumen llamada MD5. En 1996 se presentó un ataque que aunque no tenía consecuencias inmediatas en la práctica, sugería no utilizar MD5 en aplicaciones que requerían funciones de resumen con alta resistencia a colisiones. Así en el año 2008, un grupo de científicos encontró una colisión en MD5 en menos de 24 horas, utilizando un clúster relativamente modesto (algunos investigadores suponen que casi cualquier agencia de gobierno pudo haber realizado el ataque algunos años atrás).

Por su parte, la función SHA-1 fue sugerida como sustituto para MD5, sin embargo ésta fue desechada casi inmediatamente principalmente debido a la falta de confianza en sus garantías de seguridad. Ello abrió la puerta al uso de la función SHA-2, que incluso en algunos foros criptográficos había sido recomendado desde el año 2005.

[3] bajo un ataque conocido como selección adaptiva de mensajes en el modelo del oráculo aleatorio.

En el año 2014, se encontró una vulnerabilidad en SHA-1 basada en algunas modificaciones a sus especificaciones, lo que hace posible obtener resúmenes digitales idénticos de archivos diferentes. Debe señalarse que esta vulnerabilidad no puede considerarse propiamente como un ataque a la función SHA-1, debido a que requiere realizar modificaciones a los parámetros estipulados en su estándar. Sin embargo, si un atacante infectara la computadora de la víctima, entonces el ataque podría ser devastador. (véase `https://speakerdeck.com/veorq/sha-1-backdooring-and-exploitation`).

6.1. Ataques recientes a SHA-1

Recientemente en [11] se publicó un ataque práctico de colisión en contra de SHA-1. El prefijo de los mensajes a colisionar fue cuidadosamente escogido para que un atacante pudiera modificar dos documentos en formato PDF, de tal manera que a pesar de tener contenidos visuales diferentes (escogidos arbitrariamente), ambos documentos produzcan el mismo resumen. De esta manera, el ataque permite tomar un PDF original y un segundo falso cuyas modificaciones produzcan dos resúmenes idénticos.

Por ejemplo, es posible alterar un recibo digital por 500,000 pesos mexicanos, a un segundo recibo por 750,000 pesos mexicanos de tal manera que ambos documentos produzcan resúmenes digitales SHA-1 idénticos. El lector puede utilizar el sitio web [12] para experimentar con documentos escogidos arbitrariamente (aunque el sitio asegura el anonimato, se sugiere el uso de documentos digitales con información no confidencial).

6.2. Firma de documentos usando la FIEL del SAT y el ataque a SHA-1

Se utilizaron dos imágenes para producir los documentos con las características sumarizadas en la Tabla 1. Para obtener la huella digital se utilizó el comando: `openssl dgst -sha1 ARCHIVO`

Estas imágenes se subieron al Colisionador de SHA1 de Steven Weis [10] para generar dos documentos PDF diferentes. La Fig. 3, ilustra de manera general el proceso realizado.

Table 1. Tamaño y dimensiones de los archivos originales a collisionar

NOMBRE	TAMAÑO	DIMENSIONES
Lenna.jpg	48 kB	512×512 px
Huella Digital: b3742c9297ae855e5869656b5b61961e0d7ebebc		
LennaBW.jpg	35 kB	512×512 px
Huella Digital: 90fa50980f9b8f6772784f5521042a914942fe9b		

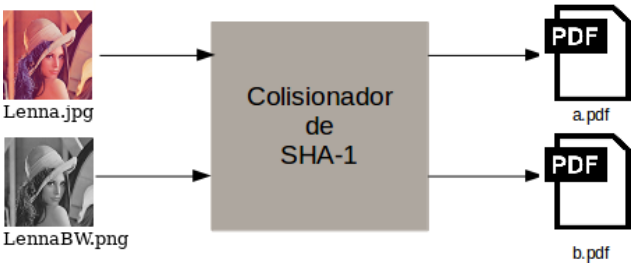

Fig. 3. Diagrama general del colisionador de SHA1 de Steven Weis

Para ejemplificar el ataque, se utilizó la foto Lenna `Lenna.jpg` como primera figura y la imagen en escala de grises `LennaBW.jpg` como segunda figura. El Colisionador genera dos PDF denominados a.pdf y b.pdf de manera tal que `a.pdf` (respectivamente `b.pdf`), es una versión modificada de la imágen `Lenna.jpg` (respectivamente `LennaBW.jpg`). Si se obtiene la función de resumen $H(m)$ de los documentos tipo PDF, con H siendo el algoritmo SHA-1, y $m = \{\texttt{a.pdf}, \texttt{b.pdf}\}$, entonces tenemos que $H(\texttt{a.pdf}) = \texttt{H}(\texttt{b.pdf})$, tal y como se puede apreciar en la Fig. 4.

Como fue descrito en la Sección 4.1, es posible utilizar la llave privada asociada a algún certificado del SAT, para producir la

```
$openssl dgst -sha1 a.pdf
SHA1(a.pdf)= 8d20b71773c6a615a39600b82666aea351ebbadf
$openssl dgst -sha1 b.pdf
SHA1(b.pdf)= 8d20b71773c6a615a39600b82666aea351ebbadf
```

Fig. 4. Huella digital obtenida de los dos documentos a.pdf y b.pdf

```
$openssl dgst -sha1 -verify RFC_FIRMANTE.pem -signature firmabinariaPDFa.txt a.pdf
Verified OK
$openssl dgst -sha1 -verify RFC_FIRMANTE.pem -signature firmabinariaPDFa.txt b.pdf
Verified OK
$openssl dgst -sha1 -verify RFC_FIRMANTE.pem -signature firmabinariaPDFb.txt a.pdf
Verified OK
$openssl dgst -sha1 -verify RFC_FIRMANTE.pem -signature firmabinariaPDFb.txt b.pdf
Verified OK
```

Fig. 5. Verificación de firmabinariaPDF.txt y firmabinariaPDF2.txt con a.pdf y b.pdf

firma de estos dos documentos diferentes que comparten una misma huella digital. La firma del documento `a.pdf`, correspondiente a la imagen Lenna original, fue almacenada en el archivo `firmabinariaPDFa.txt`, mientras que la firma del documento `b.pdf`, correspondiente a la fotografía en escala de grises, fue almacenada en el archivo `firmabinariaPDFb.txt`.

Al verificar con el certificado público correspondiente (con las instrucciones mostradas en la Fig. 5), la firma tanto del documento `a.pdf` como del documento `b.pdf` son auténticas. Es decir, que la firma `firmabinariaPDFa.txt` es válida para `a.pdf` y `b.pdf`, y asimismo también la firma `firmabinariaPDFb.txt` es válida para `a.pdf` y `b.pdf`.

7. Implicaciones del ataque a SHA-1 en México

La función resumen SHA-1 dejó de estar recomendada desde el año 2005. Sin embargo a partir del 2012 [4, 9], el SAT comenzó a utilizar certificados con la firma RSA-SHA-1 en sus certificados digitales para la emisión de facturas comerciales (7 años después de que ya no se recomendara su uso). Esta situación ha sido parcialmente corregida, ya que el SAT ha comenzado a migrar a RSA de 2048 bits combinado con SHA-2 para la emisión de sus certificados digitales; sin embargo, aún existen certificados digitales válidos que utilizan el SHA-1.

Afortunadamente, el ataque a la función resumen SHA-1, que es parte de las firmas RSA-1024 bits utilizadas durante casi una década por el SAT, requiere de la modificación de los metadatos (o las partes ilegibles) de la factura. Sin embargo, las facturas electrónicas que emite el SAT son los archivos con formato XML (Lenguaje de Marcado Extensible) generados por cada proveedor, mientras que el documento PDF es una representación gráfica de los mismos. Consideramos que la falsificación de una factura tendría que ser directamente sobre el archivo XML, mas al ser éste de un formato muy específico que requiere un proceso adicional de timbrado (o firma del SAT), se necesitarían estudios adicionales.

El problema actual radica en los contratos en formato PDF que se hayan sido firmados con un certificado del tipo RSA-1024 bits como los que utiliza en SAT para las facturas electrónicas, ya que estos son en principio, susceptibles de ser falsificados. Por otro lado, cualquier documento que se pueda exportar a formato PDF para su posterior firma, puede ser falsificado utilizando esta técnica. Actualmente el SAT no ha emitido ningún comentario sobre el uso de sus credenciales para firmar documentos diferentes a sus facturas. Aunque este ataque no aplica para la firma RSA de 2048 bits, que es la obligatoria para las nuevas credenciales que emite el SAT, de romperse la función SHA-2 el ataque se podrá replicar sin mayor esfuerzo.

8. Conclusiones

La firma digital se ha vuelto un estándar en muchos países, por lo que se ha comenzado a utilizar de manera oficial con vinculación legal. La firma basada en RSA-1024 no es la opción más sólida a nivel criptográfico, dado que es susceptible a ataques que pueden ser exitosos en la falsificación de documentos o contratos mercantiles.

Otra implicación importante es en la verificación de documentos compartidos en internet. Actualmente se tienen repositorios masivos de software, los cuales son descargados, instalados y ejecutados por usuarios de internet. Dichos repositorios suelen incluir tanto al programa ejecutable, como al código fuente para que el usuario lo compile, o lo pueda modificar. Para evitar modificaciones maliciosas, dichos repositorios utilizan como mecanismo de verificación las huellas digitales generadas por funciones resumen. En particular, los repositorios de código fuente GitHub, utilizan la función SHA-1 desde los años en que no se consideraba segura,

por lo que son vulnerables a ataques en los cuales un oponente malicioso puede modificar el código del repositorio, y la huella digital de verificación no podrá distinguir tales cambios. Esto ha sido duramente criticado, pero levemente atendido [14].

9. Trabajo a futuro

En Diciembre de 2008, Sotirov et al. [13] presentaron una construcción para generar una colisión MD5 con un ataque de prefijo seleccionado. El ataque consistió en la utilización de múltiples rutas de búsqueda de colisiones y en la posibilidad de seleccionar espacios de búsqueda en la parte alta y en la parte baja del rango de variables. Utilizando un clúster basado en procesadores IBM Power Cell, se pudo falsificar un certificado de una autoridad certificadora, mediante la inclusión de información no crítica del protocolo al certificado.

Debido a que la colisión a la función SHA-1 es similar, se podría considerar un ataque equivalente; sin embargo, en este caso el espacio de búsqueda se limitaría a al menos $2^{62.3}$ y $2^{65.1}$ como máximo, aunque se necesitarían más estudios para calcular los valores del precómputo. De ser exitoso el ataque, se podrían emitir certificados digitales falsos a nombre de cualquier AC que todavía esté basada en SHA-1 (como algunas del SAT y de Banxico [1]), esto es, suplantar sus funciones.

Agradecimientos

Se contó con apoyos del Programa de Cátedras CONACyT 2014/3163 y de Becas Nacionales.

Referencias

1. Banxico, . Certificados de la IES. http://www.banxico.org.mx/servicios/certificados-ies-firma-electr.html. Accedido el: 2019-03-12.
2. Calderón Hinojosa, F. (2012). Ley de Firma Electrónica Avanzada. Diario Oficial de la Federación. Miércoles 11 de enero de 2012. http://dof.gob.mx/abrirPDF.php?archivo=11012012-MAT.pdf.
3. Fox Quesada, V. (2012). Registro Público de Comercio. Diario Oficial de la Federación, reforma del 24 de octubre de 2003. http://dof.gob.mx/abrirPDF.php?archivo=24102003-MAT.pdf.
4. Gutiérrez Ortiz Mena, A., . Anexo 20 de la resolución miscelánea fiscal para 2011. Diario Oficial de la Federación, Martes 15 de junio de 2010. http://dof.gob.mx/abrirPDF.php?archivo=31052004-MAT.pdf.
5. (IETF), I. E. T. F. (2013). Updates to the internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. https://tools.ietf.org/html/rfc6818.
6. Information Technology Laboratory, N. (2017). Fips pub 180-4 federal information processing standards publication secure hash standard (shs). http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf. Accedido el: 2017-11-24.
7. NIST (2015). Hash functions. https://csrc.nist.gov/projects/hash-functions. Accedido el: 2017-11-24.
8. Rivest, R., Shamir, A., & Adleman, L. (1977). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, Vol. 21, No. 2.
9. Rojas Ibáñez, J., . Anexo 20 de la resolución miscelánea fiscal para 2012. Diario Oficial de la Federación, Viernes 20 de diciembre de 2011. http://dof.gob.mx/abrirPDF.php?archivo=30122011-MAT.pdf.
10. Steven, W., . SHA1 collider. https://alf.nu/SHA1. Accedido el: 2017-02-24.
11. Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). The first collision for full sha-1. Cryptology ePrint Archive, Report 2017/190. http://eprint.iacr.org/2017/190.
12. Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). Shattered. http://shattered.io/.
13. Stevens, M., Sotirov, A., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D. A., & de Weger, B. (2009). Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. Halevi, S., editor, Advances in Cryptology - CRYPTO 2009, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 55–69.
14. Torvalds, L., . The sky isn't failing. Google+, Sábado 25 de febrero de 2017. https://plus.google.com/+LinusTorvalds/posts/7tp2gYWQugL.

Article received on 16/12/2018; Accepted on 07/03/2019.
Corresponding author is Luis J. Dominguez Perez.