

Scheme for generating advanced electronic signature certificates at the Catholic University of Cuenca: A technical study based on applied cryptography models

Esquema de generación de certificados de firma electrónica avanzada en la Universidad Católica de Cuenca: Un estudio técnico basado en modelos de criptografía aplicada

Autores:

Vintimilla-Rodríguez, Telmo Ramiro
UNIVERSIDAD CATÓLICA DE CUENCA
Magister en Tecnologías de la Información
Maestría en Ciberseguridad
Cuenca – Ecuador



telmovintimilla@ucacue.edu.ec



<https://orcid.org/0000-0002-3326-5887>

Criollo-Bonilla, Ronald Raúl
UNIVERSIDAD CATÓLICA DE CUENCA
Magister en Sistemas de Información Gerencial
Maestría en Ciberseguridad
Cuenca – Ecuador



ronald.criollo@ucacue.edu.ec



<https://orcid.org/0000-0001-7103-6869>

Fechas de recepción: 25-JUN-2024 aceptación: 22-JUL-2024 publicación: 15-SEP-2024



<https://orcid.org/0000-0002-8695-5005>

<http://mqrinvestigar.com/>



Resumen

Este estudio presenta un esquema de generación de certificados de firma electrónica avanzada implementado en la Universidad Católica de Cuenca. El objetivo principal es fortalecer la seguridad en las transacciones electrónicas, optimizar la gestión de documentos digitales y reducir la dependencia de recursos físicos mediante la automatización de procesos. La metodología utilizada se basa en la combinación de algoritmos criptográficos SHA-256 y RSA 2048, con el estándar PKCS#1 v1.5 para la creación de un criptosistema de clave pública seguro y eficiente. Se llevaron a cabo pruebas exhaustivas para evaluar la robustez del esquema propuesto, incluyendo análisis de vulnerabilidades, simulaciones de ataques de fuerza bruta y resistencia a colisiones. Los resultados demostraron que el sistema es capaz de generar certificados válidos y seguros, con una alta resistencia a ataques. En una prueba de ataque de fuerza bruta, se logró determinar la contraseña del archivo PKCS#12 tras 1.950.258 intentos en 21 minutos, lo que resalta la importancia de contraseñas fuertes para la protección de certificados. Las conclusiones subrayan los beneficios de la implementación de firmas electrónicas en términos de autenticidad, no repudio e integridad de los documentos. Además, destacan la eficiencia operativa y la reducción del uso de papel, promoviendo un entorno de trabajo más sostenible. La incorporación de este esquema en la Universidad Católica de Cuenca no solo mejora la seguridad de las transacciones electrónicas, sino que también facilita la automatización del flujo de trabajo y la gestión documental, asegurando la integridad y autenticidad de los documentos en el entorno académico digital.

Palabras clave: Firma electrónica; Integridad; Modelos de criptografía; Certificados digitales; Seguridad; Gestión documental



Abstract

This study presents a scheme for generating advanced electronic signature certificates implemented at the Catholic University of Cuenca. The main objective is to enhance security in electronic transactions, optimize digital document management, and reduce dependence on physical resources through process automation. The methodology used is based on the combination of cryptographic algorithms SHA-256 and RSA 2048, utilizing the PKCS#1 v1.5 standard to create a secure and efficient public key cryptosystem. Comprehensive testing was conducted to assess the robustness of the proposed scheme, including vulnerability analysis, brute-force attack simulations, and collision resistance. The results demonstrated that the system is capable of generating valid and secure certificates, with high resistance to attacks. In a brute-force attack test, the password for the PKCS#12 file was determined after 1,950,258 attempts in 21 minutes, underscoring the importance of strong passwords for certificate protection. The conclusions highlight the benefits of implementing electronic signatures in terms of document authenticity, non-repudiation, and integrity. Furthermore, they emphasize operational efficiency and paper use reduction, promoting a more sustainable work environment. The incorporation of this scheme at the Catholic University of Cuenca not only enhances the security of electronic transactions but also facilitates workflow automation and document management, ensuring the integrity and authenticity of documents in the digital academic environment.

Keywords: Electronic signature; Integrity; Cryptography models; Digital certificates; Security; Document management



Introducción

La crisis sanitaria actuó como un catalizador que impulsó a las instituciones educativas a adoptar y aprovechar herramientas y tecnologías digitales para continuar con sus actividades educativas y operativas de manera efectiva. Ochoa et al. (2023) destacan que esta adaptación abrupta llevó a una “reacción inmediata de desarrollo e implementación de soluciones digitales” (p. 292).

En este sentido, Vintimilla-Rodríguez y Zhindón-Mora (2020), se refieren a la incorporación tecnológica como una “herramienta e instrumento para la automatización de procesos, con el objetivo de que esta sirva como apoyo y soporte operacional” (p. 422). Asimismo, Matovelle y Serrano (2019), expresan que la implementación de sistemas automatizados ayuda al “aumento de la productividad de los empleados, a la reducción de costos y como una alternativa para la reducción de errores” (p. 12), a la par resaltan la importación en la optimización de tiempos en la resolución de procesos.

Con la experiencia desarrollada como respuesta a la adopción repentina de las soluciones tecnológicas implementadas en ciertos procesos operativos institucionales, los soportes físicos tuvieron que ser reemplazados por electrónicos o digitales; sin embargo, esta solución temporal presentaba ciertos inconvenientes al momento de validar la autenticidad del documento o integridad de la información, debido a que contaban únicamente una firma digitalizada (rubrica escaneada) por parte de su autor.

En otros procesos, se desarrolló un certificado digital, que podía ser estampado en documentos electrónicos; pero, presentaba la misma limitación anterior, no podía ser validado de una forma que garantice su autenticidad, lo que podría generar incertidumbre al momento de procesar la información contenida en el documento. Esta solución permitía inserta un código QR generado por herramientas disponibles en línea, y estamparlo en el documento, el mismo que podía ser duplicado y colocado en otro documento.

Como resultado de lo expuesto anteriormente, se evidenció una reducción significativa en el uso de recursos físicos, así como una mejora notable en el desarrollo y ejecución de las actividades académico-operativas en los diferentes departamentos involucrados. Al retorno a la presencialidad, se retomaron los procesos tradicionales y el uso de recursos de soporte físico (hojas de papel) y con firma manuscrita (rubrica).

En consonancia, el compromiso de la Universidad Católica de Cuenca de transicionar hacia un ambiente digital más eficiente y sostenible, eliminando o reduciendo de manera



significativa el uso de medios de soporte físico para documentos, se propone la adopción de una política sin papel. No obstante, uno de los desafíos que podría obstaculizar su implementación es la obtención de certificados de firma electrónica por parte de la comunidad universitaria.

Esta dificultad se torna relevante al considerar que la institución cuenta con alrededor de 16.000 estudiantes en todos sus niveles educativos. El costo estimado de \$30 dólares americanos por un periodo de 2 años o \$70 dólares americanos para un certificado con vigencia de 5 años, emitido por una entidad certificadora, se convierte en una inversión significativa para los estudiantes. Este aspecto es especialmente destacado al considerar que la duración promedio de una carrera de tercer nivel o grado es de 5 años.

La certificación de la identidad de la aplicación ahorraría todos esos costes ya que se trata de procesos automatizados. Al ahorrar en costes de producción el impacto económico del Proyecto podrá ser beneficioso, ya que la creación de la aplicación tiene unos costes económicos mínimos al utilizar herramientas gratuitas para el desarrollo de esta (Núñez García-Bueno, 2022, p. 37).

Adicionalmente, es necesario realizar un análisis detallado de la documentación que los estudiantes debe firmar, esto implica identificar la documentación de archivo, es decir, si corresponde a información de comprobación inmediata o son documentos de apoyo administrativo, para posteriormente realizar la valoración documental, donde se determinará su importancia y disposición final.

El objetivo general del proyecto es definir un esquema de generación de certificados de firma electrónica en la Universidad Católica de Cuenca, mediante un análisis detallado y la aplicación de modelos avanzados de criptografía aplicada. Esto fortalecerá la seguridad en las transacciones electrónicas, mejorará la gestión de documentos digitales y reducirá los recursos físicos utilizados en los procesos. Los objetivos específicos incluyen evaluar y seleccionar el modelo de criptografía aplicada más adecuado para la universidad, desarrollar un prototipo funcional del esquema de generación de certificados de firma electrónica.

Con la implementación de la Autoridad Certificadora, todos aquellos documentos que sean generados serán firmados de manera digital, garantizando la validez del mismo, esto permitirá un flujo adecuado de los documentos digitales creados por las diferentes áreas, además de todo esto traerá consigo mismo un impacto muy importante a nivel ambiental (Ponciano et al., 2022, p. 71).



La iniciativa no solo busca fortalecer la seguridad en las transacciones electrónicas, sino también mejorar la gestión de documentos digitales, reduciendo la dependencia de recursos físicos y fomentando la sostenibilidad ambiental. Surgen interrogantes como: ¿Cuáles son los elementos de información críticos y necesarios que deben incluirse en un certificado de firma electrónica para garantizar su validez, autenticidad y seguridad en el contexto de la Universidad Católica de Cuenca? ¿Cuáles son los modelos de criptografía aplicada más adecuados para garantizar la seguridad en la generación de certificados de firma electrónica? ¿Cómo se puede optimizar la eficiencia en la gestión de documentos digitales mediante la implementación de este esquema? ¿Cuál es el impacto del esquema de firma electrónica en la seguridad de las transacciones electrónicas dentro de la Universidad Católica de Cuenca?

El propósito de este proyecto es encontrar soluciones tecnológicas que proporcionen a los estudiantes una herramienta para añadir validez y responsabilidad a documentos digitales fundamentales para sus actividades académicas, sin generar costos adicionales para ellos. La propuesta permitirá la integración de procesos que no podían ser considerados en la adopción de la política sin papel debido a la falta de certificados de firma electrónica. Produciendo cambios significativos en la gestión documental de la institución, los cuales constituyen los beneficios clave de la transición al entorno digital.

Metodología

Para la implementación del esquema de generación de certificados de firma electrónica se adoptó una metodología estructurada en varias fases, cada una de las cuales abordó diferentes aspectos técnicos y operativos del proyecto.

La primera fase consistió en una revisión sobre criptografía aplicada y tecnologías de firma electrónica. Se investigaron modelos avanzados y se analizaron las mejores prácticas. Este análisis permitió identificar los enfoques técnicos más eficaces y las tecnologías más adecuadas para la generación de certificados de firma electrónica.



Tabla 1

Revisión literaria sobre criptografía aplicada y tecnologías de firma electrónica

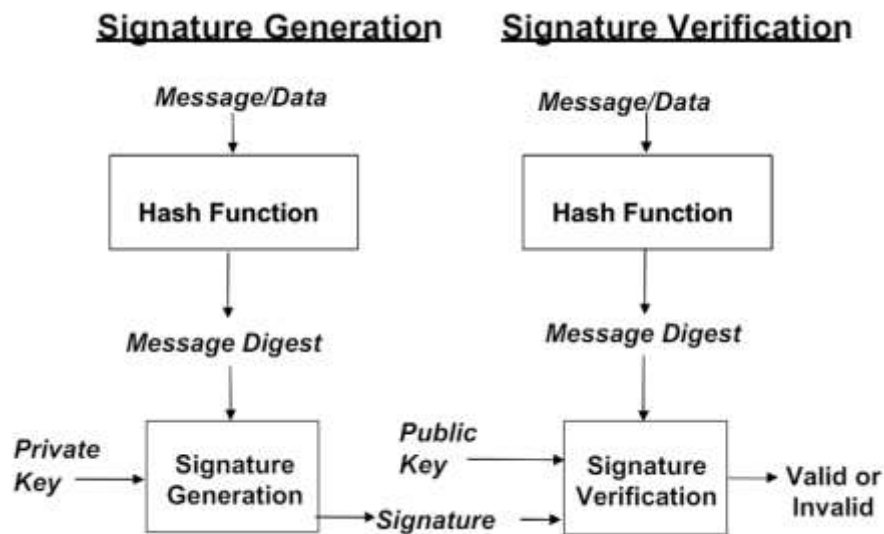
	Criptografía Asimétrica	Tecnologías de Firma Electrónica
Descripción	Criptografía que utiliza pares de claves (pública y privada) (Barker y Barker, 2019).	Herramientas y plataformas que permiten la firma digital de documentos electrónicos.
Algoritmos y estándares revisados	RSA (Rivest, Shamir y Adleman) / ECDSA (Elliptic Curve Digital Signature Algorithm) / DSA (Digital Signature Algorithm) (National Institute of Standards and Technology, 2023).	SES (Simple electronic signature / AES (Advanced electronic signature) / QES (Qualified electronic signature) (de Dueñas, 2022). PKI (Public Key Infraestructure) (Cedeño et al., 2020).
Seguridad	Alta seguridad mediante claves largas y resistencia a ataques matemáticos complejos.	Basada en criptografía asimétrica y certificados digitales que aseguran la autenticidad e integridad.
Propiedades relevantes	Confidencialidad, autenticidad, integridad, no repudio (Barker y Barker, 2019).	Autenticidad, integridad, no repudio, validez legal de la firma (QES) (Barker y Barker, 2019).
Ventajas	Clave pública puede ser distribuida libremente. Proporciona con confidencialidad, integridad y autenticación.	Simplifica la validación de documentos. Intercambio de archivos eficiente (Barker y Dang, 2015).
Desventajas	Requiere más recursos computacionales que la criptografía simétrica. Requiere manejo complejo de claves privadas.	Dependencia en servicios de terceros para la validación. Costos asociados a la emisión de certificado de firma por una autoridad certificadora.
Aplicaciones	Comunicaciones seguras, SSL/TLS, correo seguro. Autenticación de usuario, cifrado de datos.	Firmas de contratos, documentos legales, transacciones financieras. Validación de identidad en trámites electrónicos (Ley de Comercio Electrónico, Firmas y Mensajes de Datos, 2002).
Regulación y Normativas	NIST (National Institute of Standards and Technology), FIPS (Federal Information	Ley de Comercio Electrónico, Firmas y Mensajes de Datos (Ley de Comercio Electrónico, Firmas y Mensajes de Datos, 2002).



Los algoritmos de llave asimétrica, también conocidos como criptografía de clave pública, emplean un par de claves: una llave pública y una llave privada. La llave pública se utiliza para cifrar los datos, mientras que la llave privada, que se mantiene en secreto, se utiliza para descifrarlos. Los algoritmos de llave asimétrica, como RSA, DSA y ECC (Elliptic Curve Cryptography), son utilizados en aplicaciones que requieren la firma digital, el intercambio seguro de claves y la autenticación.

Figura 1

Proceso de firma digital.



Nota. Tomado de Digital Signature Processes (p. 9), por National Institute of Standards and Technology, 2023.

En la segunda fase, se seleccionaron los modelos de criptografía más adecuados para las necesidades de la universidad. Del proceso de revisión se adopta las recomendaciones de algoritmos y tamaño de claves (Tabla 2), del National Institute of Standards and Technology NIST, proponiendo que para la generación de claves de firma digital utilizadas para proporcionar un nivel de seguridad de autenticación y no repudio (Firma Electrónica Avanzada (Ramírez et al., 2020, p. 50)), se puede usar el algoritmo RSA con un tamaño de 2048 bits.

Tabla 2



Recomendaciones de algoritmos y tamaño de claves.

Key Type	Algorithms and Key Sizes
Digital Signature keys used for authentication (for Users or Devices)	RSA (2048 bits) ECDSA (Curve P-256)
Digital Signature keys used for non-repudiation (for Users or Devices)	RSA (2048 bits) ECDSA (Curve P-256 or P-384)
CA and OCSP Responder Signing Keys	RSA (2048 or 3072 bits) ECDSA (Curve P-256 or P-384)
Key Establishment keys (for Users or Devices)	RSA (2048 bits) Diffie-Hellman (2048 bits) ECDH (Curve P-256 or P-384)

Nota. Tomado de Recommended Algorithms and Key Sizes (p. 12), por Barker y Dang, 2015.

Complementariamente a la Tabla 2, en el mismo documento se expresa que la solidez del certificado de firma digital está directamente relacionada con la selección del algoritmo hash y el esquema de relleno, la Tabla 3 resume la recomendación en base a los tres aspectos antes mencionados.

Tabla 3

Recomendaciones de firma digital para CAs y Respondedores OCSP.

Public Key Algorithms and Key Sizes	Hash Algorithms	Padding Scheme
RSA (2048 or 3072 bits)	SHA-256	PKCS #1 v1.5, PSS
ECDSA (Curve P-256)	SHA-256	N/A
ECDSA (Curve P-384)	SHA-384	N/A

Nota. Tomado de Digital Signature Recommendations for CAs and OCPS Responders (p. 13), por Barker y Dang, 2015.

En base a lo antes mencionado, se definió el uso combinado de SHA-256, RSA 2048 y el método de encriptación para un criptosistema de clave pública PKCS#1 v1.5. Este estándar define las especificaciones para la implementación de criptografía RSA, incluye métodos para el cifrado para las claves y los esquemas de relleno (en este caso: deterministic). Dentro



de este estándar, existen dos esquemas de firma digital: PKCS#1 v1.5 y Probabilistic Signature Scheme (PSS) (Moriarty et al., 2016, p. 31).

X.509 versión 3 es un estándar para los certificados de clave pública, que especifica, entre otros, el formato de los certificados de clave pública. Incluye extensiones que permiten almacenar información adicional, como restricciones de uso de la clave y políticas de certificación, permitiendo una mayor flexibilidad y seguridad en la gestión de certificados (Cooper et al., 2008, p. 4).

SHA-256 es una función hash criptográfica de la familia SHA-2, que produce un hash de 256 bits. Es ampliamente utilizada para asegurar la integridad de los datos y la autenticidad de las transacciones, proporcionando una alta resistencia contra colisiones (de Dueñas, 2022, p. 26).

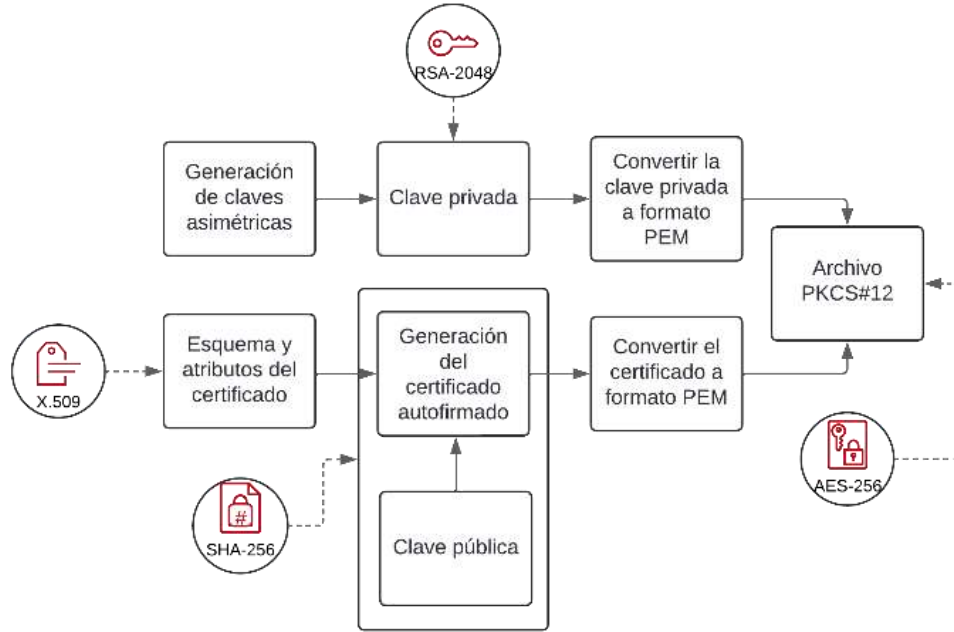
RSA 2048 es un algoritmo de cifrado asimétrico que utiliza una clave de 2048 bits para cifrar y descifrar datos. RSA es conocido por su seguridad y su capacidad para asegurar comunicaciones y transacciones digitales (de Dueñas, 2022, p. 55).

La combinación de estas tecnologías garantiza un equilibrio óptimo entre seguridad, rendimiento y compatibilidad, proporcionando una base sólida para la generación de certificados de firma electrónica y fortaleciendo la seguridad en el entorno académico digital de la Institución.

Figura 2

Esquema de generación de firma electrónica.





En la tercera fase, se procedió a diseñar y desarrollar un prototipo funcional del esquema de generación de certificados. Este prototipo incorporó los modelos criptográficos seleccionados y se implementó en un entorno controlado para garantizar la integridad del proceso. Se utilizó la biblioteca `cryptography` de Python, que emplea OpenSSL, para la generación de claves y la creación de certificados autofirmados.

Figura 3

Importación de módulos.

```
generate_pkcs12.py > ...
1 from cryptography.hazmat.primitives.asymmetric import rsa
2 from cryptography.hazmat.primitives import serialization, hashes
3 from cryptography.hazmat.primitives.asymmetric import padding
4 from cryptography import x509
5 from cryptography.x509.oid import NameOID
6 import datetime
```

El módulo `cryptography.hazmat.primitives.asymmetric.rsa` proporciona las herramientas necesarias para generar y manipular claves RSA. La serialización y deserialización de claves criptográficas a través del módulo `serialization`. La implementación de funciones hash con el módulo `hashes`. El módulo `asymmetric.padding` contiene esquemas de relleno para cifrado asimétrico. `cryptography.x509` proporciona clases y métodos especializados para trabajar con certificados X.509, mientras que `cryptography.x509.oid` incluye OIDs (Object Identifiers)

comunes utilizados en estos certificados. Finalmente, el módulo estándar datetime se utiliza para la manipulación de fechas y horas, ver Figura 3.

Figura 4

Generación de clave privada RSA.

```
7 # Generación de la clave privada RSA
8 private_key = rsa.generate_private_key(
9     public_exponent=65537,
10    key_size=2048,
11 )
```

Se comienza generando una clave privada RSA de 2048 bits mediante la función `rsa.generate_private_key()`, especificando el exponente público, que es el quinto número primo de Fermat ($F_4=2^{16}+1=65.537$) y el tamaño de la clave, ver Figura 4.

Figura 5

Definición de Nombre Distintivo (DN).

```
12 # Crear un nombre distintivo (DN)
13 subject = issuer = x509.Name([
14     x509.NameAttribute(NameOID.COUNTRY_NAME, u"EC"),
15     x509.NameAttribute(NameOID.STATE_OR_PROVINCE_NAME, u"Azuay"),
16     x509.NameAttribute(NameOID.LOCALITY_NAME, u"Cuenca"),
17     x509.NameAttribute(NameOID.ORGANIZATION_NAME, u"Universidad Católica de Cuenca"),
18     x509.NameAttribute(NameOID.COMMON_NAME, u"Telmo Vintimilla Rodríguez"),
19 ])
```

Luego, define un nombre distintivo (DN) para el titular del certificado (subject) y el emisor (issuer), utilizando atributos como el nombre del país, provincia, localidad, organización y nombre común, estructurados con la clase `x509.Name` y `x509.NameAttribute`, ver Figura 5.

Para crear el certificado autofirmado (Figura 6), se utiliza `x509.CertificateBuilder()`, donde se especifican los detalles del titular (`subject_name`), del emisor (`issuer_name`), la clave pública asociada a la clave privada generada, un número de serie aleatorio (`serial_number`), y fechas de validez mediante `not_valid_before` y `not_valid_after`. Además, se agrega una extensión `SubjectAlternativeName` para especificar un nombre alternativo de dominio (DNSName) (Cooper et al., 2008, p. 17).

Figura 6

Creación de certificado autofirmado.



```
20 # Crear un certificado autofirmado
21 cert = x509.CertificateBuilder().subject_name(
22     subject
23 ).issuer_name(
24     issuer
25 ).public_key(
26     private_key.public_key()
27 ).serial_number(
28     x509.random_serial_number()
29 ).not_valid_before(
30     datetime.datetime.utcnow()
31 ).not_valid_after(
32     # El certificado será válido por 1 año
33     datetime.datetime.utcnow() + datetime.timedelta(days=365)
34 ).add_extension(
35     x509.SubjectAlternativeName([x509.DNSName(u"ucacue.edu.ec")]),
36     critical=False,
37 ).sign(private_key, hashes.SHA256())
```

El certificado se firma utilizando la clave privada generada y el algoritmo de hash SHA-256 (hashes.SHA256()). Esto asegura la integridad del certificado y permite la verificación de su origen.

Figura 7

Conversión de clave privada y certificado a formato PEM

```
38 # Convertir la clave privada en formato PEM
39 private_key_bytes = private_key.private_bytes(
40     encoding=serialization.Encoding.PEM,
41     format=serialization.PrivateFormat.PKCS8,
42     encryption_algorithm=serialization.NoEncryption()
43 )
44 # Convertir el certificado en formato PEM
45 cert_bytes = cert.public_bytes(serialization.Encoding.PEM)
```

Después de construir el certificado, se procede a convertir la clave privada y el certificado en formato PEM (private_key_bytes y cert_bytes, respectivamente) utilizando métodos como private_key.private_bytes() y cert.public_bytes(), ver Figura 7.

Figura 8

Generación de archivo PKCS#12.

```
47 pkcs12 = serialization.pkcs12.serialize_key_and_certificates(  
48     name=b"Telmo Vintimilla",  
49     key=private_key,  
50     cert=cert,  
51     cas=None,  
52     encryption_algorithm=serialization.BestAvailableEncryption(b"ae123")  
53 )  
54 with open("certificado.p12", "wb") as f:  
55     f.write(pkcs12)
```

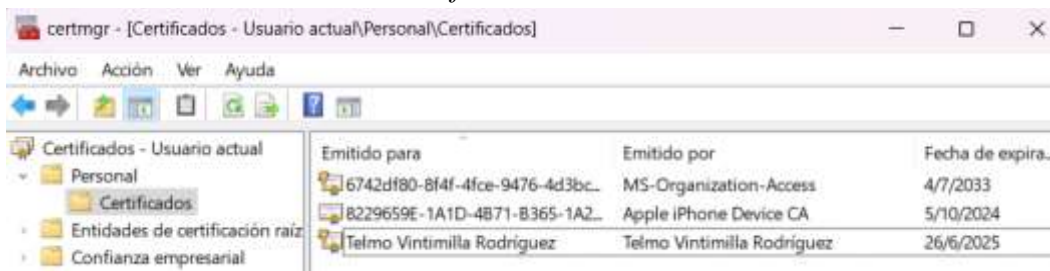
Finalmente, la clave privada y el certificado se combinan en un archivo PKCS#12 usando `serialization.pkcs12.serialize_key_and_certificates()`. Se especifica un nombre (`name`), la clave privada (`key`), el certificado (`cert`), y se elige un algoritmo de cifrado (`BestAvailableEncryption`) para proteger el archivo con una contraseña "ae123" (contraseña asignada con el propósito de realizar las pruebas de seguridad). Se procede a abrir el archivo `certificado.p12` para escritura en modo binario (`wb`), y se Escribe el contenido serializado en el archivo, `f.write(pkcs12)`, ver Figura 8.

Resultados

Como resultado de la ejecución del script, se obtiene un archivo PKCS#12 que se incorporó a un sistema operativo Windows utilizando la herramienta de gestión de certificados (Figura 9). El asistente de importación de certificados facilita una incorporación intuitiva donde solo se requiere ingresar la contraseña y seleccionar el almacén del certificado, que en este caso es la subcarpeta 'Certificados' dentro de la carpeta 'Personal'.

Figura 9

Administrar certificados de usuario – Windows

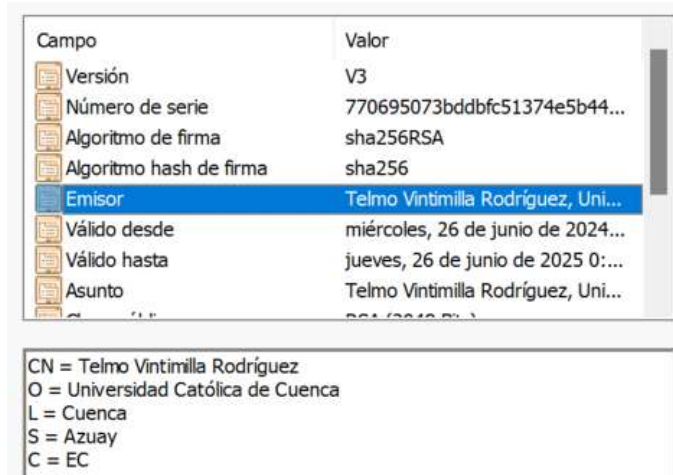


En el cuadro de diálogo correspondiente a los detalles del certificado (Figura 10), se pueden observar los campos del estándar x.509, tales como: la versión, el número de serie aleatorio, el tipo de algoritmo hash de firma, los atributos de nombre distintivo, el periodo de validez y la clave pública en formato hexadecimal (Figura 11).



Figura 10

Detalles del certificado.



Campo	Valor
Versión	V3
Número de serie	770695073bddd51374e5b44...
Algoritmo de firma	sha256RSA
Algoritmo hash de firma	sha256
Emisor	Telmo Vintimilla Rodríguez, Uni...
Válido desde	miércoles, 26 de junio de 2024...
Válido hasta	jueves, 26 de junio de 2025 0:...
Asunto	Telmo Vintimilla Rodríguez, Uni...

CN = Telmo Vintimilla Rodríguez
O = Universidad Católica de Cuenca
L = Cuenca
S = Azuay
C = EC

Figura 11

Visualización hexadecimal de la clave pública



Clave pública	RSA (2048 Bits)
Parámetros de clave pública	05 00

```
30 82 01 0a 02 82 01 01 00 85 cd cd 34 e0
1a b5 cb 38 31 62 cb eb 70 99 60 ff 89 26
c8 6b 13 03 a0 2d 05 f9 7d 99 7d 30 04 07
05 15 6a fa 4a 86 e3 08 d6 2c a8 56 9d 16
d7 a8 3c 2d 1b e1 5a 2d d7 1c 82 eb 85 a7
9c c8 6d 04 ba 36 76 c4 a8 a3 c5 16 e0 6e
79 e8 6d dc 09 85 d8 1d fe f4 a4 4a 24 b4
78 ee a1 69 45 b4 c6 fb 2c 51 fe 91 6c a4
2d 57 e0 5b 38 5b 84 2c 9b f4 c6 8b 99 05
7a 74 85 05 60 dd 0e 2b fe 65 45 66 58 f8
```

Es posible utilizar una aplicación de visualización de archivos PDF para firmar electrónicamente los documentos, añadiendo una estampa con los detalles del certificado, ver Figura 12.

Figura 12

Visualización estampa de firma electrónica.

DOCUMENTO DE PRUEBA

Telmo
Vintimilla
Rodríguez

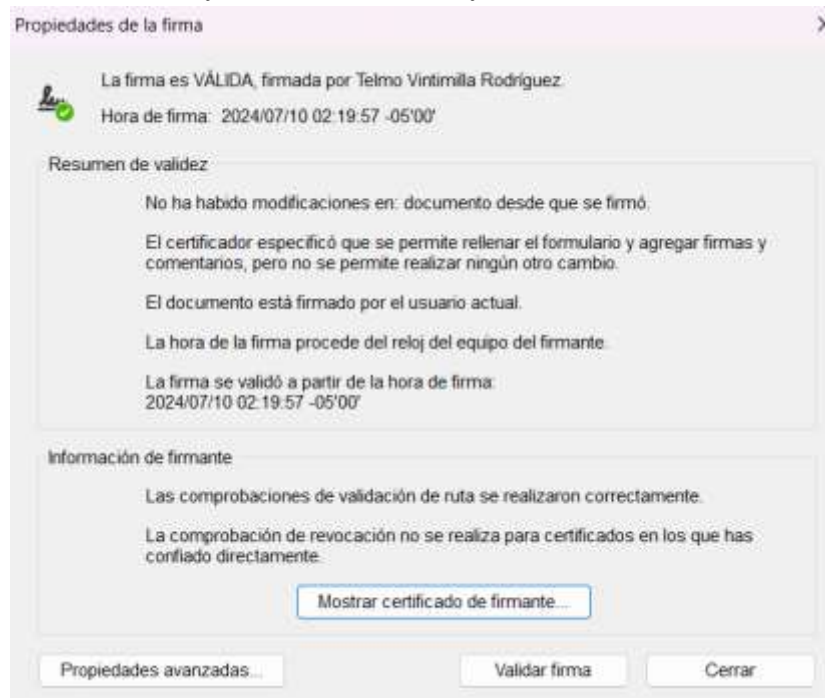
Firmado digitalmente
por Telmo Vintimilla
Rodríguez
Fecha: 2024.07.10
02:19:57 -05'00'

ESTAMPA DE FIRMA ELECTRÓNICA

Este certificado de firma es validable a través de aplicaciones de visualización similares a la utilizada para el proceso de estampado de firma electrónica (Figura 13). En estas aplicaciones se pueden visualizar detalles como: propietario de firma, hora de firma, integridad y autenticación. Es importante mencionar que para esto es necesario que el certificado de firma electrónica esté incorporado al sistema Windows.

Figura 13

Interfaz de validación de firma electrónica.



El mismo proceso puede ser ejecutado desde un script que permita verificar la firma electrónica y la validez del certificado. Utilizando la librería cryptography y el método `public_key.verify()`, se puede verificar la validez del documento firmado con SHA-256, relleno PKCS#1 v1.5 y la clave pública. Además, con el método `current_date =`



datetime.datetime.utcnow(), se comprueba si la fecha actual está dentro del periodo de validez del certificado, asegurando así que el certificado sigue siendo válido. En la Figura 14 se observa el resultado de la ejecución del código.

Figura 14

Resultado de script de verificación de firma.

```
PS C:\Users\Telmo Vintimilla\Desktop\UC-sing> & "C:/Users/Telmo Vintimilla/AppData/Local/Microso
ft/WindowsApps/python3.11.exe" "c:/Users/Telmo Vintimilla/Desktop/UC-sing/sing.py"
Certificado: <Certificate(subject=<Name(C=EC,ST=Azuay,L=Cuenca,O=Universidad Católica de Cuenca,
CN=Telmo Vintimilla Rodriguez)>, ...)>
Clave privada: <cryptography.hazmat.backends.openssl.rsa._RSAPrivateKey object at 0x000001F19A81
B550>
Firma verificada!
Certificado válido!
PS C:\Users\Telmo Vintimilla\Desktop\UC-sing> █
```

Se llevaron a cabo análisis de vulnerabilidades, simulando escenarios de amenazas realistas para evaluar la robustez del esquema ante posibles ataques. Las pruebas se centraron en la resistencia a colisiones y ataques de fuerza bruta.

En la primera prueba, se generó un script que permite crear de forma aleatoria un millón de cadenas de texto de 32 bits de longitud, a las cuales se aplica la función hash, con el objetivo de determinar si dos entradas diferentes pueden dar como resultado el mismo valor hash. En el ejercicio se generaron un total de 10 millones de cadenas sin encontrar ninguna colisión (Figura 15). Según Barker (2020), la probabilidad de colisión para hash de 256 bits es de 1 en 340 sextillones (p. 56).

Figura 15

Resultado de script de prueba de colisiones SHA-256.

```
bvWKRtCyzSS2GZ6mFNd34v0s1hqJHroa / 3c1b73bafeed30c44df6abdbd5522f1cbec0b24f5ea92e11cdba9e6ee78926e4
fZyUPEdX3KUMT1JP9qBV03qZGWRGYkh / 8676f5e7b689c20463210db0e38f25e1ed1c0336b7b0ce416a1ee6c6bf9b9e48
wCcHBU2ATiYf11XZn5FpKONTjhTnwZJ / d029ea809f717ef3f55c3abbfa8c400070acc2a992a6a2d137a5681e650fb19
pNGNx82y6qmyvoDmX8CaeLlaQgEgwPtm / 3f20a315f9deefc0d0a3e36dfd817567779b89d228c1e8d521ccbcc3bcbeba9
X1LlMgeVIq41e6XhnuBzMGd4Y5C0CIcpk / 411b4c01d6e24c041c3b775ab8528f510a43bc9cf83594858daae55983960bad
fP7HUHTUPOkaJe55K0k11Ip5B459Lqdf / 7e73f7aee7412b564181d2dbf4997a5b3d7999773e683180b7f1b6cc50dcd2ae
No se encontraron colisiones
```

En la segunda prueba, se intenta descryptar el archivo PKCS#12 con la generación de combinaciones secuenciales de cadenas de texto con un conjunto de caracteres básico (letras minúsculas y números), el bucle permite una permutación de hasta cinco caracteres. Mediante la ejecución del script se logró encontrar la contraseña en un total de 1.950.258 intentos, con una duración de 1266.17 segundos (aprox. 21 minutos) de tiempo de ejecución, ver Figura 16.



Figura 16

Resultado ataque de fuerza bruta AES-256

```
Intento #1950255: Contraseña probada 'ae120' (Tiempo transcurrido: 1266.17 seconds)
Intento #1950256: Contraseña probada 'ae121' (Tiempo transcurrido: 1266.17 seconds)
Intento #1950257: Contraseña probada 'ae122' (Tiempo transcurrido: 1266.17 seconds)
Intento #1950258: Contraseña probada 'ae123' (Tiempo transcurrido: 1266.17 seconds)
Contraseña encontrada: ae123
Contraseña crakeada exitosamente: ae123
```

Discusión

Las pruebas de vulnerabilidad realizadas demostraron la robustez del esquema. En la prueba de resistencia a colisiones, se generaron 10 millones de cadenas de texto aleatorias y se aplicó la función hash SHA-256. No se encontraron colisiones, confirmando la efectividad del algoritmo SHA-256 en evitar que diferentes entradas produzcan el mismo hash. Esta característica es crucial para mantener la integridad y autenticidad de los documentos firmados digitalmente, asegurando que cualquier alteración en el documento sea fácilmente detectable.

Por otro lado, la prueba de fuerza bruta reveló que la contraseña del archivo PKCS#12 fue encontrada después de 1.950.258 intentos, tomando 21 minutos. Este resultado subraya la importancia de utilizar contraseñas complejas y robustas para proteger los certificados digitales. Una contraseña débil puede comprometer la seguridad del certificado, permitiendo que un atacante acceda a la clave privada y, por ende, falsifique firmas electrónicas. Es esencial educar a los usuarios sobre la importancia de elegir contraseñas seguras y mantenerlas confidenciales.

Cuando se utilizan una función hash y un algoritmo de firma digital en combinación para calcular una firma digital, la seguridad de la firma está determinada por el más débil de los dos algoritmos. SHA-256 con RSA y una clave de 2048 bits, la combinación no puede proporcionar más de 112 bits de seguridad porque una clave RSA de 2048 bits no puede proporcionar más de 112 bits de seguridad, aunque SHA-256 puede admitir una seguridad de 128 bits (Barker, 2020, p. 58)

A pesar de los buenos resultados, se deben considerar algunas limitaciones. La dependencia del software para la implementación y validación de firmas electrónicas implica que cualquier falla en las herramientas utilizadas puede afectar la seguridad y la eficacia del proceso.



Además, es crucial contar con una infraestructura adecuada para gestionar los certificados y realizar las validaciones de manera eficiente. La capacitación continua del personal sobre seguridad digital y el uso correcto de las firmas electrónicas es vital para asegurar una implementación efectiva y segura del sistema en la universidad.

La implementación de certificados autofirmados presenta ventajas y limitaciones significativas que deben considerarse en su aplicación. Por un lado, los certificados autofirmados ofrecen una solución rápida y económica para entornos controlados y pruebas internas, permitiendo cifrar comunicaciones y garantizar la autenticidad de los datos dentro de una red interna o entorno de desarrollo.

Sin embargo, debido a la falta de validación por parte de una autoridad de certificación externa, estos certificados no son reconocidos automáticamente por los navegadores y otros clientes, lo que puede generar advertencias de seguridad y requerir la distribución manual del certificado a todos los usuarios involucrados. Por lo tanto, aunque los certificados autofirmados son adecuados para ciertos usos internos y específicos, su implementación debe ser cuidadosamente evaluada en función de las necesidades de seguridad y la interoperabilidad con sistemas externos y usuarios finales.

Conclusiones

La implementación de firmas electrónicas en la Universidad Católica de Cuenca asegura la autenticidad y el no repudio de los documentos. Las firmas electrónicas permiten verificar que los documentos provienen realmente del firmante, garantizando que este no pueda negar posteriormente haber firmado el documento. Esta característica es fundamental para mantener la confianza en la información contenida en los documentos, especialmente en un entorno académico donde la integridad de la información es crucial.

Las firmas digitales protegen los documentos contra cualquier alteración no autorizada. Los algoritmos de hash utilizados permiten detectar de inmediato cualquier cambio en el documento original, asegurando que los documentos mantengan su integridad desde el momento de la firma hasta su verificación final, proporcionando una capa adicional de seguridad en la gestión documental.

Al integrar el proceso de firma electrónica en los sistemas de gestión documental, se puede automatizar la validación y el archivo de documentos, lo que reduce significativamente el tiempo y los recursos necesarios para su gestión y verificación. Esto no solo mejora la



eficiencia operativa, sino que también facilita la implementación de políticas de seguridad de datos más robustas.

Finalmente, la implementación de un entorno de trabajo sin papel gracias a las firmas electrónicas no solo reduce los costos asociados con la impresión y almacenamiento de documentos físicos, sino que también contribuye a la sostenibilidad ambiental. Al adoptar estas iniciativas, las organizaciones pueden minimizar su huella ecológica, promoviendo prácticas más sostenibles y responsables con el medio ambiente. Este cambio hacia un entorno digital también facilita el acceso, almacenamiento, búsqueda y recuperación de la información, mejorando la eficiencia operativa general de la institución.



Referencias bibliográficas

Barker, E. (2020). Recommendation for Key Management: Part 1 – General. NIST Special Publication 800-57 Part 1 Revision 5, 1–171. <https://doi.org/10.6028/NIST.SP.800-57pt1r5>

Barker, E., y Barker, W. C. (2019). Recommendation for Key Management: Part 2 – Best Practices for Key Management Organizations. NIST Special Publication 800-57 Part 2 Revision 1, 1–91. <https://doi.org/10.6028/NIST.SP.800-57pt2r1>

Barker, E., y Dang, Q. (2015). Recommendation for Key Management Part 3: Application-Specific Key Management Guidance. NIST Special Publication 800-57 Part 3 Revision 1, 1–102. <https://doi.org/10.6028/NIST.SP.800-57Pt3r1>

Cedeño, C., Bolaños, F., Gregorio, A., y Saltos, W. (2020). Estudio exploratorio de la seguridad del DNI electrónico para su aplicación en Ecuador. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*, 4(1), 64–77. <https://doi.org/10.33936/isrtic.v4i1.2348>

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., y Polk, W. (2008). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Internet Engineering Task Force (IETF), RFC 5280, 1–151. <https://doi.org/https://doi.org/10.17487/RFC5280>

de Dueñas, D. (2022). Seguridad mediante Firma Digital y Certificados electrónicos [Universitat Oberta de Catalunya]. <https://openaccess.uoc.edu/bitstream/10609/145975/7/deduenasTFG0622memoria.pdf>

Galarza, D., Taco, A., Flores, V., y Sancho, J. (2019). Aplicabilidad de Algoritmos Criptográficos en firmas electrónicas en Ecuador. *Revista Científica Élite*, 1(2), 1–12. <https://revista.itsqmet.edu.ec:9093/index.php/elite/article/view/11>

Ley de Comercio Electrónico, Firmas y Mensajes de Datos, Pub. L. No. Ley 67, 1 (2002). <https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2012/11/Ley-de-Comercio-Electronico-Firmas-y-Mensajes-de-Datos.pdf>

Matovelle, D., y Serrano, E. (2019). Análisis, diseño e implementación de firmas electrónicas en documentos institucionales y la verificación mediante direccionamiento con códigos QR



para la Universidad Politécnica Salesiana [Universidad Politécnica Salesiana].
<https://dspace.ups.edu.ec/handle/123456789/18259>

Moriarty, K., Kaliski, B., Jonsson, J., y Rusch, A. (2016). PKCS #1: RSA Cryptography Specifications Version 2.2. En Internet Engineering Task Force (IETF) (Número RFC 8017, pp. 1–78). RFC Editor. <https://doi.org/10.17487/RFC8017>

National Institute of Standards and Technology. (2023). Digital Signature Standard (DSS). Federal Information Processing Standards Publication, FIPS 186-5, 1–131. <https://doi.org/10.6028/NIST.FIPS.186-5>

Núñez García-Bueno, J. (2022). Aplicación de Certificación de Identidad de un Usuario para la Entrada en Eventos [Universidad Politécnica de Madrid]. <https://oa.upm.es/71287/>

Ochoa, P., Álvarez, E., y Tinto, J. (2023). Desarrollo de habilidades para la transformación digital de las MIPYMES. Revista Conrado, 19(S1), 291–301. <https://conrado.ucf.edu.cu/index.php/conrado/article/view/3131>

Ponciano, Y., Telona, R., y González, J. (2022). Esquema para autenticación y validación de documentos electrónicos mediante una autoridad certificadora. Programación Matemática y Software, 14(1), 64–73. <https://doi.org/10.30973/progmat/2022.14.1/7>

Ramírez, E. M., Telona, R. E., y González, J. R. (2020). Prototipo de software para la firma y validación de documentos electrónicos. Programación Matemática y Software, 12(3), 49–57. <https://progmat.uaem.mx/progmat/index.php/progmat/article/view/2020-12-3-06>

Vintimilla-Rodríguez, T., y Zhindón-Mora, M. (2020). Data Mart para los estándares del componente estudiantado del modelo de evaluación externa CACES. Revista Científico-Académica Multidisciplinaria, 5(1), 418–442. <https://dialnet.unirioja.es/servlet/articulo?codigo=7659372>



Conflicto de intereses:

Los autores declaran que no existe conflicto de interés posible.

Financiamiento:

No existió asistencia financiera de partes externas al presente artículo.

Nota:

El artículo no es producto de una publicación anterior.

