Profesor: Alejandro Hevia
Auxiliar: Bryan Ortiz
Ayudantes: Vanessa Gaete Valeria León

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Proyectos Grupales
Propuestas

1. Código Enigma

Durante la Segunda Guerra Mundial los alemanes utilizaron la famosa Máquina Enigma, una máquina de encriptación basada principalmente en sustitución que gracias a la gran cantidad de combinaciones posibles que podía utilizar y que cambiaba cada día se consideraba poderosa. Y en efecto, demoró años de arduo trabajo para poder ser quebrada. Esta máquina se utilizaba por los militares para mandar mensajes importantes tales como avisos de ataques, buscando que no pudieran ser interceptados por los Aliados. La configuración de la máquina cambiaba cada día, los alemanes tenían un libro de códigos con la configuración diaria que iban siguiendo de forma coordinada para encriptar y desencriptar el mensaje correctamente. Esto añadía una gran complejidad a los Aliados para poder romper el sistema.

Este proyecto consiste en implementar una máquina Enigma. Para ello usted debe investigar sobre el funcionamiento de esta máquina, sus componentes, y el proceso de encriptación, para luego hacer una implementación coherente en el lenguaje que usted desee. La máquina debe funcionar a través de la ventanas de comandos, usted decide cómo diseñar el programa interactivo. Sin embargo la máquina debe cumplir con los siguientes requisitos:

- Debe programar su configuración a través de un mensaje del estilo "III II I V 02 21 12 10 AU IR LM ZB QE HF VD UKB-c". El cual significa que se utilizan los rotores III, II, I y V en dicho orden, las siguientes 4 instrucciones indican la posición en la cual se configura inicialmente cada rotor respectivamente, los siguientes pares de letras indican los intercambios del plugboard, y por último el texto final corresponde al reflector utilizado.
- Considere que pueden haber de 0 a 13 pares de letras para intercambio en el plugboard.
- Su máquina debe tener plugboard, 4 espacios para rotores y reflector.
- Los rotores disponibles para la utilización de su máquina deben ser: I, II, III, V, VI, VII, VIII y Beta. Se debe incluir la configuración original de las muescas para cada rotor. Sin embargo, para la muesca VIII le pedimos una configuración especial de muescas en la posiciones "V" y "A". Implemente correctamente la rotación de los rotores, incluyendo también la rotación en cascada cuando sea necesario.
- Los reflectores disponibles deben ser UKB-c y UKB-b.
- Se debe poder encriptar mensajes con su implementación.

Esta página le puede ser útil para ver la configuración de los rotores y de los reflectores solicitados: Rotores y cableado de enigma

Tenga cuidado con copiar códigos desde Internet, puesto que son fácilmente identificables por el equipo docente. Además, tal como ya se mencionó, hay distintas variedades de máquinas Enigma y en este proyecto se les pide una máquina particular. Las soluciones de Internet no satisfacen todo lo expuesto anteriormente y por lo general son más básicas puesto que ignoran alguno de los detalles de funcionamiento. No se pueden utilizar librerías que contengan implementaciones para algunas de las partes de Enigma, ni tampoco una librería que contenga el funcionamiento completo de dicha máquina. Si bien GPT sirve para entender un poco mejor el funcionamiento de Enigma, tenga cuidado con su uso puesto que con respecto a la Máquina Enigma frecuentemente da respuestas incorrectas que no son fáciles de notar a menos que usted comprenda el funcionamiento.

2. Aplicación de Firma Electrónica

En muchos trámites legales, es necesario entregar documentos firmados, ya sea por uno mismo o por un externo que dé su aprobación al documento. Si bien lo usual es utilizar una firma física, se ha vuelto común aceptar firmas digitalizadas. Aplicaciones domo DocuSign o Adobe Acrobat permiten realizar esto de manera rápida y sencilla.

El objetivo de este proyecto consiste en crear una aplicación que permita a sus usuarios firmar documentos de forma digital. De forma concreta, se espera que su aplicación permita lo siguiente:

1. **Registro**: Un usuario debe poder registrarse ingresando su nombre, email y contraseña. A cada usuario registrado le debe asociar automáticamente una clave pública y una privada. La pública puede quedar registrada dentro de la aplicación, mientras que la privada debe ser entregada al usuario (**no debe** quedar guardada en el servidor).
2. **Firmar Documentos**: Cualquier usuario registrado debe ser capaz de crear una entrada de texto y firmarla a su nombre. Note que para esto deberá utilizar su clave privada.
3. **Solicitar Firma**: Un usuario registrado U_1 puede solicitar a otro usuario U_2 que firme un texto definido por U_1. Para ello, cuando cualquier usuario U inicie sesión, la página de inicio debe mostrar dos listados: textos con firma pendiente y textos que ya han sido firmados. El primer listado presenta los textos que U debe firmar, permitiendo descargar su contenido en formato `.txt` y mostrando una opción para firmar. El segundo listado presenta documentos ya firmados, ya sea por U o peticiones de firma hacia otros usuarios por parte de U que ya fueron resueltas.
4. **Verificación:** Debe existir una vista donde cualquier usuario (ya sea registrado o no), pueda subir un documento de texto plano, la firma y el nombre del autor de la firma. La aplicación deberá indicar si la firma es válida o no.

Para la realización de este proyecto pueden utilizar cualquier librería o *framework* para aplicaciones web. Se evaluará la funcionalidad y su correctitud teórica por sobre la interfaz de la aplicación.

Adicionalmente, deberán responder las siguientes preguntas:

- ¿Por qué cree que no permitimos al servidor guardar las claves privadas? ¿Se les ocurre una solución diferente?
- Estudie la posibilidad de firmar documentos PDF en lugar de texto plano. ¿Qué se necesita? ¿Qué dificultades existen?
- ¿Qué vulnerabilidades posee esta aplicación? ¿Cómo las solucionaría?

3. Correo Autenticado

Durante el curso vimos la importancia de mantener canales de comunicación autenticados y confidenciales. Dentro de este proyecto pondremos en práctica dicho conocimiento implementando una plataforma simplificada de correo electrónico *CryptoMail*.

Un usuario deberá registrarse en *CryptoMail* para poder interactuar con otros usuarios. Cada usuario registrado contará con un buzón de mensajes recibidos, y la forma de comunicarse será insertando mensajes encriptados (mediante un sistema de encriptación autenticado y simétrico) en el buzón del usuario destinatario. *Por ejemplo, si Alice busca enviarle un mail a Bob, simplemente guardará su mensaje encriptado en el buzón de Bob.*

1. Implemente el sistema descrito, pensando primero en sólo dos usuarios. En particular, si un usuario A decide enviarle un mensaje a B se le debe solicitar el mensaje y la clave de encriptación. Al mismo tiempo, si B desea leer de su buzón el mensaje enviado por A, se le solicitará que ingrese la misma clave para poder leer el mensaje.
2. Implemente además una etapa previa a la comunicación para efectuar un acuerdo de claves. Es decir, si A desea escribirle a B, primero deberán acordar la clave.

3. Extienda la aplicación para admitir más de dos usuarios (la comunicación seguirá siendo siempre entre dos usuarios). Note que A necesitará $K_{(a,b)}$ para hablar con B, y una clave $K_{(a,c)}$ para hablar con C.

Para la realización de este proyecto pueden utilizar cualquier librería o *framework* para aplicaciones web. Se evaluará la funcionalidad y su correctitud teórica por sobre la interfaz de la aplicación.

Importante: notar que las claves siempre son solicitadas al usuario, jamás quedan almacenadas en el sistema.

Respecto a su proyecto, responda además las siguientes preguntas:

- ¿Existe algún problema de seguridad con su solución?
- ¿Qué mejoras se podrían implementar?
- Describa el esquema de encriptación autenticada implementado y justifique por qué lo escogió.

4. Amigo Secreto

En épocas navideñas, un juego popular es el "amigo secreto" (o *secret santa*). El juego consiste en que cada participante debe recibir un regalo de parte de otro participante anónimo. Una forma ~~aburrida~~ sencilla de hacer esto es colocando todos los nombres en una bolsita, de modo que cada uno saca sólo un nombre y lo mantiene secreto. También existen también aplicaciones como SorteoAmigoSecreto que automatizan la gestión. Permitiendo que cada participante además de *hints* sobre qué le gustaría recibir.

Nuestra solución es ligeramente más sofisticada, y lo mejor, ¡utiliza criptografía!. En particular, vamos a permitir que cada participante pueda tener una conversación privada anónima con su amigo secreto, permitiendo que puedan resolver dudas sobre el regalo (porque siempre surgen).

Se les pide concretamente que implementen una aplicación que facilite el siguiente proceso:

1. Cada participante deberá contar con un par de claves (pk_a, sk_a) que deben ser anónimas.
2. El sistema recolecta las pk_a (de forma anónima) y las revuelve en una lista.
3. Dada la lista aleatorizada de claves públicas anónimas, cada participante es capaz de reconocer su propia clave pero no la del resto. Luego, cada participante selecciona la clave que se encuentra justo antes que la suya (esto evita que seas tu propio amigo secreto).
4. Una vez que cada participante A_i tiene la clave pública anónima pk_a de su amigo secreto, genera una nuevo par de claves pública y privada (pk^i, sk^i) y computa el valor:

$$c_i := \mathsf{Enc}_{pk_a}(A_i \| pk^i)$$

   Es decir, genera un texto cifrado que sólo podrá ser desencriptado por su amigo secreto, en el cual le indica su identidad y una clave pública para que le envíen mensajes en caso de cualquier duda.
5. Se publican todos los c_i, y cada participante busca el texto cifrado que desencripte mediante su sk_a. Con esto, podrá saber a quién debe hacerle un regalo.

Para la realización de este proyecto pueden utilizar cualquier librería o *framework*, siempre y cuando se realice lo pedido.

Se le solicita además que responda las siguientes preguntas:

- ¿Qué debe satisfacer el sistema de encriptación para que el algoritmo propuesto funcione adecuadamente? Explique.
- ¿Qué mejoras sustanciales se les ocurren para este algoritmo?
- Realice una breve investigación sobre sistemas criptográficos de comunicación anónima.
- ¿Cree que sus amigos utilizarían la aplicación desarrollada para realizar un amigo secreto a fin de año?

5. Ataque a EMSA-PKCSv1.5

Un esquema de firmas digitales utiliza una clave pública para firmar mensajes. Esto es, Alice con su clave privada sk puede firmar un mensaje m, de manera que cualquiera que conozca la clave pública de Alice pk podrá verificar la validez de la firma.

EMSA-PKCSv1.5 es una manera específica de usar firmas de mensajes basada en RSA, según el estándar *PKCS*[1].

- Investiguen el modo de operación del esquema de firmas EMSA-PKCSv1.5. En particular, indiquen los algoritmos de firma y de verificación.
- El profesor les solicita asesoría respecto a este módulo de firmas, ya que desea firmar el acta de notas. Asuma que el sistema registra cualquier acta que se encuentre válidamente firmada.
  Si m representa el acta de notas, y σ representa la firma utilizando el esquema estudiado bajo la clave privada (secreta) sk del profesor. Demuestren que si alguien conoce pk, m y σ, entonces puede proponer una nueva clave pública pk' y un nuevo mensaje m' tal que la firma σ también verifique.
- Plantee formalmente el punto anterior como un juego, y presente un adversario que mediante una única consulta a su oráculo pueda ganar el juego.
- Estudien otros posibles ataques al esquema de firmas.
- ¿Recomendarían usar EMSA-PKCSv1.5?

6. Proof of Work Lottery

En este proyecto, se les pide crear una aplicación que permita implementar una lotería para múltiples participantes. Suponga que el sistema cuenta con un foro público donde todos/as pueden publicar mensajes de texto en forma autentificada.

La lotería, de nombre POWLOTO, está parametrizada por un nivel de dificultad $k \in \mathbb{N}$, y procede en rondas de la siguiente manera:

- Inicialmente, en la ronda $i = 0$, alguien publica el hash de un mensaje de contenido impredecible, con el formato $V_0 = H_1(\texttt{"VALORINICIALIMPREDECIBLE"})$, donde H_1 es un SHA256 con 256 bits de salida.
- En la ronda i-ésima, cada participante con identificador ID debe resolver el puzzle de la siguiente manera: debe encontrar un valor R_i tal que el valor Z_i calculado como sigue

  $$Z_i = H_2(V_i||ID||R_i)$$

  tiene los primeros k bits menos significativos en 0. La función H_2 es SHA3 con 256 bits de salida. El primero en resolver este puzzle debe publicar los valores (ID, R_i) en el foro.
- Si alguien publica valores (ID', R'_i) en el foro, todos los participantes deben verificar si los valores publicados satisfacen el puzzle. Si son incorrectos, son ignorados por todos/as. Si los valores publicados son correctos, todos/as definen $V_{i+1} = H_1(ID'||R'_i)$, se incrementa i, y se inicia la siguiente ronda.

Implemente una lotería de esta forma, explorando posibles valores para k de manera que exista un ganador (a) cada minuto, (b) cada 10 minutos, y (c) cada hora. El puzzle mencionado en este problema es un tipo de puzzle denominado PROOF OF WORK y muy utilizado en criptomonedas como Bitcoin.

- Investigue sobre proof of work en criptomonedas y explique qué supuestos debe hacer para que su implementación de lotería sea *justa*.
- ¿Por qué es importante que V_0 sea impredecible para todos?
- Investigue sobre fuentes de aleatoriedad confiables para instanciar el primer valor V_0.

[1] *PKCS:* Public Key Cryptography Standards

7. Learning With Errors (LWE)

Investigue sobre el tema "Learning With Errors" (LWE) y sus variantes. En particular,

- Explique el problema conceptualmente y con ejemplos concretos.
- Explique por qué se considera difícil. Es menos, igual o más difícil que problemas como el logaritmo discreto? Discuta.
- Explique variantes de problema.
- Investigue su uso en el diseño de esquemas de encritpación y de otros esquemas criptográficos. En particular, muestre en detalle un esquema de encriptación basado en una variante del problema LWE.
- Muestre/explore implementaciones y su eficiencia.
- ¿Qué quiere decir que los algoritmos basados en LWE sean *post cuánticos*?
- Investiguen, sean creatives.

8. Blockchain y Criptomonedas

En este problema, se les pide investigar sobre Blockchain y criptomonedas, en particular sobre tres esquemas distintos: Ethereum, Chia y Algorand.

- Explique cómo funciona cada criptomoneda, conceptualmente y con ejemplos concretos.
- Explique por qué funciona cada moneda, el rol de los mineros y observadores. En qué se basa la seguridad de estas monedas? Cuál es la dificultad computacional asociada? Discuta.
- Explique variantes de dichas monedas.
- ¿En qué consisten los contratos inteligentes?
- ¿En que consisten las side-chains?

9. Proyecto: Lectura de Artículos

Escoja alguno de los siguientes artículos e investigue respecto al tópico elegido. Debe presentarlo y explicarlo de manera de ser comprensible al resto de la clase.

1. Elliptic Curve Cryptography: Pre And Post Quantum
2. Honey Encryption: Security Beyond the Brute-Force Bound
3. A Survey of Two Signature Aggregation Techniques
4. Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms
5. Secure Complaint-Enabled Source-Tracking for Encrypted Messaging
6. A Novel Related Nonce Attack for ECDSA
7. An Overview of Privacy in Machine Learning
8. Protecting Cryptography Against Compelled Self-Incrimination
9. A review on mathematical strength and analysis of Enigma.
   Preguntas adicionales requeridas:
   - ¿Por qué la máquina Enigma no es un sistema de encriptación seguro según IND-CPA?
   - ¿Qué cambios podrían hacerse a Enigma en términos de aleatoriedad para que esta tuviera mayores posibilidades de ser IND-CPA seguro?
   - Mencione y explique en detalle al menos 3 puntos débiles de Enigma, ya sea de proceso de encriptación o forma de uso de los alemanes, que conllevaron a que pudiese ser quebrantable o que debilitaban el sistema. No es necesario ser demasiado formal matemáticamente, sin embargo intente aplicar la materia vista en el curso de forma teórica.
10. New methods for public key cryptosystems based on XTR