Instituto Tecnológico y de Estudios Superiores de Monterrey

Uso de álgebras modernas para seguridad y criptografía

Grupo 601

11 de marzo del 2024

Entregable 1: Implementación segura de esquemas de protección de datos personales con criptografía de clave pública.

Equipo 4:

José Manuel Dávila Mancilla A01732887

David Vázquez Moreno A01735864

Alfonso Elizondo Partida A01285151

Engels Emiliano Miranda Palacios A01423398

Gerardo Ramírez Chávez A01368693

Índice

Introducción

Sin lugar a dudas, el fenómeno migratorio en México está caracterizado por ser multidimensional debido a la naturaleza del país como eje de migración (El Sol de México, 2021). Desde aquellos que están de tránsito, como centroamericanos y los propios nacionales que migran hacia Estados Unidos, hasta los procesos de deportación por parte del gobierno. Sin dejar de lado los severos problemas de inseguridad.

Ahora, si bien todas las problemáticas anteriores son importantes y de fundamental necesidad su tratamiento, una que sobresale es la de los migrantes irregulares que atraviesan nuestro territorio. Esto debido a los terribles defectos sociales que salen a la luz al simplemente hablar de migrantes, así como la obligación humana que deberíamos tener por garantizar condiciones adecuadas para que otras personas alcancen su plenitud, todo dentro del marco planteado por la Comisión de los Derechos Humanos. (Prado y Gonzáles, 2023)

Con lo anterior en mente, puede que el lector esté familiarizado con que la travesía de estas personas está marcada por terribles circunstancias, como lo son la violencia en ruta debido al narcotráfico, abusos por parte de la población local, y la explotación a manos de redes de tráfico humano (Uribe y Hernández, 2024). Y es que los datos son terribles, de acuerdo con la Unidad de Política Migratoria, Registro e Identidad de Personas (UPMRIP, 2024), 540 migrantes irregulares reportaron haber sido víctimas de algún delito en territorio mexicano en 2023; de los cuales, el 45 % reportaron ser víctimas del tráfico ilegal de migrantes. Asimismo, resalta que el 12 % fueron niños víctimas de delitos varios. Todo esto es prueba de la criticidad del fenómeno migratorio y las espeluznantes situaciones que se enfrentan.

Sin embargo, no todo es malo, ya que poco a poco, han empezado a surgir grupos de ayuda para migrantes que tienen como objetivo acogerlos, protegerlos, e integrarlos. Tal es el caso de Casa Monarca, una organización asentada en la Ciudad Monterrey, Nuevo León; que apoya a las personas migrantes que transitan o buscan instalarse. De forma solidaria, Casa Monarca, les ayuda a cubrir las necesidades más acuciantes (alimento, ropa, calzado, medicina, orientación jurídica y acompañamiento) al momento de llegar a sus instalaciones. Indudablemente todas estas tareas son de bastante ayuda por cuenta propia, sin embargo, son especialmente cruciales en la Ciudad de Monterrey, ya que la sociedad neolonesa es considerada elitista y con características burguesas que suelen venir acompañadas de rechazo hacia los migrantes. De igual forma, al ser una entidad de costumbres e ideas muy conservadoras, el incremento de la migración puede generar temor en los habitantes de la región (Prado y Gonzáles, 2023).

Esto provoca en consecuencia que su seguridad sea prioritaria, no exclusivamente la física, pero la relacionada a su figura también. Aquí es donde nosotros, como estudiantes del Tecnológico de Monterrey, podemos hacer nuestro aporte al gran fenómeno migratorio, a través de la protección de su información

personal mediante la criptografía, garantizando también su integridad. A lo largo del presente proyecto, se desarrollará el estado actual de la criptografía, desde las tecnologías disponibles hasta las restricciones con fundamento legal. Se harán comparaciones entre diferentes algoritmos matemáticos de cifrado, mencionando sus ventajas, y posibles usos en el contexto que nos encontramos. Al final, se presentará una propuesta que integre todos los aspectos anteriormente mencionados, que tendrá como meta cuidar los datos proporcionados por los migrantes durante su estancia en Casa Monarca.

Estado del Arte y Marco Teórico

Teniendo esta problemática en cuenta, se deben considerar varios aspectos clave sobre el estado actual de la criptografía. En general, la criptografía es el arte de encubrir la información escrita empleando diversas técnicas que nos permiten proteger la información. Estas técnicas abarcan una variedad de métodos como lo son los algoritmos, las funciones Hashes, y firmas. Estas técnicas han ido cambiando considerablemente a lo largo de la historia (Amazon AWS, s.f.).

Históricamente, uno de los primeros usos de criptografía es con el reemplazo de símbolos tallados en la tumba de Khnumhotep II en Egipto. La escritura utilizaba jeroglíficos, pero no tenía como propósito ocultar el mensaje, sino que buscaban adornar la letra (Guseva, 2023). Así cómo los espartanos que, alrededor del siglo V a.C., que se escribían mensajes en un pergamino de cuero los cuáles solo se podían leer con una clave de descifrado. Dicha clave se obtenía con otra varilla del mismo tamaño y grosor que envolvía al pergamino. (BBVA, 2024). De las técnicas más antiguas, el cifrado César surgió durante el imperio romano. Este consistía en desplazar las letras del abecedario 3 posiciones. Este método fue utilizado para codificar mensajes en el ámbito militar y diplomático del Imperio Romano. Aunque hoy en día pueda parecer simple, este cifrado sentó las bases para el desarrollo de técnicas más avanzadas para garantizar la seguridad de información. (Guseva, 2023)

Un ejemplo de el como han ido cambiando las técnicas de criptografía es con la del Cifrado Vigenère. Este método se basa en el cifrado César, en donde se forma una tabla denominada de Vigénere la cual utiliza una serie de cifrados César que estaban entrelazados y basados en las letras de una palabra clave. Es el primer cifrado que utilizaba una clave de cifrado adecuada. El destinatario debía conocer la palabra clave acordada para decodificar el mensaje. (Guseva, 2023) El método original fue publicado por el criptólogo italiano Giovan Batista Belaso en 1553 en su publicación de ”La cifra del Sig. Giovan Batista Belaso”. Pero el mérito se le dió a Blaise de Vigénere en 1583. No fue hasta 1863 que se resolvió el cifrado por Friedrich Kasisk. (Universidad de Granada, s.f.) El siguiente cifrado reconocido fue el Cifrado Playfair, este fue creado en 1854 por un científico inglés llamado Charles Wheatstone. Este método consistía en usar pares de letras para hacer más complejo el descifrado de mensajes (National Cryptologic Foundation, 2019).

Con la aparición de múltiples conflictos y guerras durante el siglo XX, la criptografía tuvo un progreso notable. En 1917, el estadounidense Edward Hebern desarrolló la *Máquina Rotor Hebern*. Esta máquina es considerada como el primer dispositivo de cifrado electromecánica ya que fusionaba componentes de una máquina de escribir estándar y una máquina de escribir eléctrica, empleando un codificador con una llave incrustada en un disco giratorio. Su funcionamiento se basaba en una tabla de sustitución que se modificaba con cada nuevo carácter introducido, mejorando así la seguridad del cifrado (Guseva, 2023). Esta máquina tenía un defecto en el que se rompía el método al utilizar la frecuencia de las letras. Es por eso que se integró la clave en un disco giratorio que se ajustaba al teclado de la máquina de escribir. Tras cada pulsación de tecla, la tabla de sustitución, o alfabeto, experimentaba una ligera modificación. De esta manera, la máquina transformaba la sustitución estándar en una sustitución polialfabética similar al cifrado Vigènere, aunque sin necesidad de realizar una búsqueda manual del texto cifrado (THALES, 2023).

Después de solo 1 año, en 1918, el ingeniero alemán Arthur Scherbius inventó la Máquina Enigma. Esta máquina aplicaba los mismos métodos que la Máquina Rotor Hebern, pero esta utilizaba varios rotores en lugar de uno solo. El ejercito alemán comenzó a utilizarlo para enviar transmisiones codificadas.

Durante la Segunda Guerra Mundial, el cifrado Enigma fue finalmente descifrado por el criptógrafo polaco Marian Rejewski. Se creó una máquina llamada La Bomba, fue diseñada por el matemático inglés Alan Turing y fue construida por la *British Tabulated Machine Company*. Un elemento crucial en el proceso de decodificación fue la repetición constante de la frase "Heil Hitler" al final de cada mensaje cifrado, lo que permitió a los criptógrafos descifrar el contenido de cada mensaje interceptado (Guseva, 2023).

Otra contribución significativa fue hecha por la actriz estadounidense Hedy Lamarr. Durante la Segunda Guerra mundial Hedy y el compositor George Antheil elaboraron una tecnología de espectro ensanchado por salto de frecuencia basado en las señales de radio que guiaban a los torpedos de la armada norteamericana que eran muy fáciles de interceptar. Fue entonces cuando elaboraron un sistema de detección de torpedos teledirigidos. Esta tecnología no fue utilizada en la criptografía durante ese tiempo, pero sentó las bases para futuras comunicaciones inalámbricas seguras, para los sistemas de posicionamiento por satélite, como el GPS, y fue el del wifi (Sadurní, 2024).

A finales de la Segunda Guerra Mundial en 1945, Claude Shannon de Bell Labs publicó un artículo llamado *"Una teoría matemática de la criptografía"*, esto fue el punto de partida de la criptografía moderna. Sin embargo, durante los años de los 70s, el gobierno de Estados Unidos trataba el cifrado como cuestión de seguridad, las investigaciones eran clasificadas y su uso se le limitaba en guerras, diplomacia y espionaje (THALES, 2023).

A principios de los años 70, también se creó un grupo criptográfico por IBM, este grupo diseñó un cifrado de bloques para proteger los datos de los clientes, fue llamado "Lucifer". Después de algunas mejoras y varias versiones diferentes, se presentó en 1973 a la oficina de Estándares de Estados Unidos y fue aceptado como el *Estándar de Cifrado de Datos* o DES (Poston, 2019).

El estado del arte en criptografía

Desde la implementación por parte del National Institute of Standards and Technology (NIST) del DES, han surgido estándar tras estándar que con el fin de corregir los errores y vulnerabilidades de las versiones anteriores. De esa manera asegurando que el estado del arte en criptografía sea inquebrantable por la tecnología con la que contamos hoy día. Ya que mencionar a fondo el funcionamiento de estándares considerados obsoletos, no nos es útil para plantear cuales estándares siguen siendo parte del estado del arte actual, nos enfocaremos en desglosar aquellos que se utilizan todos los días en la industria.

Los principales métodos de cifrado utilizados hoy en día son los de clave simétrica y asimétrica. También existen otros métodos como la *Firma Digital*. Las principales diferencias entre estos métodos de cifrado son las siguientes:

- *Simétrica*: utiliza 1 sola clave para cifrar y descifrar los datos
- *Asimétrica*: utiliza 2 claves, una pública y otra privada
- *Firma Digital*: sello de autenticación de cifrado

Cada uno de estos esta diseñado con un uso especifico en mente, por lo que cada paradigma tiene sus ventajas y desventajas. Para poder encontrar dichas ventajas y desventajas, es necesario comparar y categorizar cada uno de los estándares por mencionar. En dicho análisis se tendrá que mencionar el objetivo de cada uno de estos, al igual que en sus características como los son: velocidad (tiempo de ejecución), costo computacional, tipo (simétrico, asimétrico, etc), tamaño de la clave, nivel de seguridad y qué tan utilizado es el estándar

Métodos de llave simétrica

Como fue mencionado anteriormente, existen diferentes métodos de cifrado, uno de estos son los de cifrado con llave simétrica, también denominado como cifrado de clave privada. Acorde a IBM (2022) los sistemas que utilizan cifrado de clave privada utilizan una única clave, que tiene que conocer tanto el destinatario como el remitente. Cabe destacar que previo al primer mensaje, ambas partes deben de conocer la clave, y para asegurarse de que la comunicación siga siendo privada ambas partes deben de cerciorarse de no compartir dicha clave.

Como se puede intuir, el cifrado de clave privada tiene un gran vulnerabilidad, que es el intercambio de llaves y su carencia de flexibilidad cuando se necesitan generar canales de comunicación privados de manera espontánea. Kaspersky (2024) menciona que si bien estos métodos cuentan con el antes mencionado "talón de aquiles" a la hora de compartir la clave. Es por eso que IBM (2022), no recomienda el uso generalizado de cifrado con este método, y nos provee con las siguientes razones.

- Se requiere una clave por cada par de entidades, por lo que el numero de claves necesarias incrementa rápidamente entre más canales de comunicación privados se quieran crear.
- Las llaves se deben de compartir entre cada par, por lo que la distribución de la clave incrementa la posibilidad del robo de la misma, haciendo el canal inseguro.
- Se requiere de un acuerdo previo al envío de un mensaje cifrado con este método, por lo que si surge de manera espontánea la necesidad de un canal de comunicación seguro no se puede conseguir tan rápidamente.

Tanto IBM (2022) como Kaspersky (2024) dejan claro los problemas que este método nos presenta, sin embargo, el cifrado de esta clase también cuenta con ciertas ventajas. Kaspersky (2024) menciona que este tipo de cifrado requiere de menos recursos computacionales y de tiempo, que un cifrado de clave asimétrica, al igual que tiene la ventaja que son resistentes a las computadoras cuánticas (hasta el momento se asume eso). Kaspersky (2024) también menciona que este tipo de cifrados suelen ser utilizados en conjunto con los asimétricos. Un ejemplo de esto, son los programas de mensajería, en los cuales se suele enviar la clave simétrica a través de un cifrado asimétrico para comenzar un canal de comunicación seguro.

Antes de comenzar a describir algunos de los estándares más utilizados, seria de utilidad mencionar que existen 2 tipos de algoritmos simétricos, los de bloque y los de flujo. Acorde a Kaspersky (2024), los algoritmos de bloque cifran la información utilizando bloques con una longitud fija (16, 32, 64, 128, 256 bits). En caso de que el mensaje o la parte final del mismo, sea menor a la longitud del bloque, el algoritmo agrega un padding para que se respete la longitud antes mencionada. En cambio los algoritmos de flujo, cambian cada bit de información utilizando un bit perteneciente a un flujo de llaves que fueron generados de manera pseudoaleatoria (en base a la clave privada), dicho flujo tiene la misma longitud que el mensaje a cifrar.

A grandes rasgos, el funcionamiento general de ambos tipos tiene varias similitudes y el mismo objetivo, cifrar los datos. Sin embargo, Grigutytė (2023) menciona que el cifrado de bloque es utilizado por la industria para cifrar contraseñas, archivos, bases de datos, discos (estructuras de datos de tamaño fijo); cabe destacar que también son el tipo preferido para la implementación de protocolos de comunicación. En cambio, el cifrado de flujo es utilizado para cifrar información que se actualiza en tiempo real, como las

conexiones a internet, el streaming de datos y comunicaciones inalámbricas.

Tomando esta información y el objetivo de nuestro reto, se vuelve obvio que los algoritmos de cifrado que nos interesan son los de que bloque. Es por eso que se investigaron algunas implementación ya probadas y utilizadas por la industria a nivel mundial. Cada uno de estos sera descrito de la manera antes planteada (velocidad, costo computacional, tipo, tamaño de la clave, nivel de seguridad, qué tan utilizado es el estándar):

AES

El estándar AES (Advanced Encryption Standard), es un algoritmo que fue desarrollado por el NIST con el propósito de sustituir al DES, ya que para 1997 se había vuelto muy lento y ya tenia vulnerabilidades conocidas (GfG, 2023a). El articulo del AES fue publicado por el NIST en el 2001, y desde entonces se ha vuelto uno de los estándares mas utilizados en la industria y hasta en operaciones gubernamentales.

Como ya sabemos, el AES es un algoritmo que pertenece a el cifrado de clave privada al igual que al cifrado por bloques. Por lo, tanto, podemos meternos directamente al funcionamiento interno de dicho algoritmo, ya que las antes mencionadas categorías de cifrado ya fueron explicadas.

Acorde a Oswald (2022) el AES tiene diferentes implementaciones, en las cuales cada una tiene un diferente tamaño de bloque, estos son de 128, 192 y 256. Cada uno de estos pasa por 10, 12 y 14 rondas correspondientemente. Para el AES cada una de estas rondas consiste de:

- **Sustitución:** El algoritmo reemplaza el texto plano con el texto encriptado basado en un cifrado predefinido.
- **Desplazamiento:** Todas las filas se desplazan una posición, excepto la primera.
- **Mezcla:** Otro cifrado, llamado cifrado de Hill, se utiliza para mezclar las columnas y evitar que alguien simplemente desplace las filas para comenzar a descifrar los datos.
- **Encriptación adicional:** Una pequeña porción de la clave de encriptación se utiliza para encriptar ese bloque de datos.

(Cabe destacar que el proceso de descifrado para el AES es el inverso del utilizado para cifrar.)

Oswald (2022) también menciona que AES-128 es recomendable, ya que ofrece un buen nivel de seguridad para la mayoría de las aplicaciones de consumo, mientras que AES-256 requiere considerablemente de más potencia de cómputo. En algunos casos, especialmente en aplicaciones gubernamentales, puede ser necesario utilizar AES-256 debido a la sensibilidad de la información, como lo requiere el gobierno de los Estados Unidos para información marcada como ”alto secreto” (Oswald, 2022).

Independientemente del tamaño de la clave, el AES se considera ampliamente seguro contra ataques

de fuerza bruta debido a sus múltiples capas de encriptación. Es por eso que este estándar es usado en diversas aplicaciones como:

- VPNs (Redes Privadas Virtuales)
- Administradores de contraseñas
- Aplicaciones móviles
- Redes inalámbricas
- Encriptación de archivos
- Videojuegos
- Encriptación de datos en discos duros
- Comunicaciones electrónicas, como a través de aplicaciones de mensajería
- Como parte de las bibliotecas de programación de Java, Python y C++
- Programas de compresión de archivos
- Navegadores de internet
- Encriptación de datos de agencias gubernamentales de los Estados Unidos

El AES no solo es seguro y una muy buena opción de cifrado, si no que también le proporciona flexibilidad a los desarrolladores de elegir entre velocidad de ejecución y nivel de seguridad. Aparte de esto, dicho estándar lleva desde el 2001 en la industria, y es un proyecto open-source, por lo que existen una gran variedad de recursos para su implementación.

TwoFish

Al igual que el AES, TwoFish es un estándar que surgió de la competencia AES del NIST en 1997 con el fin de sustituir al DES. El articulo de TwoFish fue publicado en 1998 (Schneier, s.f.), este comparte muchas características con el AES, que ya hemos mencionado en esta sección como el cifrado de clave privada, el cifrado de bloques y el uso de rondas de cifrado. Por lo tanto, ahora podemos explicar el funcionamiento de dicho algoritmo.

Acorde a Schneier (s.f.), TwoFish tiene una clave de tamaño variable que puede ir de los 128 a 256 bits, a continuación se ejemplificara como cifra y descifra dicho algoritmo. De acuerdo a Nagaraj (2023), el cifrado de TwoFish consiste en lo siguiente:

- **Expansión de Clave:** La clave de entrada se expande en un conjunto de claves de ronda utilizando el algoritmo de programación de claves.
- **Ronda Inicial:** El texto plano de entrada se divide en bloques y se aplica la operación XORecon la primera clave de ronda.
- **Transformación de Ronda:** Se realizan múltiples rondas de sustitución y permutación en los datos para proporcionar seguridad. En cada ronda, los datos se dividen primero en cuatro partes, y cada parte se transforma usando una combinación de operaciones de sustitución y permutación.

- **Ronda Final:** La salida de la última ronda se XORea con la clave de ronda final para producir el texto cifrado.

El descifrado consiste en los siguientes pasos (Nagaraj, 2023):

- **Expansión de Clave:** La clave de entrada se expande en un conjunto de claves de ronda utilizando el algoritmo de programación de claves.
- **Ronda Inicial:** El texto cifrado de entrada se divide en bloques y se XORea con la clave de ronda final.
- **Transformación de Ronda Inversa:** Se realizan múltiples rondas de sustitución e inversión de permutación en los datos para recuperar el texto plano. En cada ronda, los datos se dividen primero en cuatro partes, y cada parte se transforma usando una combinación de operaciones de sustitución e inversión de permutación.
- **Ronda Final:** La salida de la última ronda se XORea con la clave de la primera ronda para recuperar el texto plano.

Al igual que el AES este algoritmo es opensource por lo que hay una gran variedad de información del mismo disponible. Algo que es necesario mencionar, es que el AES ganó la competencia del NIST, por lo que se podría seguir considerando como la opción superior. Sin embargo, este sigue siendo reconocido por su seguridad y eficiencia, resistencia a ataques conocidos y ofrece una amplia gama de claves. Esto lo hace adecuado para aplicaciones de alta seguridad. Aun así, se ha encontrado que TwoFish es vulnerable a ataques de canal lateral y puede ser desafiante de implementar correctamente. Además, su complejidad computacional puede limitar su uso en dispositivos con recursos limitados, como lo es en el caso de nuestro reto (Nagaraj, 2023).

Métodos de llave asimétrica

Según Stohrer y Lugrin (2023) a diferencia de la encriptación simétrica que utiliza la misma clave para cifrar y descifrar datos, la criptografía de clave pública emplea un par de claves. Una de estas claves se utiliza para el cifrado y la otra para el descifrado. Para garantizar la seguridad del criptosistema de clave pública, Al igual que la mayoría de los criptosistemas públicos, la encriptación asimétrica se basa en funciones matemáticas unidireccionales. Esto significa que, aunque es fácil calcular el resultado a partir de los datos de entrada dados, es difícil recuperar los datos de entrada a partir del resultado.

Con la clave privada, que solo el destinatario conoce, es capas de desencriptar un mensaje encriptado por su clave publica que cualquier persona puede conocer, de forma que es computacionalmente inviable descifrar un mensaje sin conocer la llave privada (Computerphile, 2014).

Para que dos personas puedan entablar una comunicación dentro un criptosistema de clave pública de forma segura, cada persona debe contar con su par de claves públicas y privadas que debe ser validada por otros medios como una infraestructura de Clave Pública (PKI) (Stohrer y Lugrin, 2023).

Generalmente, no se utiliza la criptografía de clave pública para cifrar grandes cantidades de datos directamente, ya que esto suele ser computacionalmente más costoso que la encriptación simétrica. Sin embargo, es común utilizar la criptografía de clave pública para cifrar e intercambiar de forma segura las claves de los esquemas de encriptación simétrica. Las claves simétricas se utilizan entonces para el cifrado de datos en bloque. Esta combinación de criptografía de clave pública y encriptación simétrica se denomina encriptación híbrida (Wollinger y Kumar, 2006).

Este método de encriptación asimétrica ofrece una amplia gama de aplicaciones. Como se mencionó anteriormente, es una herramienta fundamental para garantizar la seguridad de las comunicaciones al cifrar mensajes, permitiendo que solo el destinatario autorizado pueda descifrarlos utilizando su clave privada. Además, este sistema también permite que cualquier persona pueda cifrar un documento utilizando su clave privada, permitiendo que la clave publica lo descifre, asegurando así la integridad y autenticidad del mensaje, lo que lo convierte en una sólida opción para la implementación de firmas digitales (Computerphile, 2014).

Otra aplicación destacada es la encriptación homomórfica, explicada por Ruiz (2023) como una técnica avanzada que permite realizar operaciones matemáticas directamente sobre datos cifrados, sin necesidad de descifrarlos primero. Esta capacidad es muy valiosa en el aprendizaje automático, ya que permite entrenar modelos y realizar operaciones estadísticas sobre datos sensibles mientras se mantiene su privacidad intacta.

Ventajas y Desventajas de la Encriptación Asimétrica:

Como lo menciona Panhwar, Ali, Panhwar, y Ali (2019) La encriptación asimétrica ofrece varias ventajas notables. En primer lugar, elimina la necesidad de compartir una clave secreta entre los comunicantes, lo que simplifica significativamente la gestión de claves y reduce los riesgos asociados con la distribución y protección de claves secretas. Además, el uso de claves públicas y privadas proporciona un mecanismo efectivo para garantizar la autenticidad y la integridad de los datos, lo que resulta fundamental en entornos donde la seguridad es primordial, como las transacciones financieras y la comunicación gubernamental.

Sin embargo, la encriptación asimétrica también presenta ciertas limitaciones. En primer lugar, los algoritmos asimétricos tienden a ser más lentos y computacionalmente intensivos que sus contrapartes simétricas, lo que puede afectar el rendimiento en aplicaciones que requieren un procesamiento rápido de grandes volúmenes de datos (Stohrer y Lugrin, 2023)..

Además, la seguridad de la encriptación asimétrica puede verse comprometida si se descubre una

vulnerabilidad en el algoritmo subyacente o si se roba la clave privada de un usuario. El tamaño relativamente grande de las claves asimétricas puede plantear desafíos en cuanto a almacenamiento y transmisión de claves, especialmente en entornos con recursos limitados(Wollinger y Kumar, 2006).

A continuación, se describirán los siguientes estándares de encriptación de la manera que fue planteada anteriormente.

ECC

El uso de curvas elípticas en criptografía proporciona una sólida base matemática para implementaciones seguras y eficientes de algoritmos criptográficos. Las curvas elípticas según Koblitz, Menezes, y Vanstone (2000) son definidas por la ecuación $y^2 = x^3 + ax + b$, son discretizables sobre campos finitos F_p, siempre que a y b pertenezcan a F_p y cumplan con la condición $4a^3 + 27b^2 \not\equiv 0(mod\ p)$. Estas curvas también contienen el punto al infinito, que actúa como la identidad en el grupo.

Los campos más convenientes para utilizar son los finitos de característica dos F_{2^m} (Koblitz y cols., 2000). Como lo explica Computerphile (2018) el grupo formado por la curva elíptica E sobre el campo F_p ($E(F_p)$) permite sumar dos puntos mediante la creación de una línea tangente a ambos puntos, encontrar su tercer cruce y tomar el punto homólogo bajo la reflexión con el eje X. Una ilustración más detallada se puede encontrar en la figura 1.

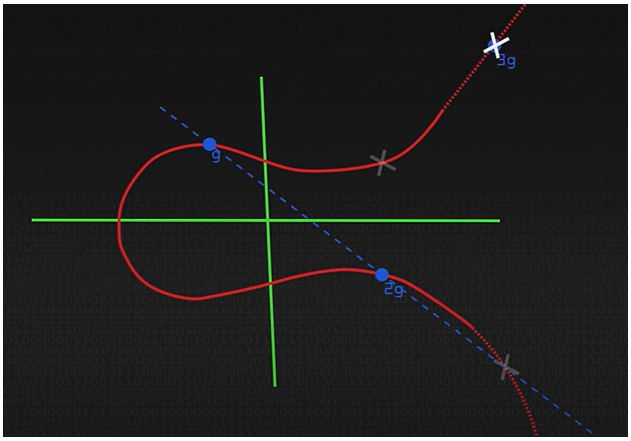

Figura 1*. Representación de la operación suma del grupo formado por $E(F_p)$ recuperada de Computerphile (2018)*

Uno de los algoritmos fundamentales que hace uso de curvas elípticas en criptografía es el *Elliptic Curve Digital Signature Algorithm* (ECDSA). Este algoritmo es una variante del conocido algoritmo DSA, pero aprovecha las propiedades de las curvas elípticas para reducir el tamaño de las claves y el tiempo de procesamiento, manteniendo un alto nivel de seguridad (Jurišic y Menezes, 1997).

Basándonos en los textos de Jurišic y Menezes (1997) La generación de claves ECDSA se realiza en los siguientes pasos:

1. Selección de una curva elíptica E definida sobre Z_p. El número de puntos en $E(Z_p)$ debe ser divisible por un primo grande n.
2. Selección de un punto $P \in E(Z_p)$ de orden n.
3. Selección de un entero aleatorio d tal que $2 \leq d \leq n-2$.
4. Cálculo de $Q = dP$.
5. La clave pública de A es (E, P, n, Q); la clave privada de A es d.

Como lo expresa Johnson, Menezes, y Vanstone (2001) para generar una firma ECDSA para un mensaje m, A sigue estos pasos:

1. Selección de un entero aleatorio k en el intervalo $[2, n-2]$.
2. Cálculo de $kP = (x1, y1)$ y $r = x1n$. (Aquí $x1$ se considera un entero, por ejemplo, mediante conversión desde su representación binaria). Si $r = 0$, se vuelve al paso 1.
3. Cálculo de $k^{-1}n$.
4. Cálculo de $s = k^{-1}(h(m) + dr)n$, donde h es el algoritmo de hash seguro (por ejemplo, SHA-1).
5. Si $s = 0$, se vuelve al paso 1.
6. La firma para el mensaje m es el par de enteros (r, s).

Para verificar la firma ECDSA de A (r, s) en m según Johnson y cols. (2001), B debe realizar lo siguiente:

1. Obtener una copia auténtica de la clave pública de A (E, P, n, Q). Verificar que r y s sean enteros en el intervalo $[1, n-1]$.
2. Calcular $w = s^{-1}n$ y $h(m)$.
3. Calcular $u1 = h(m)wn$ y $u2 = rwn$.
4. Aceptar la firma si y solo si $v = r$.

Diffie-Hellman

El algoritmo de Diffie-Hellman, un pilar fundamental en la criptografía de clave pública, se basa en la aritmética modular y la propiedad de ciclicidad de los campos de la forma $\frac{Z}{Z_p}$. Este algoritmo revoluciona la forma en que dos partes pueden establecer una clave de sesión compartida de manera segura sobre un canal de comunicación inseguro (Pound, 2017a).

Segun Panhwar y cols. (2019) el algoritmo Diffie-Hellman consta de dos partes principales: la generación de claves y el intercambio de claves. En la generación de claves, cada parte elige un número privado a y b, respectivamente. Luego, genera su clave pública utilizando un número primo grande p y un generador g, calculando $g^a p$ y $g^b p$, respectivamente.

El intercambio de claves se lleva a cabo enviando las claves públicas entre las partes, quienes las combinan con sus claves privadas para obtener una clave de sesión compartida. Esta clave de sesión se calcula elevando la clave pública del otro extremo a la potencia de su propia clave privada, es decir, $(g^b)^a p$ y $(g^a)^b p$, respectivamente. Sorprendentemente, ambas partes obtienen el mismo valor, que sirve como clave de sesión compartida para cifrar y descifrar los datos transmitidos Maurer y Wolf (2000).

Según Panhwar y cols. (2019) algunas ventajas del algoritmo de Diffie-Hellman

- Seguridad: La seguridad del algoritmo Diffie-Hellman radica en el problema del logaritmo discreto, que es intratable computacionalmente cuando se utiliza un grupo multiplicativo adecuadamente grande y un generador apropiado.
- Independencia de la clave: Las partes no necesitan compartir ninguna información previa para establecer la clave de sesión, lo que reduce significativamente los riesgos de seguridad asociados con la transmisión de claves.
- Facilidad de implementación: El algoritmo Diffie-Hellman es relativamente simple de entender e implementar, lo que lo hace ampliamente utilizado en una variedad de aplicaciones.

Desventajas del algoritmo de Diffie-Hellman basandonos en Pound (2017b):

- Vulnerabilidad a ataques de intermediarios: Aunque el algoritmo garantiza la confidencialidad de la comunicación entre las partes, es vulnerable a ataques de intermediarios, donde un atacante intercepta y modifica los mensajes durante el intercambio de claves.
- Posibilidad de ataques de hombre en el medio: Si un atacante puede interceptar y modificar el intercambio de claves, puede comprometer la confidencialidad de la comunicación o realizar ataques de tipo "hombre en el medio".

- Consumo de recursos computacionales: La generación y el intercambio de claves públicas pueden ser computacionalmente intensivos, especialmente en dispositivos con recursos limitados, lo que puede afectar el rendimiento en entornos sensibles al tiempo.

A pesar de estas limitaciones, el algoritmo Diffie-Hellman proporciona un medio seguro y eficiente para el intercambio de claves en entornos donde la confidencialidad es fundamental, tiene un sin numero de aplicaciones en la comunicación segura de internet y es comúnmente combinado con otros métodos de encriptación asimétrica para mitigar las vulnerabilidades que tiene como en el caso de (Islas-Mendoza, Jiménez-Vázquez, Silva-García, y Flores-Carapia, 2013).

Algunos ejemplos de uso es en la comunicación segura de internet es el protocolo de comunicación TLS, que puede usar un método de firma digital como el ECDSA para enviar la parte publica del mensaje junto con una firma que nos ayuda a tener seguridad de la integridad y fuente del mensaje que después nos permite usar el algoritmo de Diffie-Hellman para generar una sesión efímera de comunicación segura Islas-Mendoza y cols. (2013).

Métodos de Firma Digital

Los estándares y algoritmos antes mencionados, tenían el propósito de cifrar información, sea para su almacenamiento seguro o su envió de a través de un canal privado. Sin embargo, los algoritmos de Firma Digital, no cifran la información del mensaje, si no que se aseguran que la información en cuestión no haya sido manipulada por un tercero (IBM, 2021). Esto es posible, ya que las herramientas criptográficas utilizadas, generan un hash único para esa configuración especifica de bits que componen a un archivo o mensaje, por lo que cualquier cambio, resultaría en un hash diferente. Un ejemplo de como implementar un algoritmo de este tipo en un flujo seria el siguiente:

- El remitente calcula un resumen del mensaje y luego cifra el resumen con su clave privada, formando así la firma digital.
- Esta firma se adjunta al mensaje y se transmite al receptor.
- El receptor descifra la firma digital con la clave pública del remitente, regenerando así el resumen del mensaje.
- Luego, el receptor calcula un resumen del mensaje recibido y verifica que los dos resúmenes coincidan.
- Si coinciden, el mensaje está intacto y auténtico.

Cabe destacar que para que la firma digital sea válida, debe cumplir con ciertos criterios, como tener un certificado actual y emitido por una autoridad de certificación confiable, y que el editor sea de confianza (IBM, 2021). Aparte de esto, se puede complementar la firma digital con otro estándar de encriptación para

cifrar los datos y poder autentificar su integridad sin poner en riesgo la seguridad de los datos en cuestión.

Digital Signing Standard (DSS)

La firma digital es una forma de autenticar datos provenientes de una fuente confiable, garantizando la integridad y el origen de los documentos digitales. El *Estándar de Firma Digital* (DSS, por sus siglas en inglés) definido por el *Estándar Federal de Procesamiento de Información* (FIPS) utiliza algoritmos como el *Algoritmo de Hash Seguro* (SHA) para generar firmas digitales. A diferencia de los métodos de cifrado, el DSS se enfoca únicamente en la generación de firmas(GfG, 2023b).

En el enfoque del DSS, se genera un código hash a partir del mensaje, junto con un número generado aleatoriamente k, la clave privada del remitente ($PR(a)$), y una clave pública global ($PU(g)$). Estos datos producen una firma que contiene los componentes s y r, que luego se concatenan con el mensaje original y se envían al destinatario (GfG, 2023b).

En el extremo del destinatario, se verifica la autenticidad del remitente generando un código hash a partir del mensaje recibido. Una función de verificación toma este código hash, junto con los componentes de la firma s y r, así como la clave pública del remitente y la clave pública global. Si la salida de la función de verificación coincide con el componente de la firma r, la firma se considera válida(GfG, 2023b).

Si bien este algoritmo (y como ha sido mencionado) no tiene el objetivo de cifrar información, los beneficios y desventajas de este varían significativamente a los antes mencionados, por lo que es necesario mencionar algunas de estas ventajas y desventajas.

Beneficios:

1. Mejora la seguridad en las transacciones, previniendo fraudes de partes no autorizadas.
2. Facilita el seguimiento del estado de los documentos.
3. Agiliza la entrega de documentos.
4. Legalidad garantizada por autoridades de certificación aprobadas por el gobierno.
5. No repudio de los documentos firmados.
6. Marca de tiempo automática en los documentos firmados.
7. Prevención de manipulación de documentos.
8. Identificación del firmante.
9. Prevención de fraudes.

Desventajas:

1. Pueden surgir problemas de compatibilidad, requiriendo controladores y software actualizados.
2. La compatibilidad del software es una preocupación importante.
3. Las entidades corporativas pueden necesitar

obtener firmas digitales para etiquetado electrónico.

4. Riesgo de pérdida o robo de claves y uso de métodos de almacenamiento débiles.
5. Necesidad de estándares de interoperabilidad.
6. Corta vida útil de los productos tecnológicos.
7. Tanto los comerciantes como los destinatarios pueden necesitar comprar certificados digitales.
8. Costo del software de verificación.
9. Costos de implementación para las empresas.

RSA

El algoritmo RSA (Rivest-Shamir-Adleman) es un algritmo de encripación asímetrica que hace uso de la aritmetica modular y se puede aplicar a firmas digitales basandonos en Wollinger y Kumar (2006) sabemos que usa el siguiente algoritmo:

1. Generación de claves:

1. Se eligen dos números primos grandes distintos, p y q.
2. Se calcula el producto de estos dos números primos, $n = p \times q$, que se convierte en el módulo para el cifrado y el descifrado.
3. Se calcula la función de Euler de n, denotada como $\phi(n)$, que se define como $\phi(n) = (p-1) \times (q-1)$.
4. Se elige un número entero e tal que $1 < e < \phi(n)$ y e sea coprimo con $\phi(n)$, es decir, su máximo común divisor sea 1. e se convierte en la clave pública de cifrado.
5. Se calcula el inverso modular de e módulo $\phi(n)$, denotado como d. d se convierte en la clave privada de descifrado.

2. Cifrado de un mensaje:

1. El mensaje que se desea cifrar se convierte en un número entero m, donde $0 < m < n$.
2. El mensaje se cifra utilizando la clave pública e mediante la siguiente fórmula: $c = m^e n$. El valor cifrado c es el mensaje cifrado que se envía al receptor.

3. Descifrado del mensaje:

- El receptor recibe el mensaje cifrado c.
- Utiliza su clave privada d para descifrar el mensaje mediante la siguiente fórmula: $m = c^d n$.
- El valor m obtenido es el mensaje original descifrado.

Es importante destacar que la seguridad del algoritmo RSA se basa en la dificultad de factorizar el producto de dos números primos grandes p y q. La clave privada d es esencialmente el inverso modular de la clave pública e, y solo se puede calcular eficientemente si se conoce la factorización de n, lo cual es extremadamente difícil de realizar para números primos muy grandes (Milanov, 2009), este nivel de seguridad y por su naturaleza asímetrica lo hace util para la creación de firma digitales en documentos lo que nos permite asegurar la itegridad de un documento compartido por internet, basandonos en lo explicado por Mansour (2017) podemos usar el siguiente algoritmo para aplicarlo en firma digital:

1.Creación de una firma digital:

1. El remitente de un mensaje o documento digital tiene un par de claves RSA: una clave privada para firmar y una clave pública para verificar.
2. El remitente calcula el hash del mensaje o documento utilizando una función de hash segura, como SHA-256. El hash es una cadena de bits única y fija que representa el contenido del mensaje.
3. Luego, el remitente firma el hash utilizando su clave privada RSA. Esto se hace calculando la exponenciación modular del hash con la clave privada, utilizando el algoritmo RSA: $firma = hash^d n$, donde d es la clave privada y n es el módulo.
4. La firma digital resultante se adjunta al mensaje o documento original y se envía al destinatario.

2. Verificación de la firma digital:

1. El destinatario recibe el mensaje junto con la firma digital adjunta.
2. El destinatario calcula el hash del mensaje recibido utilizando la misma función de hash utilizada por el remitente.
3. Luego, el destinatario utiliza la clave pública RSA del remitente para verificar la firma digital. Esto se hace calculando la exponenciación modular de la firma con la clave pública: $hash\_verificado = firma^e n$, donde e es la clave pública y n es el módulo.
4. Si el hash verificado coincide con el hash del mensaje recibido, esto significa que la firma digital es válida y el mensaje no ha sido alterado en tránsito. Por lo tanto, se puede confiar en la autenticidad e integridad del mensaje.

Según Milanov (2009) ventajas de RSA como llave digital:

1. **Seguridad robusta:** RSA ofrece un alto nivel de seguridad debido a la complejidad del problema de factorización de números primos grandes, lo que lo hace resistente a los ataques criptoanalíticos.

2. **Aplicación versátil:** El algoritmo RSA se utiliza ampliamente en una variedad de aplicaciones, incluidas las comunicaciones seguras en Internet, la firma digital, la autenticación de usuarios y la protección de datos confidenciales.

3. **Estándar de la industria:** RSA ha sido ampliamente adoptado como un estándar de facto en la criptografía de clave pública, lo que garantiza su compatibilidad y interoperabilidad con una variedad de sistemas y protocolos de seguridad.

Segun Yance Sánchez (2022) desventajas de RSA como llave digital:

1. **Costo computacional:** La generación de claves RSA y las operaciones criptográficas asociadas, como el cifrado y el descifrado, pueden ser computacionalmente intensivas, especialmente para claves de mayor longitud y mensajes más largos.

2. **Tamaño de clave y almacenamiento:** Las claves RSA más seguras tienden a ser más largas en longitud, lo que puede aumentar el tamaño del archivo y la complejidad de almacenamiento de claves, especialmente en dispositivos con recursos limitados.

3. **Vulnerabilidad potencial a ataques cuánticos:** Aunque no es una amenaza inmediata, RSA es vulnerable a futuros avances en la computación cuántica, que podrían eventualmente resolver el problema de factorización de números grandes de manera eficiente, comprometiendo así la seguridad del algoritmo.

4. **Riesgo de pérdida de clave privada:** La pérdida o compromiso de la clave privada RSA puede resultar en la pérdida de confidencialidad y autenticidad de los datos cifrados, lo que resalta la importancia de una gestión segura de claves.

En resumen, RSA como llave digital ofrece una sólida seguridad y versatilidad en una variedad de aplicaciones, pero también presenta desafíos en términos de costo computacional y riesgos potenciales de seguridad. Es importante evaluar cuidadosamente sus ventajas y desventajas al seleccionar este algoritmo para implementaciones criptográficas.

Comparación de los métodos

Dadas las similitudes entre ellos, se decidió hacer una tabla de comparación entre los algoritmos de tipo simétrico y asimétrico

Para las firmas no proporciona información útil en la tabla de comparación, por lo que se hizo el siguiente listado:

Algoritmo	Velocidad	Costo	Tipo	Tamaño de la Llave (bits)	Seguridad
AES	Rápido	Bajo	Simétrico	128, 192, 256	Alto
TwoFish	Rápido	Bajo	Simétrico	128-256	Alto
RSA	Lento	Alto	Asimétrico	1024-8192	Alto
ECC	Moderado	Moderado	Asimétrico	160-512	Alto

Cuadro 1

Descripción de algoritmos criptográficos (Poggi, 2023)

- **Algoritmo:** RSA se basa en números primos y utiliza un par de claves para tanto la encriptación como la firma. DSS utiliza el algoritmo DSA con pares de claves distintas para firmar y verificar.
- **Seguridad:** RSA es más robusto pero potencialmente vulnerable a ataques cuánticos. DSS es más seguro, pero los ataques cuánticos lo afectan más.
- **Tamaño de la clave:** RSA requiere claves más grandes para una seguridad equivalente. DSS requiere claves más pequeñas debido a la eficiencia de su algoritmo.
- **Uso:** RSA se utiliza tanto para encriptación como para firmas en diversas aplicaciones. DSS se utiliza principalmente para firmas digitales, especialmente en industrias reguladas.Adobe (s.f.)

Marco Legislativo de la Criptografía en México

Como el lector podrá reconocer, los sistemas de leyes se enfrentan a enormes y continuos retos en lo que al cifrado respecta. Esto mayormente debido a la rápida difusión y evolución de dicho campo. Todavía cabe señalar, otro desafío crucial que enfrentan los gobiernos son los estragos para definir la frontera entre la protección de la intimidad y las posibles amenazas que dicha protección supone para las fuerzas del orden (Mora, 1996).

Ahora bien, dentro de las leyes en nuestro país relacionadas con la ciberseguridad, hemos notado que la historia en esta materia es reciente. Uno de los primeros documentos en los que fue mencionado algo relacionado a la protección de datos digitales fue en el Plan de Desarrollo Nacional de 1995, con el que el presidente Zedillo anunciaba que el gobierno promovería la formación de especialistas en todos los niveles, y que la actividad del gobierno federal se centrará en generar, difundir y aplicar todo tipo de innovaciones tecnológicas (Mora, 1996).

Más adelante, el Acta Federal de Comunicaciones hizo mención de que se castigará la ïnterceptación de información transmitida a través de redes públicas de telecomunicaciones", haciendo referencia a los hackers. Estas y otras regulaciones cobrarían más importancia en el 2011, después de que se presentara la mayor disputa de ciberseguridad en la hisotria de México: Los Zetas VS Anonymous. En este conflicto, el grupo de narcotraficantes consiguió secuestrar a un miembro de la agrupación Anonymous, y esta contraatacó amenzanado con revelar información de vínculos ilegales entre destacados cargos electos,

funcionarios públicos y miembros de las élites sociales con el cártel. En respuesta, Los Zetas amenzaron con iniciar una carga indiscriminada en contra de inocentes si era revelado cualquier tipo de información en Internet. Dicha crisis llegó a resolverse de forma relativamente pacífica, sin embargo tuvo un impacto gigante, ya que se convirtió en un evento que propulsó reformas y la creación de leyes de seguridad digital en México, las cuales eran prácticamente nulas antes de este año (Rodriguez-Hernandez y Velásquez, 2021).

De acuerdo a Rodriguez-Hernandez y Velásquez (2021), en el presente existe un amplio consenso por varios expertos que México tiene legislaciones competentes y actualizadas en relación a seguridad en telecomunicaciones y comercio electrónico. De entrada, ya se definió formalmente a la Ciberseguridad en la Estrategia Nacional de Seguridad del sexenio de Enrique Peña Nieto: çonjunto de, políticas, controles, procedimientos, métodos de gestión de riesgos y normas asociadas a la protección de la sociedad, el gobierno, la economía y la seguridad nacional en el ciberespacio y las redes públicas de de telecomunicaciones".

Posteriormente, en el 2014, la Ley Federal de Telecomunicaciones y Transmisiones unificó los diferentes códigos existentes de seguridad digital y de protección de datos. Esta ley en conjunto con los Códigos Civiles Federales Penales, establecieron un marco legislativo en el que se garantiza, en teoría al menos, la integridad de la privacidad. Cabe mencionarse que en el sexenio actual, el Presidente Andrés M. López no ha ofrecido una alternativa o cambio a las políticas de ciberseguridad de su predecesor ni ha renegado de ellas (Rodriguez-Hernandez y Velásquez, 2021).

Pasando finalmente a las regulaciones internacionales, la más reciente encuesta de Políticas de Criptografía realizada por el United States National Institute of Standards and Technology ("NIST") recalca que la Subsecretaría de Comercio Exterior, institución reguladora de las importaciones y exportaciones en México, no ha establecido prohibiciones ni controles en la tecnología de encriptación. Esto es una ventaja para el proyecto porque no encontraremos obstáculos para utilizar tecnologías que en otro países pueden estar bloqueadas. También recalcamos que, nuestro país se ha mostrado positivo ante el Grupo de Trabajo sobre Gobernanza de Internet de la ICANN de 2004 que incluyó a representantes estatales, de la industria, académicos y de la sociedad civil. Además, México también ha sido un miembro activo del Grupo de Expertos Gubernamentales de las Naciones Unidas y del Grupo de Trabajo de Ciberseguridad. Por otro lado, el banco central mexicano es miembro fundador del Centro de Política Monetaria Latinoamericana (CEMLA), y es miembro activo de sus iniciativas Fintech, como el foro Fintech Regulatory. De hecho, en el 2019, el CEMLA elogió la nueva Ley Fintech de México por estar centrada en çanales y actividadesz no solo en normatividad para el proveedor. Por su parte, La OEA y su Comité Interamericano contra el Terrorismo (CICTE) en particular, han sido socios estratégicos para el desarrollo

de varios países de América Latina y el Caribe en materia de ciberseguridad y políticas de resiliencia. Cabe señalarse que la Estrategia Nacional de Ciberseguridad de México reconoce la contribución del CICTE. Por último, el cumplimiento de dos acuerdos comerciales internacionales estratégicos han empujado a México a armonizar, o al menos comprometerse a armonizar, sus regulaciones y políticas en materia de seguridad informática, Fintech y derechos de propiedad intelectual: El Acuerdo Transpacífico y el acuerdo norteamericano USMCA (sucesor del TLCAN). Dichos acuerdos fueron rescatados en la Estrategia de Ciberseguridad de 2017, pero destaca el transpacífico porque México tiene un papel protagónico (Rodriguez-Hernandez y Velásquez, 2021).

En general, todo la anterior nos permitirá desarrollar un proyecto flexible y con variedad tecnológica, ya que al haber tantos compromisos en materia de seguridad, así como prácticamente nulas restricciones en la compra de soluciones de criptografía, nos es posible contemplar más formas de ayudar a nuestro socio formador. Igualmente, estas leyes nos recuerdan la importancia que tiene la protección de los datos, en un mundo cada vez más interconectado y digitalizado. Además, al revisar la legislación y la historia de México en ciberseguridad, podemos comprender mejor los desafíos y riesgos específicos que enfrentamos como país, lo que nos permite diseñar estrategias y medidas de protección más efectivas y adaptadas a nuestras necesidades y circunstancias particulares. En general, nos recuerda que los datos son un tema serio y que exige de atención, así como que debemos mantener en pie la identidad positiva que se tiene de la sociedad digital mexicana.

Recursos disponibles y Listado de Requerimientos

De las interacciones con la gente encargada de la institución de Casa Monarca se concluye que su infraestructura informática es muy sencilla; y a su vez esto se puede extender a la realidad de muchas otras casas de migrantes que llevan su labor con herramientas deficientes o inadecuadas. El equipo considera esencial el uso de tecnologías gratuitas u open-source para el desarrollo del proyecto; esto para evitar incluir en el presupuesto del albergue gastos de manejo de nube o software privado, dado que esto es de las principales cosas que se quiere reducir de acuerdo con el socio.

De forma inicial, se planea desarrollar un formulario que permita ingresar texto, que y contendrá todas las preguntas que se le hacen a un beneficiario al registrarse en la casa hogar. Dicho formulario estará conectado a una base de datos SQL manejada con *SQLite*. Dicho administrador nos resulta particularmente útil al no necesitar de un servidor; solo necesitando un disco duro para leer y escribir (SQLite, 2023). Igualmente, al ser un proyecto open-source, ofrece muchas extensiones desarrolladas por su comunidad, entre ellas, algunas dedicadas al cifrado. Relacionado a esto, sugerimos almacenar la base de datos en un disco duro externo principal, contemplamos tener uno adicional donde hacer copias periódicas

de la base, esto reemplazará la necesidad de adquirir productos basados en la nube, y por lo tanto, minimizamos la complejidad del ”sistema".

Cabe mencionarse que la desventaja que trae usar discos duros externos, es la posible pérdida, destrucción parcial o total, e inclusive robo de los aparatos. Por tanto, se reafirma la necesidad de tener más discos que funcionen como respaldos. Igualmente, no dejamos de considerar las vulnerabilidades humanas en el sistema, esto es que alguien se haga pasar por un migrante o un trabajador de Casa Monarca con el fin de usurpar información.

Al final del día el objetivo primario del proyecto es mantener el producto sencillo y redituable para la organización, pero sin olvidarse de los requerimientos de seguridad mínimos necesarios. Es así que proponemos los siguientes recursos físicos, nosotros consideramos que garantizarán el desarrollo de una solución a la medida de las necesidades del socio:

Recursos Físicos:

1. **Computadora**. Con 2GB de RAM (4 u 8GB serían deseables), procesador Intel Pentium4 o similar y sistema operativo Windows. Este equipo puede ser una de las computadoras que ya tiene Casa Monarca, aunque de ser necesario adquirir una, el costo es de $10,000.00 MXN máximo*. Sería lo más costoso en caso de no tenerse.
2. **3 unidades de Disco Duro Externo**. Con 500GB de memoria disponible. Buscaríamos que tenga puertos USB para poder usarse fácilmente. También buscaremos un diseño resistente para evitar posibles daños en caso de caídas u otros accidentes. El costo sería de máximo $850.00 MXN máximo por unidad*. En este caso, sería posible adquirir solo 2, pero lo ideal son 3. En ellos se guardará la información de forma local para no tener que adquirir servicios en la nube que pueden ser complejos y caros.

* Los precios anteriores fueron obtenidos con base al portal de RadioShack, una tienda especializada en tecnología y dispositivos electrónicos por lo que pueden variar de tienda a tienda, aunque se esperan precios similares.

Bibliotecas

Al momento, se determinó que el proyecto consistiese del uso de dos algoritmos de cifrado. Durante la captación de datos y previo al registro en la base SQL se ejecutará el primer algoritmo. Posteriormente, la base entera será cifrada.

Como se describió en la sección de algoritmos de llave simétrica, el estándar actual es el AES. Afortunadamente, el lenguaje de programación Python cuenta con la librería PyCryptoDome (s.f.), que

consiste de funciones preestablecidas de muchos algoritmos de cifrado, desde lo más sencillo a lo más complejo como AES o ECC. Se probará el algoritmo propuesto por Sherali (2024), una combinación de estos últimos dos estándares.

Se considera también la opción de una solución ”lightweight” para esta sección del cifrado; basada en lo propuesto por Dunmore, Samandari, y Jang-Jaccard (2023) en su algoritmo Cifrado por paseos en matrices (MEW, por sus siglas en inglés). Se presenta una solución eficiente y compacta, pensada para dispositivos IoT y con menor capacidad de procedimiento es una alternativa que será explorada dando prioridad al rendimiento en esta primer sección de cifrado.

A lo que respecta al cifrado de la base completa se propone el uso de la extensión SQLCipher de SQLite; similarmente, esta extensión cuenta con varios esquemas de cifrado que se probarán. Además, se tomarán en cuenta las propuestas de Wang y cols. (2020) sobre autenticación.

Bibliografía

Adobe. (s.f.). *What is a Digital Signature Standard (DSS)? | Adobe Acrobat.* Descargado de https://www.adobe.com/acrobat/business/hub/digital-signature-standards.html

Amazon AWS. (s.f.). *¿Qué es la criptografía? - Explicación sobre la criptografía - AWS.* Descargado de https://aws.amazon.com/es/what-is/cryptography/

BBVA. (2024, Febrero). De Esparta a la criptografía cuántica: breve historia de la encriptación de datos y sus tipos. Descargado de https://www.bbva.com/es/innovacion/de-esparta-a-la-criptografia-cuantica-breve-historia-de-la-encriptacion-de-datos-y-sus-tipos/

Computerphile. (2014, Jul). *Public key cryptography - computerphile.* Descargado de https://www.youtube.com/watch?v=GSIDS_lvRv4

Computerphile. (2018, Jan). *Elliptic curves - computerphile.* Descargado de https://www.youtube.com/watch?v=NF1pwjL9-DE

Dunmore, A., Samandari, J., y Jang-Jaccard, J. (2023). Matrix encryption walks for lightweight cryptography. *Cryptography*, *7*(3), 41. Descargado de https://0-doi-org.biblioteca-ils.tec.mx/10.3390/cryptography7030041

El Sol de México. (2021). También llegan a méxico migrantes sudamericanos en busca del "sueño americano". *El Sol de México (Mexico City, Mexico).*

GfG. (2023a). *Aes full form.* Descargado de https://www.geeksforgeeks.org/aes-full-form/?ref=lbp

GfG. (2023b). *Digital signature standard (dss).* Descargado de https://www.geeksforgeeks.org/digital-signature-standard-dss/

Grigutytė, M. (2023, Noviembre). *Block cipher vs stream cipher: Understanding the difference.* Descargado de https://nordvpn.com/es-mx/blog/block-cipher-vs-stream-cipher/

Guseva, A. (2023, Marzo). *History of cryptography: From antiquity to quantum future | CoinLoan Blog.* Descargado de https://coinloan.io/blog/history-of-cryptography/

IBM. (2021). *Digital signature overview.* Descargado de https://www.ibm.com/docs/en/b2badv-communication/1.0.0?topic=overview-digital-signature

IBM. (2022). *Criptografía de clave pública.* Descargado de https://www.ibm.com/docs/es/integration-bus/10.1?topic=overview-public-key-cryptography

Islas-Mendoza, E., Jiménez-Vázquez, C., Silva-García, V., y Flores-Carapia, R. (2013). Diffie-hellman protocol based on elgamal and aes cryptosystems. *IOSR J. Eng.*, *3*(7), 30–33.

Johnson, D., Menezes, A., y Vanstone, S. (2001). The elliptic curve digital signature algorithm (ecdsa).

International journal of information security, *1*, 36–63.

Jurišic, A., y Menezes, A. (1997). Elliptic curves and cryptography. *Dr. Dobb's Journal*, 26–36.

Kaspersky. (2024). *Symmetric encryption.* Descargado de https://encyclopedia.kaspersky.com/glossary/symmetric-encryption/

Koblitz, N., Menezes, A., y Vanstone, S. (2000). The state of elliptic curve cryptography. *Designs, codes and cryptography*, *19*, 173–193.

Mansour, A. H. (2017). Analysis of rsa digital signature key generation using strong prime. *Int. J. Comput*, *24*(1), 28–36.

Maurer, U. M., y Wolf, S. (2000, Mar). The diffie–hellman protocol. *Designs, Codes and Cryptography*, *19*(2), 147–171. doi: 10.1023/A:1008302122286

Milanov, E. (2009). The rsa algorithm. *RSA laboratories*, 1–11.

Mora, M. I. (1996). Privacy law issues for encryption and government control in mexico.

Nagaraj, K. (2023, 3). TwoFish Encryption: A Comprehensive Guide | 2023 - Karthikeyan Nagaraj - Medium. Descargado de https://cyberw1ng.medium.com/twofish-encryption-a-comprehensive-guide-2023-b3ad0f844870

National Cryptologic Foundation. (2019, Febrero). *National Cryptologic Foundation.* Descargado de https://cryptologicfoundation.org/event/2019/02/06/1549429200/1802-sir-charles-wheatstone-cipher-inventor-born-

Oswald. (2022, 12). *What is the advanced encryption standard (aes)?* Descargado de https://www.usnews.com/360-reviews/privacy/what-is-advanced-encryption-standard

Panhwar, M. A., Ali, S., Panhwar, G., y Ali, K. (2019, Jan). *Saca: A study of symmetric and asymmetric cryptographic algorithms.* Descargado de https://www.researchgate.net/publication/330555888_SACA_A_Study_of_Symmetric_and_Asymmetric_Cryptographic_Algorithms

Poggi, N. (2023). *Types of encryption: Symmetric or asymmetric? rsa or aes?* Descargado de https://preyproject.com/blog/types-of-encryption-symmetric-or-asymmetric-rsa-or-aes

Poston, H. (2019, Julio). *The Story of Cryptography - 20th Century Cryptography.* Descargado de https://ghostvolt.com/blog/The-Story-of-Cryptography-Part-2-20th-Century-Cryptography.html

Pound, M. (2017a, Dec). *Diffie hellman -the mathematics bit- computerphile.* Descargado de https://www.youtube.com/watch?v=Yjrfm_oRO0w

Pound, M. (2017b, Dec). *Key exchange problems - computerphile.* Descargado de https://www.youtube.com/watch?v=Yjrfm_oRO0w&t=13s

Prado, V., y Gonzáles, S. G. (2023). Actitudes de los ciudadanos neoloneses en la integración de los migrantes centroamericanos. *Journal of the Academy*, *8*, 50–70. doi: 10.47058/joa8.4

PyCryptoDome. (s.f.). *Pycryptodome — pycryptodome 3.210b0 documentation.* Descargado Marzo 13, 2024, de https://pycryptodome.readthedocs.io/en/latest/src/introduction.html

Rodriguez-Hernandez, S. M., y Velásquez, N. (2021). Mexico and cybersecurity: Policies, challenges, and concerns. En *Routledge companion to global cyber-security strategy* (pp. 484–493). Routledge.

Ruiz, P. G. (2023, Jul). *Cifrado homomórfico: una introducción a la computación segura de los datos - iic.* Descargado de https://www.iic.uam.es/noticias/cifrado-homomorfico-introduccion-a-la-computacion-segura-datos/

Sadurní, J. M. (2024, Marzo). Hedy Lamarr, una prodigiosa inventora eclipsada por su fama en Hollywood. Descargado de https://historia.nationalgeographic.com.es/a/hedy-lamarr-prodigiosa-inventora-eclipsada-por-su-fama-hollywood_14882

Schneier, B. (s.f.). *Twofish.* Descargado de https://www.schneier.com/academic/twofish/

Sherali, F. (2024). A new approach for enhancing aes-based data encryption using ecc. *International Journal of Mathematics & Computer Science*, *19*(1), 229–235.

SQLite. (2023, Octubre 10). *About sqlite.* Descargado Marzo 13, 2024, de https://www.sqlite.org/about.html

Stohrer, C., y Lugrin, T. (2023, Jan). *Asymmetric encryption.* Descargado de https://link.springer.com/chapter/10.1007/978-3-031-33386-6_3

THALES. (2023, Febrero). *A brief history of encryption (and cryptography).* Descargado de https://www.thalesgroup.com/en/markets/digital-identity-and-security/magazine/brief-history-encryption

Universidad de Granada. (s.f.). *El cifrado de Vigenère.* Descargado de https://www.ugr.es/~anillos/textos/pdf/2011/EXPO-1.Criptografia/02a11

UPMRIP. (2024). *Boletín de estadísticas sobre delitos perpetrados en contra de personas migrantes irregulares en méxico 2023.* Descargado Marzo 7, 2024, de https://portales.segob.gob.mx/work/models/PoliticaMigratoria/CEM/Estadisticas/DelitosMigIrreg/2023/BMigIrregDelitos_2023.pdf

Uribe, C., y Hernández, J. (2024). Vulnerabilidad y riesgos psicosociales en trayectorias laborales de migrantes centroamericanos y mexicanos en su estación querétaro, méxico. *Transdigital*, *5*(9). doi: 10.56162/transdigital292

Wang, Y., Shen, Y., Su, C., Ma, J., Liu, L., y Dong, X. (2020). Cryptsqlite: Sqlite with high data security. *IEEE Transactions on Computers*, *69*(5), 666-678. doi: 10.1109/TC.2019.2963303

Wollinger, T., y Kumar, S. (2006, Jan). *Fundamentals of asymmetric cryptography.* Descargado de https://link.springer.com/chapter/10.1007/3-540-28428-1_9#citeas

Yance Sánchez, C. E. (2022). *Análisis comparativo de los métodos de encriptación aes y rsa, para las seguridades de los sistemas de información* (B.S. thesis). Babahoyo: UTB-FAFI. 2022.