Proyecto 03
Esquema de Secreto Compartido de Shamir

Universidad Nacional Autónoma de México

Facultad de Ciencias

Modelado y Programación

Integrantes:
Hannia Laura López Ceballos
Mónica Miranda Mijangos

25 de enero de 2022

1. Desarrollo

El esquema de secreto compartido de Shamir hace posible que un sólo dato pueda ser ocultado de manera que, a partir de él, se generan n diferentes datos y que con, al menos $t \leq n$ cualesquiera de ellos sea posible recuperar el dato original.

Es bien sabido que r puntos en el plano determinan un único polinomio de grado $r-1$, a saber, el polinomio que interpola los r puntos. Bastan, por ejemplo, dos puntos para determinar la línea que los une, o tres puntos para encontrar la única parábola que pasa por ellos. Así, en el esquema de Shamir de secreto compartido, el dato que se pretende ocultar y compartir, digamos K, se considera como el término independiente de un polinomio de grado $t-1$:

$$P(x) = c_{t-1}x^{t-1} + c_{t-2}x^{t-2} + ... + c_1x + K$$

Así, $P(x)$ puede ser reconstruido a partir de cualesquiera t puntos en los que haya sido evaluado. Para construir $P(x)$ basta generar $t-1$ coeficientes de manera aleatoria, los que, junto con el secreto K, constituyen todos los coeficientes de P. Se procede luego a evaluar a P en $n \geq t$ puntos diferentes, obteniéndose así los n puntos: $\{(x_1, P(x_1)), (x_2, P(x_2)), ...(x_n, P(x_n))\}$ que pueden ser distribuidos entre las n personas autorizadas para acceder al secreto K. Si se logran reunir, al menos t de los n puntos distribuidos, es posible calcular el polinomio interpolante de grado $t-1$ que pasa por los puntos reunidos, cuyo término independiente es K, el secreto que se deseaba compartir.

1.1. Definición del Problema

El propósito del proyecto final, es elaborar un programa que implemente el esquema de Shamir para compartir la clave necesaria para descifrar un archivo que suponemos contiene información confidencial. Dicha información debe poder ser accedida siempre que estén presentes, al menos u, de los n miembros de un grupo de personas con autorización para acceder a ella.

A partir de un documento u otro tipo de archivo que suponemos contiene información confidencial, se debe generar una versión cifrada de él. Para ello se utilizará un mecanismo de cifrado simétrico, por ejemplo **AES** (Advanced Encryption Standard). El documento claro (como llamaremos al original) será cifrado por la entidad que lo produce usando una contraseña sólo conocida por dicha entidad y que debe mantenerse en secreto aún cuando ya no será utilizada. A partir de esta contraseña se debe generar la clave de cifrado K, con la que será cifrado el documento claro. Además de producir el documento cifrado, la clave será considerada como el término independiente de un polinomio de grado $t-1$, mismo que será evaluado en n puntos diferentes. Tanto el documento cifrado, como las n evaluaciones del polinomio, pueden ser distribuidos entre las personas con acceso autorizado al documento claro.

Si luego se logran reunir al menos $t \leq n$ de estas personas, usando el programa, pueden descifrar el documento cifrado recuperando el claro.

1.2. Análisis del Problema

- Requsitos funcionales:

  - ¿Qué debe entregar como salida dado cierto tipo de entrada?

    El programa debe funcionar en dos modalidades, para cifrar (opción **c**) y para descifrar (opción **d**).

    Para **cifrar** debe recibir la opción c seguido de el nombre donde se guardará el archivo con las n evaluaciones, el número de avaluaciones requeridas ($n > 2$), el mínimo número t de evaluaciones para descifrar ($1 < t \leq n$) y el nombre del archivo con el texto claro. Además de una contraseña que será solicitada durante la ejecución del programa.
    Regresará un archivo con las evaluaciones con extensión $.frg$ y el archivo cifrado con el nombre original y una extensión $.aes$ más.

    Para **descifrar** debe recibir la opción d seguido de el nombre del archivo con al menos t evaluaciones del polinomio con extensión $.frg$ y el nombre del archivo con el texto cifrado con extensión $.aes$.
    Y regresará el archivo con el nombre original y el texto claro.

  - ¿Qué se debe llevar a cabo?

    A partir de la contraseña p tecleada por el usuario obtener su código hash con SHA-256, establecerlo como el término independiente del polinomio y obtener las n evaluaciones.
    Luego, usarlo como la llave de cifrado de AES para cifrar el documento claro y guardar tanto el documento cifrado como las evaluaciones del polinomio.

    Para descifrar solo debemos hacer el proceso inverso, calculando el último valor del polinomio, y usarlo como llave para devolver el texto encriptado al original.

- Requisitos no funcionales:

  - Eficiencia

    El sistema recibe cinco entradas diferentes, sin embargo no es necesario que sea capaz de procesar x transacciones por segundo. Por lo que la funcionalidad del sistema y transacción de negocio no requiere responder al usuario en un cierto límite de tiempo.

  - Tolerancia a fallas

    El programa lanza un mensaje al ususario cuando le dan una entrada incorrecta o un archivo que no contiene datos, es decir, el programa se detiene imprimiendo en la línea de comandos indicandole al usuario que el programa falló.

  - Amigabilidad

    La interfaz del programa debe ser amigable para que el usuario pueda meter la entrada junto con el archivo de texto a cifrar.

  - Seguridad

    Nuestro programa utliza el estándar de cifrado de datos AES que admite claves de cifrado de 256 bits, longitud recomendada para documentos clasificados de seguridad nacional, por lo que podemos asegurar que nuestros datos están bien protegidos.

  - Interoperabilidad

    Es la capacidad de los sistemas de información y de los procedimientos a los que éstos dan soporte, de compartir datos y posibilitar el intercambio de información

y conocimiento entre ellos, en este caso no hay intercambio de información de un sistema a otro.

1.3. Selección de la mejor alternativa

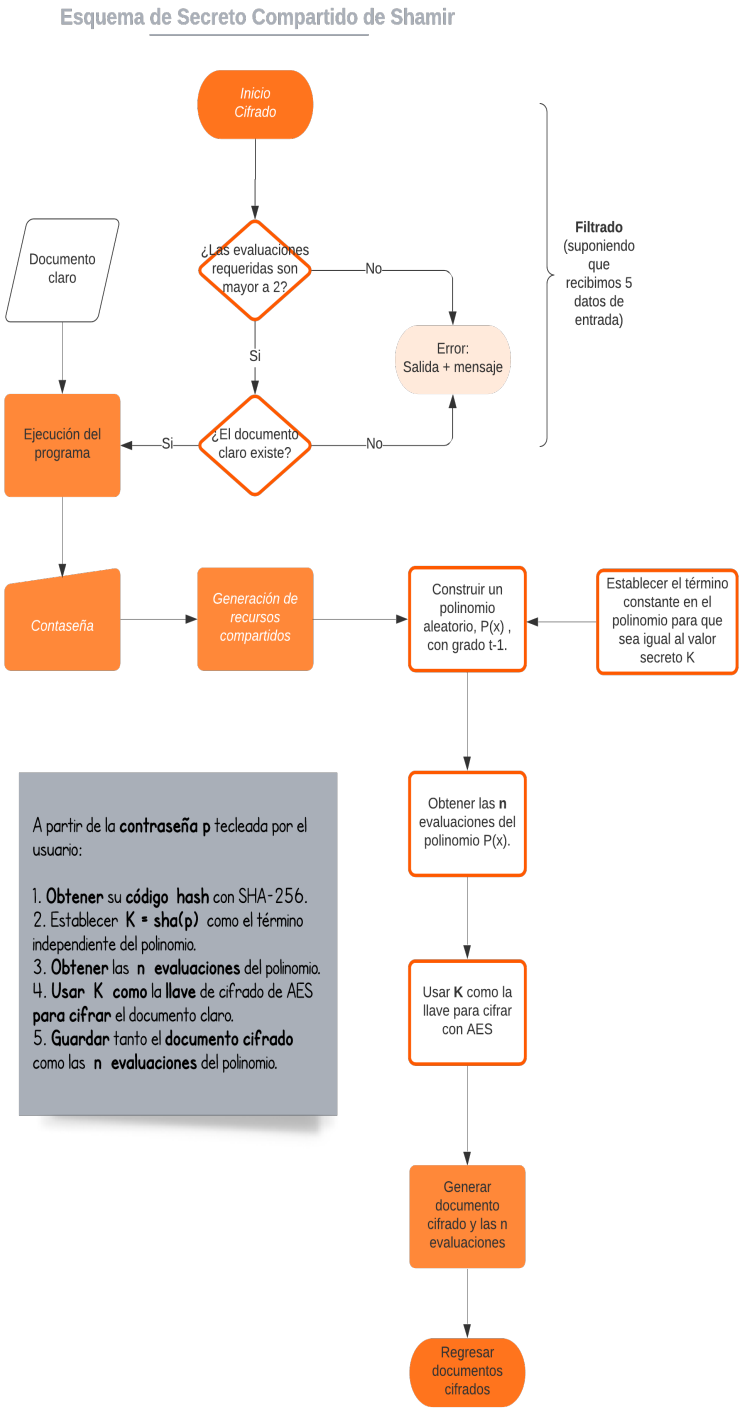
- Paradigma de programación: Orientado a Objetos.
- Lenguaje: Python.
  Ya que es un lenguaje multiparadigma enfocado en la legibilidad del código, además de ser interpretado, es decir, puede compilar el código fuente original en una forma intermedia más compacta, y después traducir eso al código de máquina; dinámico, las variables pueden tomar valores de distinto tipo, y multiplataforma.
- Herramientas para pre o post procesar datos:
  - Utilizamos **getpass** para solicitar al usuario una contraseña sin hacer eco.
  - Utilizamos **hashlib** para implementar SHA-256.
  - Utilizamos **Cryptodome.AES** para cifrar los archivos solicitados.
- Frameworks o bibliotecas:
  - ¿Qué tan útiles son? Son muy útiles ya que son amplias y cuentan con gran cantidad de producciones en contenidos, constan de diversos módulos que permiten el acceso de funcionalidades específicas de modo que podemos concentrarnos en lo que realmente nos ocupa.
  - ¿Qué se necesita para usarlas? Únicamente se necesita importarlas con la palabra import seguido del nombre de la biblioteca a utilizar.
  - ¿Están bien documentadas? Si, siguen un estándar oficial de codificación, el cual es PEP-8.
  - ¿Han sido probadas? Si, son programadas y probadas por terceros.
  - ¿Vale la pena usarlas? Si, ya que nosotros como programadores nos queremos evitar la necesidad de repetir en nuestros programas una y otra vez el mismo código, lo que también hace a nuestro programa poco eficiente.
  - ¿Son seguras? Con python podemos cifrar la información, se puede cifrar desde archivos hasta carpetas, con esto se puede mantener una integridad en la información. Existen muchas bibliotecas que nos permiten elegir el algoritmo de cifrado que deseemos utilizar.

1.4. Diagrama de flujo

CIFRAR

Se debe proporcionar, en la línea de llamada:

1. La **opción** `c`.
2. El **nombre del archivo** en el que serán **guardadas** las `n` **evaluaciones** del polinomio.
3. El **número total de evaluaciones** requeridas ($n > 2$).
4. El **número mínimo de puntos** necesarios **para descifrar** ($1 < t \leq n$).
5. El nombre del **archivo** con el documento **claro**.
6. Ya con el programa **en ejecución** se debe **solicitar** al usuario una **contraseña** (sin hacer eco en la terminal).



A partir de la **contraseña** `p` tecleada por el usuario:

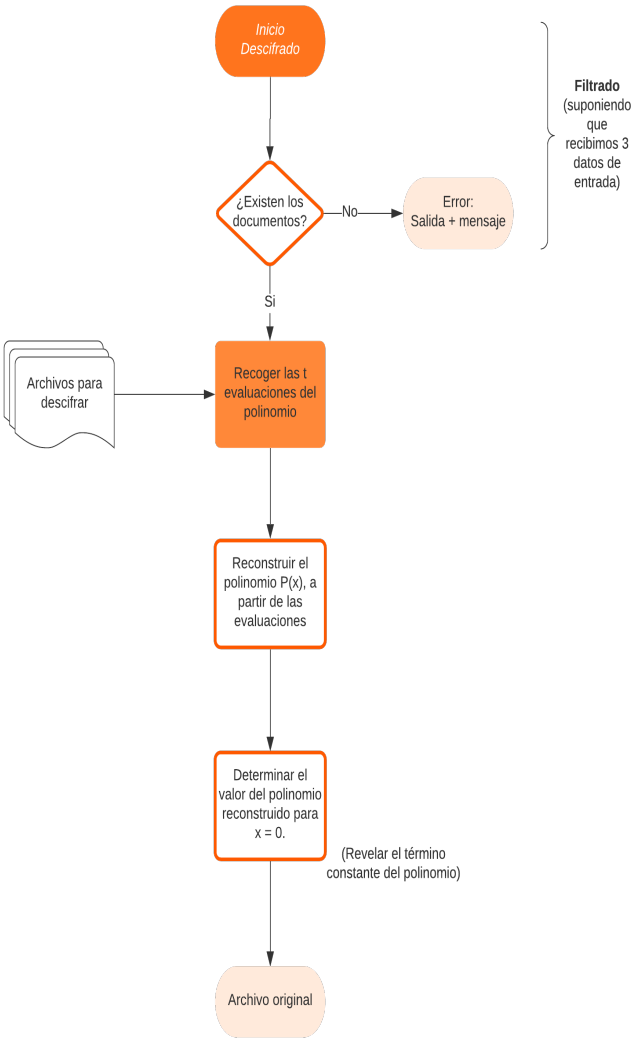
1. **Obtener** su **código hash** con SHA-256.
2. Establecer `K = sha(p)` como el término independiente del polinomio.
3. **Obtener** las `n` **evaluaciones** del polinomio.
4. **Usar** `K` **como** la **llave** de cifrado de AES **para cifrar** el documento claro.
5. **Guardar** tanto el **documento cifrado** como las `n` **evaluaciones** del polinomio.

DESCIFRAR

Se debe proporcionar, en la línea de llamada:

1. La **opción** `d` .
2. El **nombre del archivo** con, al menos, `t` de las `n` **evaluaciones** del polinomio.
3. El **nombre** del **archivo cifrado**.

Esquema de Secreto Compartido de Shamir



1.5. Mantenimiento

- ¿Cuánto cobrarías por el proyecto y futuro mantenimiento? Para poder poner un precio al proyecto podemos desglosar una lista de tareas a traves de las cuales se estimara el tiempo que se le dedicara a cada una en horas, hay que tener en cuenta que a cada una de las tareas se le deben agregar los costos que se requerirán para que el sistema funcione, como lo sería:

  - Equipo específico para el proyecto.
  - Servidores en la nube.
  - Dominios
  - Licencias
  - Bases de datos

- Una buena forma de determinar el precio por hora del proyecto es tomando en cuenta la experiencia que se tiene y en todo caso lo que se desea ganar mensualmente. Tomando esto en cuenta, de acuerdo a nuestra experiencia y horas trabajadas, el precio por hora sería de 197.75 MXN lo que nos da un total de 31,600 MXN.

- Mantenimiento: El tipo de mantenimiento que se le puede dar a este proyecto es adaptativo, el cual consiste en la modificación del programa debido a cambios en el entorno en el que se ejecuta, tambien se podría contar con un manejo perfectivo o preventivo en los cuales se mejoran o añaden nuevas funcionalidades, así como la mejora de la eficiencia y tiempo de ejecución del programa, la estructuración de los programas para aumentar su legibilidad o incluir nuevos comentarios.