Implementación segura de protocolos basados en criptografía de clave pública para Fundación Teletón

Secure Implementation of protocols based on public key cryptography for Fundación Teletón

Ernesto Ignacio Borbón Martínez A01701515, Gerardo Villegas Contreras A00571388,
José de Jesús Gutiérrez Aldrete A01637812, Luis Felipe Villaseñor Navarrete A01023976,
Pablo Ignacio Ochoa Sordo A01637785, Andrea Velázquez De Dios A01632556,
María Isabel Moreno Cornejo A01707706, Rodrigo Morales Aguayo A01632834

5 de mayo de 2022

Resumen

En la era digital, cada vez más organizaciones mudan sus procesos autógrafos o en papel a procesos electrónicos, y la Fundación Teletón es una de ellas. Las firmas digitales respaldan esta transición, y son de alta relevancia pues garantizan la validez y autenticidad de los documentos. Fundación Teletón busca optimizar el sistema de compras, autorizaciones y firmas electrónicas, consiguiendo que sus operaciones digitales sean seguras. Para ello se implementó un esquema de firma digital con múltiples participantes. Este sistema se auxilia de la librería cryptography de Python para la creación de claves y el firmado; cuenta con dos tipos de usuarios: regulares y administradores; y es capaz de firmar documentos, unificar las firmas de dos o más usuarios y verificar las firmas.

In the digital age, more and more organizations are moving their autograph or paper processes to electronic processes, and Fundación Teletón is one of them. Digital signatures support this transition, and are highly relevant as they guarantee the validity and authenticity of the documents. Fundación Teletón seeks to optimize the system of purchases, authorizations and electronic signatures, ensuring that its digital operations are secure. For this, a digital signature scheme with multiple participants was implemented. This system uses the cryptography Python library for creating keys and signing; It has two types of users: regular users and administrators; and it is capable of signing documents, unifying the signatures of two or more users and verifying the signatures.

Introducción

Hoy en día, es cada vez más evidente que las actividades que lleva a cabo cada individuo o una organización involucran la intervención de elementos relacionados con Seguridad Informática y Criptografía buscando contramedidas que no afecten el desempeño del algoritmo criptográfico utilizado, manteniendo así un protocolo criptográfico eficiente y seguro. El objetivo principal del código presentado es implementar protocolos de criptografía de clave pública para proteger ambientes que requieren rápido intercambio y almacenamiento de información. En el presente repositorio se presentará a la organización socio-formadora Teletón una implementación de firmado digital para documentos a través de algoritmos de criptografía de clave pública implementado en Python 3.3.

Métodos

En la elaboración de este sistema de firmas digitales, se utilizaron los siguientes elementos criptográficos.

Hash

MD5 para las contraseñas

Es un gran error de seguridad guardar las contraseñas de los usuarios en la base de datos en texto plano. Cualquier persona que consiga acceder a la misma, tendría las contraseñas de todos. Además, considerando las malas prácticas de ciberseguridad del público en general, es muy probable de que esta misma contraseña se utilice para otros sitios.

Es por esto que lo que se almacena es un hash de la contraseña del usuario y cada vez que se pide la contraseña al usuario, ésta se hashea para compararla con la almacenada en la base de datos. El algoritmo que utilizamos para hashear las contraseñas se denomina *Message-Digest Algorithm 5* (MD-5) que produce un resumen de hash de 128 bits. [Rivest, 1992]

SHA-256 para las documentos

Con el objetivo de poder firmar cualquier tipo de documento y eficientar el algoritmo, el programa hashea el documento a firmar con el algoritmo SHA-256 previo a su firmado. De este modo, lo que se firma es un resumen de hash de 256 bits en lugar de un documento de peso variable reduciendo el tiempo y los recursos utilizados para firmar y para verificar las firmas. [APPEL, 2015]

Curvas elípticas

La criptografía con curvas elípticas es una alternativa a criptosistemas de clave pública como RSA y ElGamal, con la gran ventaja de que mantiene el mismo nivel de seguridad, aunque tiene claves mas cortas. En este proyecto se emplea el algoritmo de firmado digital ED25519 que esta implementado en la librería cryptography de Python. El beneficio de utilizar una firma EdDSA bien construida es que podría llevar alrededor de 4 millones de años o requerir una computadora cuántica. [Josefsson and Liusvaara, 2017]

Resultados

¿Qué puede hacer nuestro sistema?

- Identificar a cada usuario con una clave privada y su clave pública correspondiente. Antes de crear un nuevo usuario el sistema revisa si ya existe un usuario la nueva clave pública generada. En caso de estar ya registrada en la base de datos se genera otra clave privada y se repite el proceso hasta encontrar una clave única para el usuario.

- Los certificados generados guardan las claves privadas encriptadas y tienen una duración de 1 año, después de este tiempo cualquier firma de un documento con este certificado será invalido.

- Antes de firmar cualquier documento el sistema es necesario que se ingrese la contraseña del usuario para desencriptar la clave privada y poder firmar documentos. Como mecanismo de doble autenticación, varias acciones solicitan volver a ingresar la contraseña, a pesar de que el inicio de sesión también la solicitó.

Interfaz

La página principal de la interfaz se como se muestra en la Fig. 1. Esta consta de una ventana en donde se arrastran o seleccionan los archivos con los que se operará, los tres botones principales (Verificar firma, Firmar y Unificar firmas), una barra con el menú de preferencias (donde se cambia el tema ya sea claro u oscuro), el menú de configuraciones y un previsualizador de archivos.

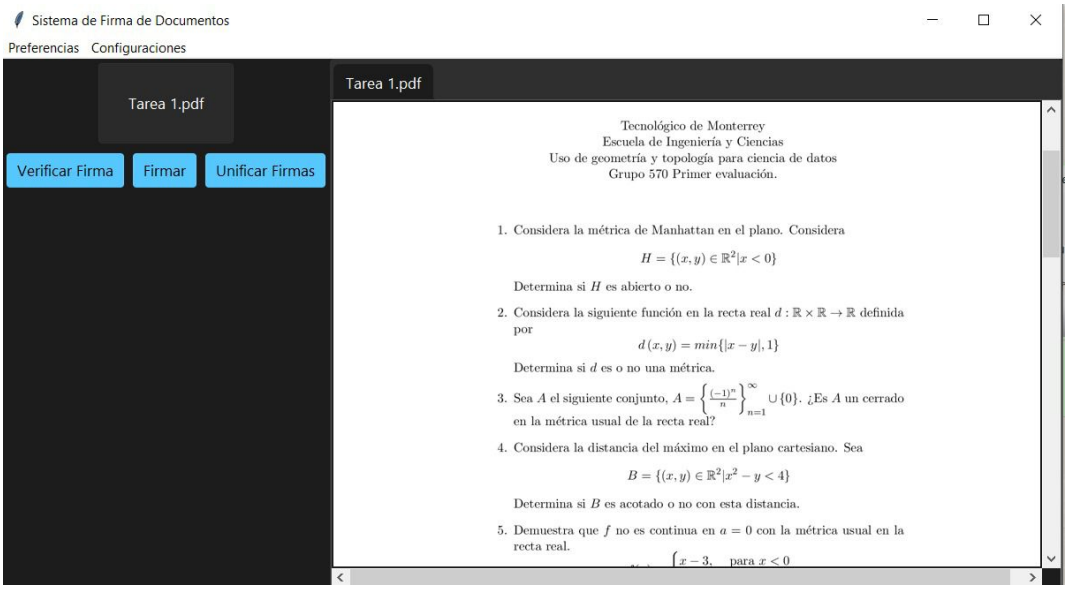

Figura 1: Página principal del sistema

Conclusiones

En conclusión se logro generar un programa el cual le permite al usuario firmar un documento, juntar su firma con la de otra persona de un mismo documento y verificar las firmas que obtuvo el usuario.

Debido al poco tiempo en el que se desarrollo el proyecto no se lograron implementar algunas características que hubieran mejorado la calidad del programa, como lo es la visualización de los usuarios no válidos y su validación, agregar mas usuarios tipos admin, cambiar la contraseña de un usuario en caso de que se le olvide y que se le notifique al usuario cuanto tiempo le quede a su certificado antes de expirar.

Anexo

Para conocer más del código del mecanismo y el almacenamiento de datos visita
https://github.com/ShoyChoy/Firma-digital-CRIT/blob/main/README.md

Referencias

[APPEL, 2015] APPEL, A. W. (2015). Verification of a Cryptographic Primitive: SHA-256.

[Josefsson and Liusvaara, 2017] Josefsson, S. and Liusvaara, I. (2017). Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032.

[Rivest, 1992] Rivest, R. L. (1992). The MD5 Message-Digest Algorithm. RFC 1321.