Grado en Matemática Computacional

Proyecto Final de Grado

Criptografía en clave pública y privada. RSA

Supervisor:
Aarón Martínez Romero
Tutor académico:
Fernando Hernando Carrillo
Cotutor académico:
Julio José Moyano Fernández

Autor:
Mar Escobar Benet

Fecha de lectura: 28 de Octubre de 2015
Curso académico 2014/2015

Resumen

En este trabajo se detalla el sistema antifraude ideado para la empresa PayNoPain, la cual, es una pasarela de pago por Internet. Además aprovechamos el trabajo dirigido para explicar y describir los posibles métodos de encriptación que pueden ser útiles en este trabajo. Para empezar, se detalla la historia y evolución de la criptología. Antes de estudiar a fondo la encriptación, se hace un breve repaso sobre la aritmética modular. A continuación, se estudia la encriptación de clave privada y los métodos de encriptación utilizados mediante esta técnica. También se profundiza en la encriptación de clave pública y los esquemas utilizados mediante esta encriptación y, por último, se profundiza en uno de estos métodos, el RSA.

Palabras clave

Encriptación, criptografía, clave pública, clave privada, RSA.

Keywords

Encryption, cryptography, public key, private key, RSA.

Índice general

Capítulo 1

Introducción

La palabra criptografía proviene de la unión de los términos griegos $\kappa\rho\upsilon\pi\tau\omega$ kripto (oculto) y $\gamma\rho\alpha\varphi\omega\varsigma$ graphos (escribir), y su definición es: *escritura oculta*.

Los orígenes de la criptología se remontan en las profundidades de la historia. Desde los tiempos más remotos, las personas han utilizado diversos métodos con el fin de lograr que un mensaje no llegara a manos de personas no autorizadas a leerlo.

Algunos de los testimonios más antiguos sobre la ocultación de la escritura que se conocen se remontan a Herodoto, quien hizo una crónica de los conflictos entre Grecia y Persia en el siglo V a.C. Fue este método el que salvó a Grecia de ser ocupada por Jerjes, Rey de Reyes persa. Herodoto cuenta en su crónica que Demarato, un griego exiliado en la ciudad Persa de Susa, tuvo conocimiento de los preparativos de Jerjes para atacar Grecia y decidió alertar a los espartanos mediante un mensaje oculto en tablillas de madera. El método de ocultación consistió en retirar la cera de las tablillas, escribir el mensaje y luego volver a cubrir con cera.

1.1. Contexto y motivación del proyecto

Hasta hace pocos años la criptología solo resultaba interesante para agencias de seguridad, gobiernos, grandes empresas y delincuentes. Sin embargo, en poco tiempo, y debido al rápido crecimiento de las comunicaciones electrónicas, esta ciencia se ha convertido en un tema central que despierta el interés del público en general. Destaca especialmente el cambio que ha sufrido la investigación en criptología en el último tercio del pasado siglo, ya que ha pasado del tema clásico del cifrado y su seguridad a los más actuales campos de las firmas digitales y los protocolos criptológicos. Dicha variación es una consecuencia inmediata del impacto de las nuevas tecnologías en la sociedad, que cada vez demanda más servicios telemáticos seguros. Así, ante las situaciones de peligro nacidas a raíz de

los nuevos servicios, se hacen necesarias soluciones diferentes.

La aparición de la criptología asimétrica o de clave pública ha permitido que se desarrollen una serie de nuevas tecnologías como firma digital, autentificación de usuarios y el cifrado de datos sin intercambio previo de secretos (clave privada), que es muy importante, en general, en comercio electrónico desde canales inseguros como Internet.

En este trabajo se trata la criptología y en especial la criptología de clave pública y el esquema RSA. El trabajo consta de siete capítulos. En el segundo capítulo, "Estancia en prácticas", se hace un breve resumen del trabajo realizado en la empresa durante la estancia en prácticas. En el tercer capítulo, "Criptología", se presenta una primera visión sobre la criptología, la criptografía y los criptoanálisis. En el cuarto capítulo, "Aritmética modular" se hace una introducción a conceptos básicos que en los siguientes capítulos serán utilizados. En el quinto capítulo, "Clave privada" se explica el fundamento de este sistema, junto con una breve cronología de la evolución de la criptología de clave privada. También se detallan las diferentes técnicas donde se aplica este tipo de encriptación. En el sexto capítulo, "Clave pública" se desarrolla el mismo procedimiento que en el capítulo anterior, aunque sin cronología. Sin embargo, se hace más hincapié en los tipos de criptosistemas de clave pública que existen. Por último, en el capítulo séptimo, "RSA", se estudia más a fondo el criptosistema de clave pública RSA.

Capítulo 2

Estancia en prácticas

En esta sección vamos a detallar mi estancia en prácticas, el objetivo de mi proyecto, el trabajo realizado en la empresa y el aprendizaje obtenido en ella.

2.1. Empresa

Mi estancia en prácticas se ha desarrollado en la empresa PayNoPain. Se trata de una pequeña empresa, situada en uno de los edificios del campus de la Universidad Jaume I, la cual proporciona una pasarela de pago por Internet. Ofrece distintos proyectos vinculados a empresas externas en los cuales predomina el pago seguro por Internet. La empresa consta de 20 trabajadores e internamente se distribuye por distintos sectores que representan las áreas que abarca la empresa.

Durante las 290 horas correspondientes a esta asignatura, yo formaba parte del sector destinado a la pasarela de pago. Y mi labor allí era confeccionar un sistema antifraude. Es decir, proporcionar seguridad en la relación cliente-banco para que ninguno de los dos fuese estafado.

La pasarela de pago es un sistema de puntuaciones dinámicas que monitoriza las transacciones a tiempo real de las procesadoras de pago que trabajan con PayNoPain. Se dirige principalmente a las casas de apuestas y juegos online. Por lo tanto, debe ser un medio de transmisión seguro entre un usuario que realiza pagos por Internet, bien sea un usuario de un casino, o de una casa de apuestas por ejemplo, y la entidad bancaria a la cual está suscrito.

2.1.1. Transacciones

En esta pasarela se trabaja mediante transacciones. Una transacción financiera es un acuerdo, comunicación o movimiento llevado a cabo entre un comprador y un vendedor en la que se intercambian un activo contra un pago. El comprador y el vendedor son entidades u objetos separados, que generalmente intercambian productos de valor, como información, bienes, servicios o dinero. Este tipo de operación se conoce como una transacción de dos partes, siendo la primera parte la entrega de dinero y la parte segunda la recepción de bienes.

En cada transacción se recogen los siguientes datos que son almacenados en la base de datos:

- Importe
- Identificador de usuario (por cada cliente)
- Número de tarjeta (PAN):
  - Banco emisor
  - País de emisión
  - Tipo de tarjeta
- Titular de la tarjeta
- Moneda
- IP
- Cliente

2.1.2. Puntuaciones

Cuando un usuario hace una transacción, los datos pertenecientes a la transacción son evaluados por una serie de reglas, impuestas por la pasarela, que atribuyen una puntuación a la operación que se va a realizar. Dicha puntuación puede estar en tres umbrales distintos:

- Transacción sin riesgo: no hay sospecha de que la operación sea peligrosa y se realiza la solicitud al banco para poder efectuar la operación.
- Transacción sospechosa: no están seguros de que la operación sea fraudulenta y se exigen más datos que puedan proporcionar a la operación más fiabilidad, como por ejemplo, pedir el número PIN de la tarjeta. Se activa un proceso llamado reglas de

negocio, que en lugar de aplicar una medida tan restrictiva como es bloquear una transacción, se buscan otras medidas que no tengan un impacto tan grande en la facturación.

- Transacción frauduenta: se considera la operación como un fraude y no permite hacer la solicitud al banco. Directamente se rechaza el pago.

Cuantos más puntos se consiguen al evaluarse todas las reglas, más probabilidad de ser considerada transacción fraudulenta tiene la operación.

Cada cliente, es decir, un casino, una casa de apuestas, etc., elige la importancia que le atribuye a cada regla dentro de un umbral predefinido por PayNoPain; asigna el umbral de puntuación que cada regla va a tomar.

2.1.3. Formas de identificarse

En la pasarela de pago hay tres formas de identificarse:

- Mediante el número de tarjeta: el usuario registra su número de tarjeta.
- Mediante IP: se accede a su localización a través de la IP con la que está conectado a la pasarela.
- Mediante identificador de usuario: cada cliente tiene una lista de sus usuarios, los cuales se pueden identificar en la pasarela con su identificador propio creado por ese cliente.

2.1.4. Reglas de puntuación actuales

Las reglas de puntuación que evalúan una transacción son las siguientes:

- Número de transacciones correctas realizadas con la misma tarjeta en 24 horas.
- Número de transacciones correctas realizadas a través de la misma IP en 24 horas.
- Número de transacciones correctas realizadas por un identificador de usuario externo con la misma tarjeta en 24 horas.
- Número de transacciones correctas realizadas con la misma tarjeta entre 1 y 90 días.
- Número de transacciones correctas realizadas a través de la misma IP entre 1 y 90 días.

- Número de transacciones correctas realizadas por un identificador de usuario externo con la misma tarjeta entre 1 y 90 días.
- Número de transacciones correctas realizadas con la misma tarjeta en un minuto.
- Número de transacciones correctas realizadas a través de la misma IP en un minuto.
- Número de transacciones correctas realizadas por un identificador de usuario externo con la misma tarjeta en un minuto.
- Número de transacciones incorrectas realizadas con la misma tarjeta en 24 horas.
- Número de transacciones incorrectas realizadas a través de la misma IP en 24 horas.
- Número de transacciones incorrectas realizadas por un identificador de usuario externo con la misma tarjeta en 24 horas.
- Número de transacciones incorrectas realizadas con la misma tarjeta entre 1 y 90 días.
- Número de transacciones incorrectas realizadas a través de la misma IP entre 1 y 90 días.
- Número de transacciones incorrectas realizadas por un identificador de usuario externo con la misma tarjeta entre 1 y 90 días.
- Frecuencia máxima permitida de pagos a través de la misma IP.
- Frecuencia máxima permitida de pagos con una misma tarjeta.
- Número de pagos semanales superiores a una cantidad determinada siendo esta la habitual para el tipo de comercio.
- Número de pagos mensuales superiores a una cantidad determinada siendo esta la habitual para el tipo de comercio.
- Número máximo de tarjetas distintas que pueden ser usadas a través de una misma dirección IP.
- Número máximo de IPs distintas que pueden usar la misma tarjeta.
- Importe de una transacción superior a una cantidad determinada.
- Tarjetas distintas usadas para un mismo identificador de cliente en los últimos 90 días.
- No coincidencia del país emisor de la tarjeta y el país de origen de la dirección IP.
- Uso de Proxy. (Un Proxy, o servidor proxy, en una red informática, es un servidor, que hace de intermediario en las peticiones de recursos que realiza un cliente a otro servidor.)

- Uso de un nodo de la red TOR. (Es una red de comunicaciones distribuida de baja latencia y superpuesta sobre internet, en la que el encaminamientos de los mensajes intercambiados entre los usuarios no revela su identidad, es decir, su dirección IP y que mantiene la integridad y el secreto de la información que viaja por ella.)
- Uso sucesivo de tarjetas con numeraciones muy similares o secuenciales.
- Coincidencia de algunos de los campos con las listas negras (identificador de usuario, número de tarjeta, IP o país).
- Tarjeta usada por distintos identificadores de usuario en los últimos 90 días.
- Fiabilidad del país de procedencia de la transferencia.
- IP utilizada por otro usuario en un máximo de tres días.
- Tipo de dispositivo de conexión que utiliza el usuario.
- Suma en céntimos de los últimos importes en un máximo de 30 días.
- Umbral de puntuación para que una transacción sea sospechosa o fraudulenta.

2.2. Desarrollo del proyecto

La finalidad de mi proyecto era diseñar un sistema antifraude con soporte en algoritmos y funciones matemáticas que se basara en la experiencia, es decir, que mediante la información almacenada en la base de datos de anteriores transacciones, fuera capaz de determinar si la nueva transacción era fraudulenta o no, comparando si se asemejaba a comportamientos anteriormente registrados.

2.2.1. Propuestas de desarrollo del proyecto

Para poder llevar a cabo el proyecto se plantearon dos propuestas: las redes neuronales o una función de regresión logística.

Redes neuronales

Las redes neuronales son un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. Su gran interés es el aprendizaje que obtienen gracias al procesamiento automático. Imitan el funcionamiento de las redes neuronales de los organismos vivos: un conjunto de neuronas conectadas entre sí y que trabajan en conjunto, sin que haya una tarea concreta para cada una. Con la experiencia, las neuronas

van creando y reforzando ciertas conexiones para aprender algo que se queda fijo en el tejido.

Su esencia está basada en matemáticas y estadística. Se trata de una idea sencilla: dados unos parámetros hay una forma de combinarlos para predecir un cierto resultado, pero el problema reside en cómo combinarlos. Encontrar la combinación que mejor se ajusta es entrenar la red neuronal. Una red ya entrenada se puede usar para hacer predicciones o clasificaciones, es decir, para aplicar la combinación.

Las redes neuronales tienen la capacidad de aprender mediante una etapa denominada *etapa de aprendizaje*. Consiste en proporcionar a la red neuronal datos como entrada al mismo tiempo que se le indica cuál es la salida esperada. Además se pueden conseguir respuestas en tiempo real.

Función de regresión logística

Una función de regresión simple es un modelo que representa la dependencia lineal de una variable respuesta y, respecto a otra variable explicativa x.

El nombre de modelo de regresión proviene de los trabajos de Galton en biología a finales del siglo XIX. Galton estudió la dependencia de la estatura de los hijos (y) respecto a la de sus padres (x), encontrando lo que denominó una regresión a la media: los padres altos tienen, en general, hijos altos, pero, en promedio, no tan altos como sus padres; los padre bajos tienen hijos bajos, pero, en promedio, más altos que sus padres. Desde entonces, los modelos estadísticos que explican la dependencia de una variable y respecto de una o varias variables cuantitativas x se denominan modelos de regresión.

La regresión logística es una de las técnicas estadístico-inferenciales más empleadas en la producción científica contemporánea. Surge en la década del 60, su generación dependía de la solución que se diera al problema de la estimación de los coeficientes. El algoritmo de Walker-Duncan para la obtención de los estimadores de máxima verosimilitud vino a solucionar en parte este problema, pero era de naturaleza tal que el uso de computadoras era imprescindible.

La identificación del mejor modelo de regresión logística se realiza mediante la comparación de modelos utilizando el cociente de verosimilitud, que indica a partir de los datos de la muestra cuanto más probable es un modelo frente al otro.

El modelo general de regresión es la extensión para k variables explicativas del modelo simple para una. En general, una variable respuesta y, depende de muchas otras variables $x_1, x_2, ..., x_n$, aunque algunas de estas variables pueden ser no observables o, incluso, desconocidas para el investigador.

Supondremos que, en el rango de valores de interés, la función f admite una aproxi-

mación lineal, con lo que resulta el modelo de regresión múltiple:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \text{ donde } \beta_i \in \mathbb{R}$$

Algunos ejemplos donde se utiliza este modelo son:

- determinar la influencia sobre el rendimiento de un proceso en función de la temperatura, la presión, la humedad relativa y el tiempo de operación.
- explicar la remuneración de los puestos directivos en las empresas españolas en función de las características del individuo que lo ocupa (edad, titulación, años de experiencia, etc.), del puesto analizado (número de personas que dependen del puesto, nivel jerárquico, etc.), y de la empresa (sector, tamaño, beneficios, ...).
- estudiar la cantidad de lluvia recogida en función de variables climáticas y variables que describen métodos artificiales de producción de lluvia mediante iodato de plata.
- explicar el rendimiento escolar mediante variables de la escuela (materiales utilizados, formación y motivación del profesorado, ...), de la familia del estudiante, y de sus amigos en clase.

Para construir un modelo de regresión múltiple, en primer lugar, estableceremos un conjunto de hipótesis respecto a la distribución de la perturbación, y la relación entre la variable dependiente y las independientes; en segundo lugar, tomaremos una muestra y estimaremos los parámetros del modelo, construyendo intervalos de confianza para describir la incertidumbre presente en su estimación; finalmente, se contrastará la validez de las hipótesis en que nos hemos basado para realizar la estimación del modelo.

Supondremos que tenemos k variables matemáticas $(x_1, x_2, ..., x_k)$, y una aleatoria y. A las variables x se las llama *variables explicativas, exógenas, independientes o regresores*, y a la y, *variable explicada, endógena, respuesta o dependiente.*

Admitiremos que, una observación cualquiera puede escribirse:

$$y_i = \beta_0 + \beta_1 x_{1i} + ... + \beta_k x_{ki}$$

donde cada coeficiente β_i mide el efecto marginal sobre la respuesta de un aumento unitario en x_i cuando todas las otras variables permanecen constantes.

Si las variables explicativas pueden estar relacionadas entre sí debemos estudiarlas conjuntamente.

Sea Y una variable dependiente binaria (con dos posibles valores: 0 y 1). Sean un conjunto de k variables independientes, $(X_1, X_2, ..., X_k)$, observadas con el fin de predecir/explicar el valor de Y.

El objetivo consiste en determinar:

$$P[Y = 1/X_1, X_2, ..., X_k] \mapsto P[Y = 0/X_1, X_2, ..., X_k] = 1 - P[Y = 1/X_1, X_2, ..., X_k]$$

Para ello, se construye el modelo $P[Y = 1/X_1, X_2, ..., X_k] = p(X_1, X_2, ..., X_k; \beta)$ donde:

$$p(X_1, X_2, ..., X_k; \beta) : R^k \longrightarrow [0, 1]$$

que depende de un vector de parámetros $\beta = (\beta_1, \beta_2, ..., \beta_k)$.

Con el fin de estimar $\beta = (\beta_1, \beta_2, ..., \beta_k)$ y analizar el comportamiento del modelo estimado se toma una muestra aleatoria de tamaño n dada por $(x_i, y_i)_{i=1,2,...,n}$ donde el valor de las variables independientes es $x_i = (x_{i1}, x_{i2}, ..., x_{ik})$ e $y_i \in [0, 1]$ es el valor observado de Y en el i-ésimo elemento de la muestra.

Como $(Y/X_1, X_2, ..., X_k) \in B[1, p(X_1, X_2, ..., X_k; \beta)]$ la función de verosimilitud viene dada por:

$$L[\beta/(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)] = \Pi_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1-y_i}$$

donde $p_i = p(x_i; \beta) = p[(x_{i1}, x_{i2}, ..., x_{ik}); \beta]_{i=1,2,...,n}$

El modelo logístico establece la siguiente relación entre la probabilidad de que ocurra el suceso, dado que el individuo presenta los valores $(X_1 = x_1, X_2 = x_2, ..., X_k = x_k)$:

$$P[Y = 1/x_1, x_2, ..., x_k] = \frac{1}{1+e^{(-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - ... - \beta_k x_k)}}$$

El objetivo es hallar los coeficientes $(\beta_0, \beta_1, ..., \beta_k)$ que mejor se ajusten a la expresión funcional.

Se conoce como ratio de riesgo al cociente de probabilidades:

$$\frac{P[Y=1/X_1,X_2,...,X_k]}{1-P[Y=1/X_1,X_2,...,X_k]} = \frac{p(X_1,X_2,...,X_k;\beta)}{1-p(X_1,X_2,...,X_k;\beta)} = e^{\beta_1+\beta_2 x_2+...+\beta_k x_k}$$

se toma como primera variable explicativa la variable constante que vale 1.

Tomando logaritmos neperianos en la expresión anterior, se obtiene una expresión lineal para el modelo:

$$L[P(Y=1)] = Ln[\frac{P[Y=1/X_1,X_2,...,X_k]}{1-P[Y=1/X_1,X_2,...,X_k]}] = \beta_1 + \beta_2 X_2 + ... + \beta_k X_k$$

Esta función logística antes descrita es la utilizada en los cálculos para predecir si la transacción era fraudulenta o no.

2.2.2. PHP

En PayNoPain utilizan como lenguaje de programación *PHP* y su filosofía de trabajo está basada en el desarrollo tipo *TDD*. Ambos conceptos eran desconocidos para mi ya que anteriormente no había trabajado con ellos.

PHP es un lenguaje de *scripting* de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser implementado en páginas HTML. Pretende facilitar a los desarrolladores web la escritura de forma dinámica y rápida de páginas web.

La filosofía *TDD* (*Text-driven development*) también conocido como desarrollo guiado por pruebas de software, es una práctica de ingeniería del software que involucra otras dos prácticas: escribir los casos primero y refactorizar. Primero se escribe un caso del problema que quieres solucionar y se comprueba que el caso fallan. Luego, se implementa el código necesario para hacer que el primer caso pase satisfactoriamente y cuando esto sucede, se refactoriza el código escrito. Refactorizar el código significa limpiar el código sin modificar su funcionalidad, pretendiendo conseguir una consistencia interna y mayor claridad en el código.

La idea de la filosofía *TDD* es lograr un código limpio que funcione, es decir, que los requisitos sean traducidos a casos y así, cuando todos los casos posibles que tiene el problema que queremos solucionar pasen como satisfactorios se garantizará que el software cumple con los requisitos que se han establecido.

El ciclo que sigue siempre esta filosofía es:

1. Elegir un requisito: se elige de una lista el requerimiento que se cree que nos dará mayor conocimiento del problema y que a la vez sea fácilmente implementable. Se suele empezar con los casos base de los programas.
2. Escribir una prueba: se escribe una prueba para el requisito anteriormente implementado.
3. Verificar que la prueba falla: si la prueba no falla es porque el requisito ya está anteriormente implementado o porque la prueba es errónea.
4. Escribir la implementación: escribir el código más sencillo que haga que la prueba pase satisfactoriamente.

5. Ejecutar las pruebas automatizadas: verificar si todo el conjunto de pruebas funcionan correctamente.

6. Eliminación de duplicación: refactorizar el código para poderlo hacer más simple y más legible.

7. Actualización de la lista de requisitos: actualizar la lista de requisitos omitiendo el ya implementado.

Para poder trabajar estos conceptos, hice algunos ejercicios, como por ejemplo: hacer conversiones de números a números romanos.

2.2.3. Base de datos

Los datos necesarios para mi proyecto se encuentran en la base de datos de pruebas correspondiente a las transacciones. En ella podemos encontrar los siguientes campos:

- id: clave principal que identifica al usuario en su base de datos.
- cli_id: identificador del cliente
- ext_id: identificador del usuario dentro del cliente.
- ord_id: identificador del código de la orden que el banco le ha atribuido a la operación.
- card_pan: número pan de la tarjeta encriptado, aunque se puede desencriptar.
- card_hash: número pan de la tarjeta encriptado, a partir del card_pan con un hash.
- card_expire: fecha de caducidad de la tarjeta.
- card_holder: propietario de la tarjeta.
- card_country: país de expedición de la tarjeta según el código ISO 3166
- card_type: tipo de tarjeta.
- card_bin: seis primeros dígitos de la tarjeta que proporcionan el banco de donde procede, el país y el propietario.
- merchant_code: código que proporciona el banco que representa al cliente, es decir, cuando vas a crear una relación de transferencias con un cliente por Internet, el banco te proporciona un identificador directo de las transacciones con un cliente determinado.

- response_code: código que se atribuye a la respuesta que el banco ha proporcionado sobre ese movimiento. Es decir, si está bien o no y en caso de no estar bien, representan el porqué falla. Son códigos universales para todos los bancos.
- authorisation_code: código de autorización que proporciona el banco. Son únicos, pero cada seis meses pierden su caducidad.
- date_created: fecha y hora de creación de la transacción.
- amount: cantidad en céntimos.
- currency: tipo de moneda de pago.
- type: código de tipo de pago.
- secure: número binario que representa la utilización de PIN en la transacción.
- risk: número binario que representa si la transacción ha sido considerada de riesgo.
- fraud: número binario que representa si la transacción ha sido considerada de fraude.
- fraud_score: puntuación obtenida tras evaluar las reglas puntuales antifraude.
- fraud_rules: reglas que han obtenido una puntuación elevada y han producido la puntuación de fraude.
- remote_address: IP desde donde el usuario efectúa el pago.
- test: número binario que representa si la transacción está en modo prueba o no.
- status: representa el estado de la operación tras realizar todos los pasos. Si es 1 quiere decir que la pasarela ha cobrado y la transacción ha ido bien; por el contrario, si es 0, puede haber sido considerado el pago como un fraude o simplemente ha sido incorrecto.

Para poder trabajar con la base de datos hacía consultas SQL para poder obtenerlos. Las consultas SQL son la forma de recuperar la información que nos interesa de una base de datos.

2.2.4. Perfiles

En el proyecto se distinguían tres perfiles distintos: perfil usuario, perfil cliente o perfil PayNoPain. Estos perfiles indicaban los datos que se iban a considerar como experiencia.

Cuando se realiza una transacción nueva se comprueba que el identificador ya pertenezca a la base de datos, en ese caso, se considera el perfil usuario como perfil de experiencia. En caso de que no aparezca en la base de datos, se asumirá como experiencia

el perfil de cliente, es decir, se comparará con el comportamiento habitual registrado en ese comercio. En caso de no tener constancia de que el comercio ha trabajado con la pasarela, se considerará el perfil PayNoPain que recoge la experiencia de toda la base de datos.

A continuación pasamos a detallar las reglas correspondientes a cada perfil.

Perfil usuario

Las consideraciones que se tienen en este perfil son las siguientes:

- Relación entre el número de transacciones almacenadas en la base de datos relacionadas con el usuario y la cantidad de transacciones que se han realizado con la tarjeta de la nueva transacción, de la nueva entrada.
- Distancia entre el país en el que se ha realizado el pago y el país de procedencia de la tarjeta.
- Diferencia en minutos entre la hora de la transacción y la hora punta habitual para el usuario. Se considerarán cuatro franjas horarias en las 24 horas del día y por tanto, se considerarán cuatro horas punta distintas. Se evaluará a qué franja horaria pertenece cada transacción y se calculará la diferencia en minutos.
- Diferencia entre las transacciones realizadas en un plazo de tiempo, sea un mes o un día, y el número de transacciones que el cliente considera que son habituales para ese periodo de tiempo.
- Diferencia entre el dinero gastado en un plazo de tiempo, sea un mes o un día, y el dinero que el cliente considera que es habitual gastar en ese periodo de tiempo.
- Relación entre todas las transacciones que ha realizado y las transacciones que han sido realizadas mediante una cantidad habitual.
- Relación entre la cantidad de transacciones que ha realizado el usuario y las que han sido consideradas de riesgo.
- Relación entre la cantidad de transacciones que ha realizado el usuario y las que han sido consideradas de fraude.
- Fiabilidad del dispositivo utilizado para realizar el pago, es decir, se considera más fiable si se trata de un dispositivo móvil; por el contrario, si se trata de un ordenador, la transacción pierde fiabilidad.
- Fiabilidad del tipo de red utilizado para realizar la transacción. Se considera menos fiable si la red es considerada de tipo TOR.
- Fiabilidad de la operación, es decir, se considera más fiable si en la operación se ha requerido el número PIN.

Perfil cliente

Las consideraciones que se tienen en este perfil son las siguientes:

- Relación entre el número de transacciones almacenadas en la base de datos relacionadas con el cliente y la cantidad de transacciones que se han realizado con la tarjeta de la nueva transacción, de la nueva entrada.
- Distancia entre el país en el que se ha realizado el pago y el país de procedencia de la tarjeta.
- Diferencia en minutos entre la hora de la transacción y la hora punta habitual para el cliente. Se considerarán cuatro franjas horarias en las 24 horas del día y por tanto, se considerarán cuatro horas punta distintas. Se evaluará a qué franja horaria pertenece cada transacción y se calculará la diferencia en minutos.
- Relación entre todas las transacciones que ha realizado y las transacciones que han sido realizadas mediante una cantidad habitual.
- Fiabilidad del dispositivo utilizado para realizar el pago, es decir, se considera más fiable si se trata de un dispositivo móvil; por el contrario, si se trata de un ordenador, la transacción pierde fiabilidad.
- Fiabilidad del tipo de red utilizado para realizar la transacción. Se considera menos fiable si la red es considerada de tipo TOR.
- Fiabilidad de la operación, es decir, se considera más fiable si en la operación se ha requerido el número PIN.

Perfil PayNoPain

Las consideraciones que se tienen en este perfil son las siguientes:

- Relación entre el número de transacciones almacenadas en la base de datos y la cantidad de transacciones que se han realizado con la tarjeta de la nueva transacción, de la nueva entrada.
- Distancia entre el país en el que se ha realizado el pago y el país de procedencia de la tarjeta.
- Fiabilidad del dispositivo utilizado para realizar el pago, es decir, se considera más fiable si se trata de un dispositivo móvil; por el contrario, si se trata de un ordenador, la transacción pierde fiabilidad.
- Fiabilidad del tipo de red utilizado para realizar la transacción. Se considera menos fiable si la red es considerada de tipo TOR.

- Fiabilidad de la operación, es decir, se considera más fiable si en la operación se ha requerido el número PIN.

2.2.5. R

R es un programa matemático usado para cálculos estadísticos. Para nosotros era de gran importancia porque una vez introducidos los datos en un formato muy particular obteníamos inmmediatamente la función de regresión logística. Aunque puede ser implementado en Python con el fin de no llamar a R. Por tanto, debíamos adaptar los datos obtenidos en la base de datos e introducirlos en el programa.

Además, este programa dispone de un módulo específico para realizar las regresiones logísticas, este módulo es el *R Commander*. Así pues, una vez obtenidos los datos e introducidos en el programa *R Commander* se podían realizar estudios sobre regresiones. Fue gratificante ver que realmente predecía que las transacciones almacenadas como fraude tenía una probabilidad muy alta de ser fraude, es decir, hacía las predicciones bien y podía utilizar este programa con esta metodología para mi proyecto.

Antes de poder confirmar que, efectivamente, era una buena opción utilizar este programa, hicimos pruebas con varios usuarios y varios perfiles y así poder cerciorarnos de que los resultados eran buenos. De esta forma podíamos predecir para una nueva entrada, mediante las variables que el programa calculaba basándose en los datos que le habíamos introducido, si resultaba ser una transacción fraudulenta o no.

2.2.6. Consultas

A la hora de implementar las consultas para obtener los datos de la base de datos, nos dimos cuenta que muchas de las características que habíamos considerado en los perfiles no eran viables y proporcionaban poca información. Además, también nos percatamos que realmente eran necesarias las mismas consultas para los tres perfiles distinguiendo únicamente de quién queríamos obtener los datos, es decir, si los datos queríamos que fueran de un usuario únicamente, o de un cliente o, por el contrario, queríamos los datos de toda la base de datos.

Para obtener todos los datos necesarios para los perfiles hicimos esta serie de consultas:

- Identificador en la base de datos.
- Número de transacciones realizadas, que se han almacenado hasta ese momento en la base de datos.
- Binario que indica si en la transacción se ha utilizado PIN.

- Número de transacciones almacenadas hasta ese momento en la base de datos que han utilizado PIN.
- Binario que indica si la transacción ha sido considerada de riesgo.
- Binario que indica si la transacción ha sido considerada de fraude.
- Número de transacciones almacenadas hasta ese momento en la base de datos que han sido consideradas de riesgo.
- Número de transacciones almacenadas hasta ese momento en la base de datos que han sido consideradas de fraude.
- Cantidad de dinero perteneciente ha cada transacción.
- Cantidad de dinero más veces utilizada según todas las cantidades almacenadas hasta ese momento en la base de datos.
- Número de transacciones realizadas almacenadas hasta ese momento en la base de datos con la cantidad utilizada en la nueva transacción.
- Cantidad de dinero, almacenada hasta ese momento en la base de datos, gastado en un mes.
- Cantidad de dinero, almacenada hasta ese momento en la base de datos, gastado en un día.
- Cantidad de dinero considerada, almacenada hasta ese momento en la base de datos, que se prevé gastar en un mes.
- Cantidad de dinero considerada, almacenada hasta ese momento en la base de datos, que se prevé gastar en un día.
- Tarjeta utilizada en cada transacción.
- Número de tarjetas distintas utilizadas, almacenadas hasta ese momento en la base de datos.
- Número de transacciones que se han realizado, almacenadas hasta ese momento en la base de datos, con la tarjeta utilizada en la nueva transacción.
- Diferencia en minutos entre la hora considerada como hora habitual, según las horas almacenadas hasta el momento en la base de datos, y la hora a la que se realiza cada transacción.
- Distancia en kilómetros entre el país de expedición de la tarjeta y el país de procedencia de la IP mediante la cual se ha realizado el pago.
- Diferencia entre la cantidad perteneciente a cada transacción y la cantidad considerada como cantidad habitual.

- Diferencia entre la cantidad de dinero que se ha gastado en un día para cada transacción y la cantidad prevista que en un día se debe gastar.
- Diferencia entre la cantidad de dinero que se ha gastado en un mes para cada transacción y la cantidad prevista que en un mes se debe gastar.
- Número de transacciones realizadas con la cantidad considerada como cantidad habitual.
- Relación entre el número de transacciones consideradas como transacción de riesgo y el número de transacciones realizadas.
- Relación entre el número de transacciones consideradas como transacción de fraude y el número de transacciones realizadas.
- Relación entre el número de transacciones en las que se ha utilizado PIN y el número de transacciones realizadas.
- Relación entre el número de tarjetas distintas utilizadas y el número de transacciones realizadas.
- Relación entre el número de transacciones realizadas con la tarjeta de la nueva transacción y el número de transacciones realizadas.
- Relación entre el número de transacciones realizadas con la cantidad de dinero perteneciente a la nueva transacción y el número de transacciones realizadas.

2.2.7. Utilización de los datos

Con todos estos datos obtenidos mediante las consultas, se creó un documento tipo *txt* para poderlo procesar con el programa *R*.

Por falta de tiempo no se pudo implementar el script que realizase los cálculos, mediante este documento antes convertido, en el programa *R*. Aunque cabe destacar que, una alternativa a la creación de este script podría ser la resolución del problema mediante una función de mínimos cuadrados que nos calculase las mismas variables que *R* nos puede calcular.

Siendo que la parte experimental con el *R* estaba probada, la implementación del script puede ser abordada por un informático ya que, antes de terminar mi estancia en prácticas, configuré un tutorial sobre los pasos necesarios en el programa *R* y *R Commander* para poder hacer los cálculos.

El objetivo de mi proyecto fue alcanzado ya que encontré la forma de poder predecir mediante la experiencia, si se trataba de una transacción fraudulenta o no.

2.3. Aprendizaje del proyecto

Durante mi estancia en prácticas he aprendido conceptos que antes desconocía, como puede ser la programación en *PHP* o la metodología *TDD*. Además, asistí a los cursos de formación que la propia empresa ofertaba y aprendí a refactorizar códigos y poderlos hacer más legibles y más claros.

He tenido una buena experiencia en la empresa y junto a nuevos conocimiento también he adquirido amistades nuevas con los compañeros y un poco más de práctica jugando al ping-pong.

2.4. Motivación para el TFG

Cuando se realiza una transacción comercial a través de Internet, la información que contiene la tarjeta de crédito se codifica electrónicamente, a fin de proteger la identidad del cliente y la integridad de la transacción.

Dado que parte de los datos que se almacenan en la base de datos han de estar encriptados hemos decidido realizar nuestro trabajo sobre criptología, y en particular, nos centraremos en el RSA.

Además veremos cómo usar el RSA como firma digital y hacer más segura la comunicación, mejorando el sistema antifraude.

Capítulo 3

Criptología

En alguna ocasión debes haber querido enviar un mensaje a un amigo y deseado que ningún intruso conozca el contenido. En alguna otra ocasión, incluso tú mismo has sido el intruso que intentaba descubrir el contenido de mensajes. Los seres humanos, a través de la historia, han inventado mecanismos para proteger los mensajes y ponerlos a salvo del ataque de instrusos. Y, como intrusos, también han utilizado su inteligencia para descifrar mensajes supuestamente bien protegidos. No poco esfuerzo se invierte en esta tarea, ya que muchas veces, lo que se desea proteger es de gran valor, como la identidad de un ser humano, la seguridad de una transacción comercial o el posicionamiento de un ejército en una guerra mundial. La ciencia que protege y pone al descubierto la información es la criptología.

En la historia universal hay eventos en los que la criptología ha jugado roles de transcendental importancia. Durante la segunda guerra mundial, los Estados Unidos usaron el lenguaje de los indios navajos, con traductores navajos, para enviar mensajes a los comandos en el frente del Pacífico. Ni japoneses ni alemanes pudieron descifrar la compleja sintaxis del lenguaje. También es conocido que durante la guerra mundial, la inteligencia británica, con ayuda del espionaje checoslovaco, fue capaz de descifrar los mensajes codificados del alto comando alemán a la flota del Atlántico.

En esta sección vamos a desarrollar el fundamento teórico que hemos trabajado en este proyecto. Nos vamos a centrar en hablar sobre la criptología, la encriptación de claves, el tipo de encriptación que puede haber según la clave y me centraré en el *RSA*.

3.1. Introducción

La criptología es la ciencia que estudia los criptosistemas, o también conocidos como los sistemas critográficos o códigos secretos, es decir, es la disciplina científica que estudia

la escritura secreta, mensajes que, al ser procesados de cierta manera, se convierten en difíciles o imposibles de leer por entidades no autorizadas. Dentro de esta ciencia, se distinguen dos ramas fundamentales, que son: la *criptografía* y el *criptoanálisis*. La criptografía se encarga de concebir e implementar los criptosistemas. El criptoanálisis se constituye por métodos que se encargan de romper los criptosistemas.

La criptología tienen una historia milenaria y sus inicios se pierden en el alba de la civilización. Se ha rastreado su origen en incripciones funerarias egipcias en las que la escritura jeroglífica habitual era substituida por otra diferente. Sin embargo, el propósito no era propiamente criptográfico, sino una especie de juego o desafío al lector.

Cabe destacar, que inscripciones con el mismo carácter esotérico para iniciados, han seguido empleándose a lo largo de la historia en epitafios funerarios, así como en diferentes propuestas de escritura secreta. Además, numerosos literatos, como por ejemplo Casanova, Allan Poe, ..., eran criptólogos aficionados y dejaban constancia en sus obras de diversos métodos de cifrado.

3.2. Criptografía

El primer método de criptografía conocido, data del siglo V a.C. y era conocido como "Escítala", un sistema de criptología utilizado por los éforos espartanos para el envío de mensajes secretos. El sistema consistía en dos varas del mismo grosor que se entregaban a los participantes de la comunicación. Para enviar un mensaje se enrollaba una cinta de espiral a uno de los bastones y se escribía el mensaje longitudinalmente, así en cada vuelta de cinta aparecía una letra cada vez. Una vez escrito el mensaje, se desenrollaba la cinta y se enviaba al receptor, que sólo tenía que enrollarla a la vara gemela para leer el mensaje original. Aunque el mensajero fuera atrapado, el mensaje seguiría estando protegido por el cifrado que lo definía.

El objetivo clásico de la criptografía es el intercambio de mensajes a través de un canal seguro. Tradicionamente tal objetivo se conseguía poniéndose, a priori, de acuerdo emisor y receptor en una cierta información secreta, la *clave* o *llave*, que permita cifrar los mensajes anteriores.

Los sistemas criptográficos clasicos, hoy denominados de *clave privada* se revelaron insuficientes o inadecuados para las nuevas necesidades; por ello, aparecieron los denominados sistemas criptográficos de *clave pública*. Son estos los que necesitan un substrato matemático más fuerte y los más estudiados.

Claude Shannon, conocido como el padre de la Teoría de la Información, se ocupó del problema de la Criptografía. Para él, la Teoría de la Información y la Criptografía están relacionados y por eso, lo que estudió fueron los sistemas criptográficos desde el punto de vista de la Teoría de la Información. La hipótesis que predominaba en sus trabajos

era, que el adversario que eventualmente intercepta el canal, tiene una cantidad ilimitada de conocimientos y capacidad de cálculo. Con esta hipótesis, demuestra la existencia de criptosistemas incondicionalmente seguros, como son los sistemas de secreto perfecto, donde el tamaño de la clave es al menos tan grande como el del mensaje intercambiado y además se utiliza una única vez, este tipo de sistemas tienen llaves de un solo uso.

Los criptosistemas de cave pública están basados en la Teoría de la Complejidad Computacional y tratan de conseguir que el descifrado del mensaje secreto resulte imposible en la práctica, a menos de poseer una cierta información suplementaria que solo posee el receptor legal.

La criptología cubre hoy en día objetivos distintos sobre la transmisión secreta de información a diferencia de los objetivos clásicos. Este tipo de aplicaciones se engloban dentro de lo que se denominan *protocolos criptográficos.*

Un protocolo es un conjunto bien definido de etapas, en las que dos o más personas se implican y acuerdan realizar una tarea específica. Un protocolo criptográfico es un protocolo que utiliza como herramienta algún algoritmo criptográfico.

Algunos de estos protocolos son:

- Protocolos de Autentificación: el concepto de autentificación puede aludir al mensaje tratando de garantizar que éste no ha sido alterado (*autentificación de mensaje*) o a la identidad del remitente (*autentificación de usuario*). La identificación del usuario puede ser directa, comprobando una característica propia de aquel, como la *firma digital* o, por el contrario, indirecta, donde el usuario demuestra estar en posesión de una pieza secreta de información.

- Protocolos para compartir secretos: distribuir un cierto secreto entre un conjunto P de participantes de forma que ciertos subconjuntos prefijados de P puedan, uniendo sus participaciones, recuperar dicho secreto.

- Pruebas de conocimiento cero: permite a un individuo convencer a otro de que posee una cierta información sin revelarle nada sobre el contenido de la misma.

- Transacciones electrónicas seguras: permite realizar de forma electrónicamente segura las operaciones bancarias habituales: firma electrónica de contratos, etc.

- Elecciones electrónicas: permite realizar un proceso electoral electrónicamente, garantizando la deseable privacidad de cada votante y la imposibilidad de fraude.

3.3. Criptosistemas

Los criptosistemas pretenden modificar (enmascarar) el mensaje a enviar, de manera que resulte ininteligible para un eventual interceptor, pero permitiendo que el legítimo receptor del mismo pueda, con un esfuerzo razonable, recuperar la información original. Para conseguir esto, la criptografía recurre a técnicas matemáticas más o menos sofisticadas.

Un criptosistema es una terna $(\mathcal{M}, \mathcal{C}, \mathcal{K})$, donde:

- $\mathcal{M}$ es el conjunto de mensajes originales (o en claro);
- $\mathcal{C}$ es el conjunto de mensajes cifrados;
- $\mathcal{K}$ es un conjunto finito de llaves (o claves);

junto con dos funciones:

$$\text{Cifrado: } c : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$$
$$\text{Descifrado: } d : \mathcal{C} \times \mathcal{K} \to \mathcal{M}$$

tales que $d(c(\mathcal{M}, k)) = M$ para todo $(\mathcal{M}, k) \in \mathcal{M} \times \mathcal{K}$.

Un elemento M del conjunto $\mathcal{M}$ se denomina habitualmente mensaje en claro y es una sucesión finita de signos o letras de un cierto alfabeto $\mathcal{A}$. El resultado de aplicar a M la función c de cifrado da lugar a un mensaje cifrado C, que es también una colección de signos en un segundo alfabeto $\mathcal{B}$. La función c depende de un parámetro o llave $k \in \mathcal{K}$. El mensaje original M se recupera a partir de C mediante la función d de descifrado.

Como hemos comentado anteriormente, el propósito del criptosistema es descubrir el contenido del mensaje cifrado y/o de la llave empleada, además de alterar o perturbar el proceso de comunicación. Los tipos de ataques que un criptosistema puede realizar se engloban en dos tipo:

- Activos: el criptoanalista lleva a cabo actividades que perjudican a la comunicación, por ejemplo, hacerse pasar por un transmisor autorizado, intentar substituir el mensaje por otro distinto, etc.
- Pasivos: el criptoanalista se limita, a partir de un mensaje cifrado $\mathcal{C}$, a intentar recuperar el mensaje en claro M o conseguir la clave k. Estos ataques pueden clasificarse en tres grupos:
    - ataque a texto cifrado conocido: el atacante conoce solamente cierta cantidad de texto cifrado;

- ataque a texto claro conocido: el criptoanalista conoce cierta cantidad de texto claro y su correspondiente cifrado;
- ataque a texto claro elegido: el criptoanalista puede elegir de forma arbitraria un texto claro y obtener su correspondiente cifrado.

Capítulo 4

Aritmética modular

En esta sección vamos a introducir la aritmética modular y los conceptos relacionados que serán necesarios a lo largo de la memoria.

La aritmética modular es también conocida por la aritmética del reloj por su analogía con el comportamiento de un reloj con sus horas. Cuando a las 10 de la mañana se le agregan 5 horas se llega a las 3 de la tarde, es decir $10 + 5 = 3$. También si a las 2 de la tarde se le quitan 4 horas, el resultado es las 10, lo que equivale a decir que $2 - 4 = 10$. Esta aritmética del reloj se le llama más generalmente *aritmética módulo 12* y se realiza dentro del conjunto $\mathbb{Z}_{12} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ cuyos elementos se llaman *enteros módulo 12*. En realidad, cualquier número entero es *equivalente* a un entero módulo 12 que se obtiene como el residuo (nunca negativo) de la división entera por 12. Por ejemplo, 29 es equivalente a 5 módulo 12 y se escribe como $29 \equiv 5 \ (mod\ 12)$, porque al dividir $\frac{29}{12}$ da resto 5. Esto también se expresa como $mod(29, 12) = 5$.

Esto es un ejemplo de un caso particular de la aritmética modular. Veremos detalladamente todo lo que la aritmética modular conlleva.

4.1. Congruencia de números enteros

Definición 1. *Se llama relación de equivalencia sobre un conjunto A a cualquier relación R entre sus elementos que verifica las siguientes propiedades:*

1. *Reflexiva: aRa, para cualquier $a \in A$.*
2. *Simétrica: si $a, b \in A$ y aRb entonces bRa.*
3. *Transitiva: si $a, b, c \in A$ y aRb y bRc, entonces aRc.*

Una relación R sobre un conjunto A produce una partición del conjunto en subconjuntos disjuntos, llamados *clases de equivalencia*, cada uno de ellos formado por elementos que están relacionados entre sí. Esta partición se representa por A/R y se llama *conjunto cociente*.

Definición 2. *Dado un número entero fijo $p > 1$ y dos números enteros cualesquiera $a, b \in \mathbb{Z}$, se dice que a es congruente con b módulo p, y se indica como $a \equiv b \ (mod\ p)$, si $p|(a-b)$.*

Es fácil ver que $a \equiv b \ (mod\ p)$ si y solo si coinciden los restos de dividir los números a y b por p, que se llaman *residuos módulo p*. En módulo p los posibles residuos son: $0, 1, 2, ..., p-1$. Por continuar con el ejemplo anterior, si $a = 29, b = 5$ y $p = 12, 29 \equiv 5 \ (mod\ 12)$ porque $mod(29, 12) = 5$ y $mod(5, 12) = 5$.

Propiedades 1. *La relación de congruencia módulo $p > 1$ verifica las siguientes propiedades:*

1. *Reflexiva: $a \equiv a \ (mod\ p)$, para todo $a \in \mathbb{Z}$.*
2. *Simétrica: $a \equiv b \ (mod\ p) \rightarrow b \equiv a \ (mod\ p)$.*
3. *Transitiva: $a \equiv b \ (mod\ p)$ y $b \equiv c \ (mod\ p) \rightarrow a \equiv c \ (mod\ p)$.*

Por verificarse estas tres propiedades, sabemos que la relación de congruencias es una *relación de equivalencia*.

4.1.1. El conjunto $\mathbb{Z}_p$

Cada clase del conjunto cociente de $\mathbb{Z}$ por la relación de congruencia módulo p está formada por todos los números enteros con el mismo residuo módulo p. Puesto que hay p posibles residuos, habrá p clases distintas, cada una de ellas asociada a un residuo r, $0 \leq r \leq p-1$, y que se representa por $[r_p]$, $\bar{r}_p$ ó $\bar{r}$ si no hay lugar a error. El conjunto de todas las clases se representa por $\mathbb{Z}_p$, es decir:

$$\mathbb{Z}_p = \{\overline{0}, \overline{1}, \overline{2}, ..., \overline{p-1}\} \text{ donde } \bar{r} = \{a \in \mathbb{Z} \mid a \equiv r(\text{mod } p)\} = \{np + r \mid n \in \mathbb{Z}\}$$

4.2. Adición, substracción y multiplicación

Sea $N \in \mathbb{Z}_{>0}$, la suma, la resta y la multiplicación en $\mathbb{Z}_N$ son muy sencillas, basta con realizar la operación usual en $\mathbb{Z}$ y quedarnos con el resto módulo N. Veamos algunos ejemplos.

En $\mathbb{Z}_{13}$ el producto $8 \times 11 = 10$ ya que $8 \times 11 = 88$ y $\frac{88}{13}$ tiene de resto 10.

Con la resta se sigue el mismo procedimiento excepto cuando el resultado es negativo, por ejemplo, $5 - 10 = 8$ porque $5 - 10 = -5$, pero tenemos que ver ese -5 a qué clase corresponde para poder estar dentro del cuerpo $\mathbb{Z}_{13}$, es decir, buscamos el número $13 - 5$ que es 8.

Esto ocurre porque no hay números negativos, todos los números en $\mathbb{Z}_N$ son negativos y positivos a la vez. Todo número en este sistema es siempre el negativo de otro. En $a + b = 0$, a es el negativo de b y b es el negativo de a. Por ejemplo, para efectuar $3 - 8$, en $\mathbb{Z}_{12}$, a 3 le sumamos el negativo de 8 que es 4. Por tanto, $3 - 8 = 3 + (-8) = 3 + 4 = 7$.

4.3. Residuos de operaciones aritméticas

Dado un número entero $p > 1$, si $a \equiv \alpha \ (mod\ p)$ y $b \equiv \beta \ (mod\ p)$, entonces:

$$(a + b) \equiv (\alpha + \beta) \ (mod\ p) \qquad ab \equiv \alpha\beta \ (mod\ p) \qquad a^b \equiv \alpha^b \ (mod\ p)$$

No es cierto, en general, que $a^b \equiv \alpha^\beta \ (mod\ p)$.

Simplificación de congruencias de productos

- Si $mcd(c, p) = 1$ entonces: $ac \equiv bc \ (mod\ p) \rightarrow a \equiv b \ (mod\ p)$
- En general: $ac \equiv bc \ (mod\ p) \rightarrow a \equiv b \ (mod\ \frac{p}{mcd(c,p)})$

4.4. División

Definición 3. *Se llaman divisores de cero a cualquier $\bar{a}, \bar{b} \in \mathbb{Z}_p$, con $\bar{a} \neq \bar{0} \neq \bar{b}$, tales que $\bar{a} \cdot \bar{b} = \bar{0}$.*

Teorema 1. *Existen divisores de cero en $\mathbb{Z}_p$ si y solo si p no es primo.*

Definición 4. *Se dice que $\bar{a} \in \mathbb{Z}_p$ es un elemento inversible o unidad si existe $\bar{b} \in \mathbb{Z}_p$ tal que $\bar{a} \cdot \bar{b} = \bar{1}$, y se indica $\bar{a}^{-1} = \bar{b}$ y a^{-1} para referirse a cualquier elemento de la clase.*

Ejemplo 1. *El inverso de $a = 4$ en $\mathbb{R}$ es $b = \frac{1}{4}$ porque $a \times b = 4 \times \frac{1}{4} = 1$. El inverso del entero 4 es el decimal $\frac{1}{4}$.*

Esto es una anomalía que no queremos que suceda en $\mathbb{Z}_p$. Queremos que en $\mathbb{Z}_N$ los inversos de los elementos de $\mathbb{Z}_N$ estén en $\mathbb{Z}_N$, como sucede con los números negativos, pero esto no siempre es así. Por ejemplo, en $\mathbb{Z}_9 = \{0, 1, ..., 8\}$, ningún elemento es inverso de 3, porque ningún número multiplicado por 3 dará 1 (3 es divisor de cero). Tendrá que dar 10 para que al hacerlo módulo 9 dé 1, y este entero no existe. Sin embargo, $2 \times 5 = 1$, así pues, el 5 es el inverso del 2 y al contrario. Podemos afirmar que en $\mathbb{Z}_9$, el 2 es inversible, es decir, tiene inverso y su inverso es $2^{-1} = 5$.

La división $\frac{a}{b}$ la entendemos como $a \times b^{-1}$, es decir, multiplicamos a por el inverso de b.

Los elementos de $\mathbb{Z}_9$ invertibles son $1, 2, 4, 5, 7$ y 8. El resto, $0, 3, 6$, no son invertibles, porque comparten factores con el módulo 9. Los elementos invertibles no comparten factores o divisores con el módulo. Podemos así afirmar el siguiente resultado:

Teorema 2. *Sea $N \in \mathbb{Z}_{>0}$ y $a \in \mathbb{Z}_N$. La condición necesaria y suficiente para que a tenga inverso en $\mathbb{Z}_N$ es que $mcd(a, N) = 1$.*

Además sabemos que existen $\phi(N)$ elementos invertibles en $\mathbb{Z}_N$, donde $\phi(N)$ es la función de Euler. ¿Cómo calcular el inverso? Por el Teorema de Bézout sabemos que si $mcd(a, b) = 1$ entonces existen enteros x, y tales que $ax + by = 1$, o haciendo $mod(b)$ tenemos que $ax \equiv 1 \ mod \ (b)$. Para poder calcular este inverso utilizamos basamos en el algoritmo de Euclides.

Destacar también que si aplicamos la función de Euler sobre un número p primo, resulta $\phi(p) = p - 1$. Además, si en este caso, escogemos dos números primos, p, q, y aplicamos la función de Euler sobre la multiplicación de ambos, resulta $\phi(p \cdot q) = (p - 1)(q - 1)$.

Definición 5. *Dado $n \in \mathbb{N}$, se define $\phi(n)$ como la cantidad de números naturales menores o iguales que n que son primos relativos con el propio n.*

Ejemplo 2. *$\phi(15) = 8$ ya que hay 8 números naturales menores que 15 que son primos relativos con 15:* $\{1, 2, 4, 7, 8, 11, 13, 14\}$

Ejemplo 3. *Veamos un ejemplo de las afirmaciones anteriores sobre la función de Euler:*
$\phi(7) = 1, 2, 3, 4, 5, 6 = 6$
$\phi(3 \cdot 5) = (3 - 1)(5 - 1) = 8 \rightarrow \phi(15) = 1, 2, 4, 7, 8, 11, 13, 14 = 8$

Si $mcd(a, b) = 1$, quiere decir que $ax = 1 \ mod \ (b)$ siendo x el inverso de a. Para poder calcular este inverso nos basamos en el algoritmo de Euclides.

4.5. Algoritmo de Euclides

Todos los algoritmos de clave púlica que veremos en la memoria involucran operaciones modulares y en particular el cálculo del inverso, lo que parece necesario depender de un algoritmo eficiente que lo calcule. En lo que resta de capítulo describiremos con detalle como funciona.

Definición 6. *Un algoritmo es una descripción explícita de cómo debe ser resuelto un problema computacional en particular.*

La eficiencia de un algoritmo puede ser medida en base a la cantidad de pasos elementales que se necesiten para resolver dicho problema.

El máximo común divisor de dos enteros puede obtenerse escogiendo el mayor de todos los divisores en común. Hay un proceso más eficiente que utiliza repetidamente el algoritmo de la división para hallar este cálculo. Este método es el algoritmo de Euclides.

El problema inicial es que queremos encontrar el máximo común divisor entre dos números enteros positivos a y b. En la escuela nos enseñaron a hallarlo mediante la descomposición en factores primos de dos números y tomábamos los factores comunes a ambos con el menor exponente con el que aparecían. El problema de este procedimiento es que si los números son muy grandes, o sus factores primos lo son, el cálculo resulta ser bastante complicado.

4.6. Idea del algoritmo

La idea que sigue el algoritmo de Euclides es la siguiente:

Para calcular el máximo común divisor entre dos números enteros positivos a y b, aunque podríamos tomar dos números negativos ya que trabajaremos con el módulo de estos, dividimos el más grande, sea a, entre el más pequeño, sea b. Esta división nos proporciona un cociente, c_1, y un resto, r_1. Si $r_1 = 0$, entonces $mcd(a, b) = b$. Si $r \neq 0$, dividimos en el dividendo el valor que toma el anterior cociente, c_1, y en el divisor, el resto anterior, r_1, obteniendo así otro cociente, c_2, y otro resto, r_2. Si $r_2 = 0$, entonces $mcd(a, b) = r_1$. Si $r \neq 0$, volvemos a realizar el mismo procedimiento.

De esta forma, el máximo común divisor entre a y b es el último resto distinto de cero que obtengamos con el procedimiento anterior.

4.7. Pseudocódigo

El código del algoritmo se basa en la siguiente estructura:

```
INPUT: a, b
    if a < b:
        aux = a
        a = b
        b = aux
    while b != 0:
        r = a mod b
        a = b
        b = r
    end
OUTPUT: return a
```

4.8. Aclaraciones

Si analizamos el algoritmo de Euclides, se ve claramente que necesitamos demostrar que, el máximo común divisor entre a y b es igual al máximo común divisor entre b y r_1. De este modo, esa igualdad se mantendrá durante todo el proceso y llegaremos a que el último resto distinto de cero, es el máximo común divisor de los dos enteros positivos iniciales.

Teorema 3. *El máximo común de dos números enteros positivos a y b con $a > b > 0$, coincide con el máximo común divisor de b y r, siendo r el resto que se obtiene al dividir a entre b.*

Demostración. Sea $d = mcd(a, b)$ y $t = mcd(b, r)$. Vamos a demostrar que $d = t$.

$\Longrightarrow$

Por definición de máximo común divisor, se tiene que d es un divisor tanto de a como de b. Por tanto, $a = a_1 d$ y $b = b_1 d$.

Por otro lado, por el algoritmo de la división se tiene que

$$a = bq + r, \text{ con } 0 \leq r < b$$

de donde se tiene que

$$r = a - bq = a_1 d - b_1 dq = (a_1 - b_1 q)d$$

Por tanto, d es un divisor de r. Como habíamos dicho antes, d también es un divisor de b, entonces, debe dividir a su máximo común divisor. Por tanto, d es un divisor de t.

$\Longleftarrow$

t es un divisor tanto de b como de r. Por ello se tiene que $b = pt$ y $r = st$. Si sustituimos estas dos igualdades en la ecuación del algoritmo de división tenemos que

$$a = ptq + st = (pq + s)t$$

Por lo tanto, t es un divisor de a. Como también lo era de b, debe ser un divisor de su máximo común divisor, es decir, t es un divisor de d.

Puesto que t es un divisor de d y d es un divisor de t, podemos afirmar que $t = d$. De este modo aseguramos que el algoritmo de Euclides funciona. □

4.9. Ejemplos de aplicación del algoritmo

Vamos a ver un par de ejemplos de aplicación del algoritmo de Euclides.

Vamos a calcular $mcd(721, 448)$. Para ello, dividimos el número mayor entre el menor; si el resto no es cero dividimos el divisor entre el resto sucesivamente hasta que el resto se haga cero. Así queda cada secuencia:

- $721 = 448 \cdot 1 + 273$
- $448 = 273 \cdot 1 + 175$
- $273 = 175 \cdot 1 + 98$
- $175 = 98 \cdot 1 + 77$
- $98 = 77 \cdot 1 + 21$
- $77 = 21 \cdot 3 + 14$
- $21 = 14 \cdot 1 + 7$
- $14 = 7 \cdot 2 + 0$

Se tiene que $mcd(721, 448) = 7$, el último resto que no es nulo.

Veamos otro ejemplo, $mcd(25134, 19185)$, siguiendo el mismo procedimiento:

- $25134 = 19185 \cdot 1 + 5949$
- $19185 = 5949 \cdot 3 + 1338$
- $5949 = 1338 \cdot 4 + 597$
- $1338 = 597 \cdot 2 + 144$
- $597 = 144 \cdot 4 + 21$
- $144 = 21 \cdot 6 + 18$
- $21 = 18 \cdot 1 + 3$
- $18 = 3 \cdot 6 + 0$

Se tiene que $mcd(25134, 19185) = 3$.

4.10. Algoritmo de Euclides extendido

En este algoritmo, además de encontrar el máximo común divisor de los números enteros a y b, como el algoritmo de Euclides hace, también encuentra los enteros x e y que satisfacen la identidad de Bézout, $ax + by = mcd(a, b)$.

El algoritmo de Euclides extendido es particularmente útil cuando a y b son primos entre sí, puesto que x es la inversa multiplicativa modular de un módulo a, e y es la inversa multiplicativa modular del módulo b. Esto tiene valor en un cálculo de la llave del algoritmo de cifrado de clave pública RSA.

Recordemos lema de Bézout:

Lema 1. *Sean $a, b \in \mathbb{Z}$, alguno distinto de cero. Entonces existen $n, m \in \mathbb{Z}$ tal que $an + bm = mcd(a, b)$.*

Veamos el algoritmo:

Dados $a \geq b > 0$,

1) tomar como valores iniciales

$$a_0 := a, \qquad a_1 := b, \qquad x_0 := 1, \qquad x_1 := 0, \qquad y_0 := 0, \qquad y_1 := 1$$

2) Para cada $i = 0, 1, \cdots$, iterar las siguientes asignaciones

$$\begin{aligned} a_i &:= q_{i+1}a_{i+1} + a_{i+2} \\ x_i &:= q_{i+1}x_{i+1} + x_{i+2} \\ y_i &:= q_{i+1}y_{i+1} + y_{i+2} \end{aligned}$$

hasta obtener un resto $a_i = 0$.

3) Si a_{n+1} es el primer resto nulo, entonces $d = a_n$, $x = x_n$, $y = y_n$.

4.11. Ejemplo del algoritmo de Euclides extendido

Vamos a calcular mediante el algoritmo de Euclides extendido $mcd(32, 12)$ junto con los elementos x e y de la ecuación del lema de Bézout.

$a = 32, b = 12$	$x_0 = 1, x_1 = 0$	$y_0 = 0, y_1 = 1$
$32 = 2 \cdot 12 + 8$	$1 = 2 \cdot 0 + 1$	$0 = 2 \cdot 1 - 2$
$12 = 1 \cdot 8 + 4$	$0 = 1 \cdot 1 - 1$	$1 = 1 \cdot (-2) + 3$
$8 = 2 \cdot 1 + 0$	$1 = 2 \cdot (-1) + 3$	$(-2) = 2 \cdot 3 - 8$

Por lo tanto, tenemos que

$$(-1) \cdot 32 + 3 \cdot 12 = 4 = mcd(32, 12)$$

Capítulo 5

Clave privada

El cifrado de clave privada, también conocido como *cifrado simétrico* o *cifrado de clave secreta* consiste en utilizar la misma clave para el cifrado y el descifrado.

El cifrado consiste en aplicar una operación, un algoritmo, a los datos que se desea cifrar utilizando la clave privada para hacerlos ininteligibles.

Las personas que comparten el sistema, comparten y guardan en secreto las dos funciones, es decir, guardan la llave para poder realizar el cifrado y descifrado del mensaje. Si se conoce la clave, cifrar y descifrar el mensaje es un cálculo fácil de realizar y el secreto quedará descubierto.

Vamos a mencionar algunos de los métodos criptográficos de clave privada utilizados en la historia.

5.1. Código de substitución

Antes de mencionar este método, debemos dejar clara la siguiente definición.

Definición 7. *Una función f, de un conjunto $\mathcal{A}$ a $\mathcal{B}$ es biyectiva si, para cada $y \in \mathcal{B}$ hay exactamente un $x \in \mathcal{A}$ que cumple que $f(x) = y$. Una función biyectiva es inyectiva y suprayectiva, creando una correspondencia "uno a uno" entre los elementos de los dos conjuntos.*

La idea de este método es substituir el mensaje perteneciente a un alfabeto concreto y convertirlo en un mensaje perteneciente a otro alfabeto distinto teniendo ambos alfabetos el mismo cardinal. Es decir, sean los alfabetos $\mathcal{A}$ y $\mathcal{B}$, se establece la biyección entre ambos, $\varphi : \mathcal{A} \to \mathcal{B}$, se sustituye cada letra del mensaje en claro por su imagen en φ.

Normalmente el alfabeto $\mathcal{B}$ es el propio alfabeto $\mathcal{A}$ permutado. Ambos comunicantes acuerdan la permutación definida para $\mathcal{A}$ que forma la llave de cifrado. Para poder obtener el descifrado se deberá realizar la permutación inversa sobre $\mathcal{B}$ para obtener el mensaje en claro escrito en el alfabeto $\mathcal{A}$.

Uno de los métodos más conocidos que siguen esta filosofía es el *Código César*. Consiste en desplazar cada letra del alfabeto con su orden habitual, un número de posiciones determinado k, este número de desplazamientos constituyen la clave. Identificando cada letra con el número n correspondiente a la posición que ocupa, el cifrado viene dado por la fórmula

$$c \equiv n + k(mod\ N)$$

donde la suma se realiza módulo N que es el cardinal del alfabeto.

Un ejemplo significativo de este método es la palabra HAL, nombre que se le atribuye a la computadora psicópata de la película '2001. Una Odisea del Espacio'. Esta palabra esconde el mensaje en claro IBM, cuyo significado es: apropiado para una computadora. En este criptograma la clave utilizada ha sido $k = 1$.

5.2. Análisis de frecuencias

En cualquier texto escrito, el orden en el que aparece cada letra, sin importar el idioma que se utilice, no es aleatorio, si no, que está sometido a las reglas de su correspondiente gramática. Cada grafía tiene una frecuencia distinta y existen tablas donde se registran estos estudios de los diferentes idiomas.

Esto es un impedimento para los cifrados de substitución, ya que, aunque se permute el alfabeto, seguirá apareciendo con la misma frecuencia las letras y serán fácilmente distinguibles las grafías más frecuentes y por tanto, será asequible el descifrado. Para evitar o dificultar este problema se han estudiado variantes del método de substitución como son:

- Polisubstituciones: la substitución se realiza entre varias letras formando un mismo bloque. La dificultad respecto al método de substitución es mayor, pero siguen habiendo estudios sobre la frecuencia de aparición de bloques de letras concretas.
- Códigos homófonos: se distingue principalmente porque el cardinal del alfabeto $\mathcal{B}$ es mayor que el cardinal del alfabeto $\mathcal{A}$. De esta manera, la aplicación que forman $\varphi : \mathcal{A} \to \mathcal{B}$ es "uno a varios", pero se considera que si $x, y \in \mathcal{A}, x \neq y$, sea $\varphi(x) \cap \varphi(y) = \emptyset$. De esta forma enmascaramos la frecuencia de las letras dificultando el ataque mediante análisis de frecuencias.

- Substituciones polialfabéticas: la imagen de cada letra depende de su posición dentro del mensaje a cifrar.

En este modelo destaca el Código de Augusto. En este método los comunicantes acuerdan una palabra o frase que será la clave utilizada para el cifrado del mensaje. Así pues, sea $k_1k_2\cdots k_n$ dicha clave y sea $M = m_1m_2\cdots m_n$ el mensaje a cifrar. El cifrado se obtiene si sumamos modularmente cada letra del mensaje con la correspondiente clave, como hacíamos en el Código de César.

Con este método seguimos teniendo el problema de la frecuencia de los caracteres, así que lo más óptimo sería crear este tipo de cifrado sobre un lenguaje sin redundancia. Este tipo de lenguajes se caracteriza por utilizar todas las combinaciones de letras de su alfabeto con la misma frecuencia, y además, estas combinaciones son todas válidas. Con este lenguaje sería imposible realizar análisis de frecuencias y sería inmune a cualquier tipo de criptoanálisis, porque al intentar realizar el descifrado del mensaje no habría un criterio para saber si el mensaje obtenido es correcto.

5.3. Códigos de transposición

En este método se consigue el cifrado mediante transposiciones, permutaciones, de las letras que contiene el mensaje en claro. Así, el mensaje se divide en bloques de tamaño predefinido n y cada bloque se cifra de forma individual, según la permutación que los comunicantes han elegido como llave. A este tipo de cifrado se le conoce como *cifrado en bloque.*

Definición 8. *Sea $I_n = \{1, ..., n\}$, una permutación de I_n es una aplicación biyectiva $\sigma : I_n \to I_n$. El conjunto $\mathcal{S}_n$ de todas las posibles $(n!)$ permutaciones de I_n es un grupo para la composición de aplicaciones, denominado grupo simétrico de orden n.*

Puesto que se conserva la frecuencia de las letras, seguimos teniendo el mismo problema de vulnerabilidad mediante ataques de análisis de frecuencias. Aunque, por el contrario, con este método, hemos conseguido destruir las estructuras gramaticales del lenguaje al reordenar las letras de cada bloque.

Veamos el siguiente ejemplo para entender este mecanismo.

Ejemplo 4. *Supongamos que elegimos $n = 5$, siendo n el tamaño de los bloques y la permutación tomada como clave será*

$$\sigma = \begin{bmatrix} 12345 \\ 34152 \end{bmatrix} \downarrow$$

Tomamos como mensaje en claro M = ATACARMAÑANA y lo dividimos según el tamaño de bloque que hemos elegido (ATACA), (RMAÑA), (NAXXX). Puesto que no hay suficientes letras para completar el último bloque, lo rellenamos con un carácter aleatorio. Si aplicamos la clave, es decir, la permutación σ, sobre los bloques obtenemos el mensaje cifrado de tal forma: (AAATC), (AARMÑ), (XXNAX), por tanto nuestro mensaje será: $C(M)$ = AAATCAARMÑXXNAX.

5.4. Códigos lineales

Este tipo de códigos están basados en el cifrado en bloque. Los mensajes son bloques con longitud n, $x = (x_1, x_2, ..., x_n)$ donde los x_i son letras de un alfabeto elegido. Si el cardinal del alfabeto es N, estas letras pueden identificarse con elementos de $(\mathbb{Z}/N\mathbb{Z})$, por tanto, el mensaje en claro antes descrito, $x = (x_1, x_2, ..., x_n)$, puede considerarse un elemento de $(\mathbb{Z}/N\mathbb{Z})^n$. Cabe destacar que si N es un número primo, entonces $(\mathbb{Z}/N\mathbb{Z})$ es un cuerpo, en caso contrario, será un anillo. La clave será una matriz A de tamaño $n \times n$ inversible, es decir, también existirá A^{-1}, quiere decir que su determinante verifica la condición $mcd(det(A), N) = 1$, como hemos explicado en el apéndice, $det(A)$ es una unidad del anillo $(\mathbb{Z}/N\mathbb{Z})$.

El cifrado del mensaje se consigue multiplicando el mensaje, el vector $x = (x_1, x_2, ..., x_n)$ por la clave, la matriz A, $c(x) = y = xA$. El mensaje cifrado será también un elemento de $(\mathbb{Z}/N\mathbb{Z})^n$.

Una variante del cifrado lineal es el cifrado afín. En este método, la llave viene dada por un par (A, b) donde A es una matriz inversible como la definida anteriormente y $b = (b_1, b_2, ..., b_n) \in (\mathbb{Z}/N\mathbb{Z})^n$. Un bloque del mensaje se cifra ahora mediante $c(x) = y = xA + b$. Para poder hallar el descifrado necesitamos una clave de tipo (A^{-1}, bA^{-1}), y el descifrado se realiza mediante la operación $x = yA^{-1} - bA^{-1}$.

Los códigos lineales son muy seguros porque la multiplicación matricial destruye o enmascara la estructura del lenguaje haciendo inviable el ataque mediante análisis de frecuencias. Además, el número de matrices inversibles, de llaves posibles, o de pares (A, b) es muy elevado si tenemos un n grande. Por el contrario, estos códigos son muy vulnerables a ataques de tipo *a texto claro conocido*.

Si disponemos de suficiente parejas (x, y) de texto claro $x = (x_1, x_2, ..., x_n)$ y $y = (y_1, y_2, ..., y_n)$ texto cifrado, la matriz clave A se puede obtener simplemente resolviendo un sistema de ecuaciones lineales. Es decir, es suficiente con conocer n pares $(x^1, y^1), (x^2, y^2), ..., (x^n, y^n)$, sabiendo que $x^1, x^2, ..., x^n$ son vectores linealmente independientes.

En el caso afín sería muy similar. Para poder obtener la clave (A, b) es suficiente con conocer $n + 1$ pares independientes, es decir, basta con conocer $(x^1, y^1), (x^2, y^2), ..., (x^n, y^n)$, (x^{n+1}, y^{n+1}).

5.5. DES y secuencias cifrantes

Como hemos visto anteriormente, los sistemas de cifrado por sustitución y transposición han resultado ser muy vulnerables, sin embargo, aplicar sucesivamente estos sistemas ha proporcionado un nuevo sistema suficientemente seguro. Así es como nace el sistema DES.

Este sistema combina sustituciones, transposiciones y una llave de 56 bits. Además, en la actualidad, sigue siendo un sistema seguro y de uso generalizado. Se basa en codificar bits aislados del mensaje o de la llave. Por esta razón, estos ataques pertenecen al dominio de la ingeniería inversa más que al criptosistema.

Los sistemas de secreto perfecto o incondicionalmente seguros implicaban como llave una sucesión indefinidamente larga de elementos del alfabeto elegidos al azar. Aunque, por razones teóricas y prácticas, resultan inviables estas sucesiones, siendo sustituidas por sucesiones pseudoaleatorias, que, sin ser aleatorias ya que se obtienen a partir de algoritmos matemáticos, presentan al criptoanalista rival una apariencia aleatoria.

Capítulo 6

Clave pública

En clave pública cada usuario i del sistema posee un par de llaves (c_i, d_i), la primera de las cuales es pública, es conocida por cualquier persona, y es necesaria para que cualquier usuario j que desee comunicarse con el usuario i le pueda enviar el mensaje M. Este mensaje se cifrará de la forma $C = c_i(M)$. Por el contrario, la clave d_i, considerada como la clave privada, es conocida solamente por el usuario i y es necesaria para poder recuperar (descifrar) el mensaje M enviado por j o de cualquier otro usuario. Así pues, el mensaje se descifrará de la siguiente forma: $M = d_i(C) = d_i(c_i(M))$.

Sin embargo, estos sistemas deben cumplir que el conocer la clave pública no permita calcular la clave privada. Los sistemas de cifrado ofrecen un abanico superior de posibilidades pudiendo emplearse para establecer comunicaciones seguras por canales inseguros, puesto que únicamente viaja por el canal la clave pública, o bien para llevar a cabo autenticaciones.

Los sistemas de clave pública requiere una base matemática mucho más fuerte, donde se necesitan conceptos de aritmética, algoritmia y teoría de la complejidad computacional.

6.1. Condiciones de un sistema de clave pública

El origen de la criptografía de clave pública se le atribuye a Whitfield Diffie y Martin Edward Hellman en el año 1976, los cuales consideraron unos principios teóricos que debían satisfacer los sistemas con esta propiedad. Estos principios son conocidos como *las condiciones de Diffie-Hellman* y son los siguientes:

1. El cálculo de las llaves, públicas y privadas, debe ser computacionalmente sencillo, es decir, dado por un algoritmo de complejidad polinómica.

2. El proceso de cifrado debe ser computacionalmente sencillo.

3. El proceso de descifrado, conociendo la llave secreta, debe ser también computacionalmente sencillo.

4. La obtención de la llave secreta, a partir de la pública, debe ser un problema computacionalmente imposible, es decir, dado por un algoritmo de complejidad exponencial.

5. La obtención del mensaje en claro, conociendo el mensaje cifrado y la llave pública, debe asimismo ser computacionalmente imposible.

Sin embargo, ambos autores afirman que, para asegurar que se cumplen las condiciones anteriores, es necesario lo que se denomina una *función trampa*.

Definición 9. *Una función $f : A \to B$ se denomina función de una vía si*

a) para todo elemento $x \in A$ es computacionalmente sencillo calcular $f(x)$.

b) dado $y \in Im(f)$, es computacionalmente imposible, en general, determinar un elemento $x \in A$ tal que $f(x) = y$.

Definición 10. *Una función trampa es una función de una vía, f, con la propiedad adicional de que existe una función inversa secreta, la trampa, que permite calcular eficientemente el inverso de f en cualquier punto.*

Es fácil multiplicar dos números primos distintos p, q y obtener el número $N = p \cdot q$. Sin embargo, el proceso inverso, es decir, dado N lo suficientemente grande encontrar sus factores primos p y q es mucho más difícil. En la dificultad de factorizar un número de gran magnitud reside la seguridad de algunos sistemas de clave pública. La seguridad es meramente computacional, dado que el tiempo y recursos que hay que invertir para poder deducir la clave privada a partir de la pública son demasiados, pero matemáticamente es un problema soluble.

6.2. Funcionamiento del sistema

En un sistema de cifrado con clave pública, los usuarios tienen una clave aleatoria que sólo ellos conocen, la clave privada. A partir de esta clave, automáticamente se deduce un algoritmo para hallar la clave pública. Los usuarios intercambian esta clave pública por medio de canales no seguros.

Cuando un usuario quiere enviarle un mensaje a otro usuario, sólo debe cifrar el mensaje que desea enviar utilizando la clave pública del receptor, que puede encontrar, por

ejemplo, en un servidor de claves como un directorio LDAP. El receptor podrá descifrar el mensaje mediante su clave privada que sólo él conoce.

Este sistema se basa en una función que es fácil de calcular en una dirección, llamada *función trapdoor de único sentido*, y que, matemáticamente, resulta muy difícil de invertir sin la clave privada, llamada *trapdoor*.

Veamos ilustrado esto en un ejemplo, un usuario crea de forma aleatoria una pequeña llave metálica, la clave privada, y luego produce una gran cantidad de candados, clave pública, que guarda en un casillero al que puede acceder cualquiera, el casillero será el canal no seguro. Para enviarle un documento, cada usuario puede usar un candado abierto, cerrar con este candado una carpeta que contiene el documento y enviar la carpeta al dueño de la clave pública, el dueño del candado. Sólo el dueño podrá abrir la carpeta con su clave privada.

6.3. Esquemas

Los criptosistemas de clave pública pueden ser esquemas basados en la factorización, como RSA, y esquemas basados en el logaritmo discreto, como Diffie-Hellman, ElGamal o Massey-Omura, y la firma digital.

6.3.1. RSA

Este criptosistema se basa en la dificultad de factorizar un número natural compuesto y la ventaja de la facilidad de la operación inversa de multiplicación. La dificultad computacional de este problema es muy alta, además, todos los algoritmos de factorización conocidos tienen una complejidad computacional exponencial.

Cada usuario i del sistema debe elegir una pareja de primos p_i, q_i, suficientemente grandes. Se calcula además $n_i = p_i q_i$ y $\phi(n_i)$, donde $\phi(n_i)$ es la función de Euler. A continuación, el usuario elige un número de forma arbitraria e_i, $0 < e_i < \phi(n_i)$, tal que $mcd(e_i, \phi(n_i)) = 1$ y su inverso modular $d_i \equiv e_i^{-1}\ (mod\ \phi(n_i))$.

Definición 11. *Dos números enteros a, b son primos relativos si $mcd(a, b) = 1$*

En el próximo capítulo lo explicaremos con detalle.

6.3.2. El logaritmo discreto

Sea G un grupo abeliano finito (multiplicativo) y sea g un elemento de orden n de G. Dado un elemento a perteneciente al subgrupo generado por g, se define el logaritmo discreto de a en base g como el entero k, $0 \leq k \leq n-1$, tal que:

$$g^k = a$$

Se dice también que k es el índice de a en base g.

El problema del logaritmo discreto consiste en, dados g y a, calcular k.

Aunque en este caso el logaritmo discreto se ha definido en un grupo multiplicativo, se puede definir de forma general en un grupo. Además, es posible definir el logaritmo discreto en grupos aditivos como el conjunto de puntos de un curva elíptica que hablaremos más tarde.

Ejemplo 5. *Sea* ${\mathbb{F}^*}_{2131}$ *el grupo multiplicativo de los enteros módulo 2131. Se tiene que* ${\mathbb{F}^*}_{2131} =< 37 >$. *Como* $1217 \equiv 37^5 (mod\ 2131)$, *el logaritmo discreto de* 1217 *en base* 37 *es* 5.

La importancia del estudio de este problema, radica en el interés del logaritmo discreto como operación inversa a la exponenciación en un grupo. La exponenciación modular es una operación sencilla y se conocen métodos eficientes para calcularla. En cambio, el logaritmo discreto módulo un entero cualquiera, no siempre puede realizarse de forma eficiente.

Diffie-Hellman

Se eligen y hacen públicos un cuerpo finito $\mathbb{F}_q$ y un elemento primitivo $g \in \mathbb{F}_q$. Supongamos que dos personas, Alicia(A) y Benito(B), quieren acordar una clave secreta en común. Entonces proceden de la siguiente manera:

1. A y B eligen dos enteros, a y b respectivamente, con la única condición de que $2 \leq a, b \leq q-2$.
2. A transmite g^a a B y B transmite g^b a A.
3. A calcula $(g^b)^a$ y B calcula $(g^a)^b$.

La clave común será entonces g^{ab}.

En la elección de a y b no se consideran los enteros 1 y $q-1$ ya que, en ambos casos, el algoritmo pierde toda su seguridad.

- Si $a = 1$ (o $b = 1$), entonces $g^a = g$ (o $g^b = g$). Por lo tanto, si se sabe que $g^a = g$ y $1 \leq a, b \leq q - 1$, es fácil deducir que $a = 1$ y que la clave compartida es g^b.

- Si $a = q - 1$ (o $b = q - 1$), entonces $g^a = 1$ (o $g^b = 1$). Por lo tanto, si se sabe que $g^a = 1$ y que $1 \leq a, b \leq q - 1$, es fácil deducir que $a = q - 1$ y hallar la clave compartida $g^{ab} = (g^b)^{q-1}$.

Ejemplo 6. *Alicia y Benito quieren establecer una clave común utilizando el método de intercambio de claves de Diffie-Hellman. Trabajan en un cuerpo* $\mathbb{F}^*_{23}$ *y toman 5 como elemento primitivo. Entonces Alicia escoge un entero* $a = 7$ *y Benito, otro* $b = 13$*. Alicia envía a Benito* $5^a \equiv 17 \pmod{23}$ *y él le envía a ella* $5^b \equiv 21 \pmod{23}$*. A continuación, Alicia calcula* 21^7 *y Benito* 17^{13}*. Ambos obtienen la clave común* $5^{7 \cdot 13} \equiv 10 \pmod{23}$*.*

ElGamal

Se conocen el cuerpo $\mathbb{F}_q$ y un elemento primitivo g del mismo.

Cierto usuario del sistema, A, elige un entero a tal que $2 \leq a \leq q - 2$ y calcula g^a. El entero a es su clave privada y g^a es la clave pública. Si otro usuario, B, quiere mandar un mensaje m a A ha de hacer lo siguiente:

1. Elegir un elemento k, $2 \leq k \leq q - 2$
2. Enviar el par (g^k, mg^{ak}) a A

En la elección de a y k, los enteros 1 y $q - 1$ se descartan por razones similares a las expuestas en el intercambio de claves de Diffie-Hellman.

A partir del par (g^k, mg^{ak}), es fácil para A obtener el mensaje original m de la siguiente manera:

1. Calcula $g^{ak} = (g^k)^a$
2. Halla $(mg^{ak})/g^{ak} = m$

El segundo paso para recuperar el mensaje puede ser sustituido por:

1. Calcula $(g^k)^{q-1-a}$
2. Halla $(g^k)^{q-1-a} mg^{ak} = mg^{k(q-1)-ka+ak} = m(g^{q-1})^k = m$

Ejemplo 7. *Alicia y Benito quieren intercambiar mensajes utilizando el criptosistema de ElGamal en el cuerpo* $\mathbb{F}_{157}$ *con generador* $g = 5$. *Para ello, Alicia escoge su clave privada* $a = 25$ *y comparte su clave pública* $g^a = 34$. *Supongamos que Benito quiere mandar el mensaje* $m = 19$ *a Alicia. Entonces elige un entero* $k = 89$ *y le envía el par* $(5^{89}, 19 \cdot 5^{25 \cdot 89}) = (131, 45)$. *Para obtener el mensaje original, Alicia halla* $5^{25 \cdot 89} \equiv 85$ *(mod* 157*) y calcula* $45/85 \equiv 19$ *(mod* 157*). Alicia también puede recuperar el mensaje a partir de* $5^{89(157-1-25)} \equiv 133$ *(mod* 157*) y calculando* $133 \cdot 45 \equiv 19$ *(mod* 157*).*

Massey-Omura

Este criptosistema se basa en el caso particular del protocolo de los tres pasos, el cual, se detalla a continuación.

Se utiliza la conmutatividad de ciertas funciones para conseguir, en tres pasos, que dos personas intercambien un mensaje de forma segura sin compartir ninguna clave. El proceso es el siguiente:

Paso 1: El emisor del mensaje m, A, elige una clave de cifrado e_A y su correspondiente clave de descifrado d_A y envía el mensaje cifrado $C(e_A, m)$ al receptor.

Paso 2: El receptor, B, elige una clave de cifrado e_B y su correspondiente clave de descifrado d_B. A continuación, cifra el mensaje que ha recibido $C(e_B, C(e_A, m))$ y se lo envía a A.

Paso 3: A descifra el mensaje recibido con su clave d_A y envía el resultado a B. Esto es, $D(d_A, C(e_B, C(e_A, m))) = C(e_B, m)$ porque la función de cifrado es conmutativa.

Finalmente, B obtiene el mensaje utilizando su clave de descifrado: $D(d_B, C(e_B, m)) = m$.

El criptosistema de Massey-Omura trabaja sobre el cuerpo $\mathbb{F}_q$.

Imaginemos que un emisor A quiere enviar un mensaje $m \in \mathbb{F}^*_q$ al receptor B. En primer lugar, A elige un entero c tal que $1 \leq c < q-1$ con c y $q-1$ primos entre sí y calcula c^{-1} $(mod\ q-1)$. B realiza el mismo proceso, es decir, escoge un entero d con las mismas características que c y calcula d^{-1} $(mod\ q-1)$. A continuación, A y B comienzan el siguiente intercambio de mensajes cifrados:

1. A calcula $x \equiv m^c$ $(mod\ q)$ y se lo transmite a B.
2. B calcula $y \equiv x^d \equiv (m^c)^d$ $(mod\ q)$ y se lo envía a A.
3. A calcula $z \equiv y^{c-1}$ $(mod\ q)$ y se lo transmite a B. (Nótese que $z \equiv m^{cdc^{-1}} \equiv m^d$ $(mod\ q)$

4. B finalmente calcula $z^{d-1} \equiv m \ (mod\ q)$).

Ejemplo 8. *Alicia quiere enviar a Benito el mensaje* $m = 13$ *en* $\mathbb{F}^*_{53}$ *utilizando el criptosistema de Massey-Omura. Para ello, Alicia elige* $c = 3$ *y calcular* $c^{-1} = 35$. *Benito escoge* $d = 7$ *y obtiene* $d^{-1} = 15$. *Entonces, comienza el intercambio de mensajes cifrados:*

1. *Alicia envía* $13^3 \equiv 24$ *(mod* 53*).*
2. *Benito calcula* $24^7 \equiv 36$ *(mod* 53*).*
3. *Alicia envía a Benito* $36^{35} \equiv 15$ *(mod* 53*).*
4. *Benito obtiene el mensaje* m *calculando* $15^{15} \equiv 13$ *(mod* 53*).*

6.3.3. El logaritmo discreto elíptico

Los criptosistemas de logaritmo discreto han sido estudiados sobre el cuerpo finito $\mathbb{F}_q$. Sin embargo, el mismo problema puede plantearse sobre cualquier grupo abeliano finito A. Aunque algunos expertos exigen que tal grupo debe cumplir las siguientes condiciones:

- El grupo ha de ser cíclico.
- Debe disponerse de un algoritmo eficiente para la multiplicación de sus elementos.
- El orden del grupo debe ser conocido.

Aunque el problema del logaritmo discreto se considere intratable, su dificultad puede variar según el grupo concreto. Para conseguir mayor seguridad en este algoritmo, se propuso como grupo candidato el grupo $E(\mathbb{F}_q)$, puntos de una curva elíptica sobre un cuerpo finito $\mathbb{F}_q$.

El grupo $E(\mathbb{F}_q)$ es, o bien cíclico, o producto de dos grupos cíclicos. El cálculo de su cardinal es siempre posible, mediante un algoritmo de complejidad polinomial. Además, para curvas particulares tal cardinal es conocido a priori o muy fácil de determinar.

Cabe destacar que la suma de puntos en una curva elíptica implica sólo la realización de un número pequeño de operaciones elementales en el cuerpo base $\mathbb{F}_q$.

El empleo del grupo de puntos de una curva elíptica, utilizada para los criptosistemas basados en el problema del logaritmo discreto, presentan las siguientes ventajas respecto al caso del grupo $\mathbb{F}_q$:

- Los ataques al problema del logaritmo discreto parecen más difíciles en el caso de $E(\mathbb{F}_q)$ que en el caso de un cuerpo finito de tamaño semejante. El empleo de curvas

elípticas permite utilizar grupos de tamaño menor y, por tanto, claves también menores, lo que simplifica las computaciones necesarias.

- Fijado el cuerpo $\mathbb{F}_q$ existen muchas curvas elípticas sobre él. Esto ofrece la ventaja de que, en un sistema con muchos usuarios, todos pueden compartir el mismo hardware y sin embargo, cada usuario puede seleccionar una curva diferente.

La adaptación de los criptosistemas clásicos, basados en el problema del logaritmo discreto sobre $\mathbb{F}_q$, al caso elíptico es inmediata. El único problema que se plantea es el de la identificación de los mensajes a cifrar con elementos del grupo $E(\mathbb{F}_q)$, es decir, con puntos de la curva (recordemos que con $\mathbb{F}_q$ identificábamos cada mensaje con un número menor que q y este con un elemento del cuerpo).

Es necesario disponer de un método para realizar la identificación de un mensaje m con un punto $P_m \in E$. Una condición exigible a tal identificación, es que la operación para recuperar el mensaje m a partir de P_m sea fácil de realizar. Habitualmente se utilizan métodos probabilísticos que permiten realizar la identificación con probabilidad de fallo arbitrariamente pequeña.

6.3.4. Firma digital

La firma digital es básicamente un conjunto de datos asociados a un mensaje que permiten asegurar la identidad del firmante y la integridad del mensaje. La firma digital debe tener las siguientes características:

- única, pudiendo generarla solamente el usuario logístico;
- no falsificable, el intento de falsificación debe llevar asociada la resolución de un problema numérico intratable;
- fácil de autenticar, esto es, cualquier receptor puede establecer su autenticidad;
- irrevocable, el autor de una firma no puede negar su autoría;
- fácil de generar.

La firma digital debe depender tanto del mensaje como del autor. Si esto no fuese así, el receptor podría modificar el mensaje y mantener la firma, produciendo así un fraude. Los criptosistemas de clave pública pueden ser fácilmente utilizados para generar firmas digitales.

Un usuario i con clave (c_i, d_i) procede de la siguiente manera para firmar sus mensajes. A cada mensaje $M \in \mathcal{M}$, le asocia la firma $s = d_i(M)$. Entonces, cualquier usuario puede

calcular $c_i(s)$ y verificar que coincide con M. Sin embargo, solo i puede deducir el valor de s para el que $c_i(s) = M$, esto es, solo i puede calcular la firma.

Con este algoritmo, si un usuario A quiere firmar un mensaje lo primero que debe hacer es establecer la clave pública (p, q, g, y) y la clave privada x. Para elegir dichas claves, ha de seguir las siguientes instrucciones:

1. Elegir un primo, p, de tamaño L, donde $512 \leq L \leq 1024$ y $64 \mid L$.
2. Escoger otro primo, q, de tamaño 160, tal que $p \equiv 1 \ (mod\ q)$.
3. Sea h un entero tal que $1 < h < p-1$ y $h^{(p-1)/q} \neq 1 \ (mod\ p)$. Tomar $g \equiv h^{(p-1)/q} \ (mod\ p)$. Las condiciones expuestas sobre h garantizan que g es un generador del único subgrupo cíclico de orden q de $\mathbb{F}^*{}_p$. Como $g^q \equiv h^{p-1} \equiv 1 \ (mod\ p)$, el orden de g es divisor de q, esto es, 1 ó q, pero no puede ser 1 ya que $g \neq 1$. Entonces el orden de g es q y, por lo tanto, genera el único subgrupo de orden q de $\mathbb{F}^*{}_p$.
4. Escoger x tal que $1 < x < q - 1$.
5. Calcular $y \equiv g^x \ (mod\ p)$.

El usuario A está ahora en disposición de firmar su mensaje M. Debe obtener un par de enteros (r, s) a través de los siguientes cálculos:

1. Elige un entero k verificando $0 < k < q$.
2. Obtener $r \equiv (g^k \ (mod\ p))(mod\ q)$
3. Calcular $s \equiv k^{-1}(H(m) + xr) \ (mod\ q)$, donde H es la función hash SHA-1.

Definición 12. *Una función hash es una aplicación $h : \sum^* \to \sum^n, n \in \mathbb{N}$ que transforma una cadena de longitud arbitraria en una de longitud fija.*

Si el receptor del mensaje firmado quisiera asegurarse de que este ha sido realmente enviado por A, debería realizar los siguientes cálculos:

1. $w = s^{-1} \ (mod\ q)$.
2. $u_1 = H(m)w \ (mod\ q)$.
3. $u_2 = rw \ (mod\ q)$.
4. Finalmente, verificar si $g^{u_1} y^{u_2} \equiv r \ (mod\ p)$.

Efectivamente, si A es el firmante, se tiene que:

- $g^q \equiv h^{p-1} \equiv 1 \ (mod \ p)$
- $k \equiv H(m)s^{-1} + xrs^{-1} \equiv H(m)w + xrw \ (mod \ q)$.

Por tanto,

$$g^k \equiv g^{H(m)w+xrw+zq} \equiv g^{H(m)w} g^{xrw} g^{zq} \equiv g^{u1} y^{u2} \ (mod \ p)$$

Capítulo 7

RSA

En 1977 Ronald Rivest, Adi Shamir y Leonard Adleman crearon el denominado sistema RSA. Este criptosistema, el primero de clave pública, uno de los más populares hoy en día por su uso en Internet, está basado en congruencias. Recordemos qué es esto.

7.1. El sistema

En el criptosistema RSA son de vital importancia los números primos ya que constituyen la pieza básica en la construcción de este.

Quienes deseen crear un juego de claves, pública y privada, en el criptosistema RSA primero seleccionan dos números primos p, q diferentes lo suficientemente grandes. Entonces calculan su producto $n = p \cdot q$. Después evalúan la función de Euler $\phi(n) = (p-1)(q-1)$, y seleccionan un número entero positivo e con $1 < e < \phi(n)$ tal que e sea coprimo con $\phi(n)$. Finalmente calculan el número entero d con $1 < d < \phi(n)$ tal que

$$d \cdot e \equiv 1 (mod\ \phi(n))$$

Tanto la verificación de que e es primo con $\phi(n)$ como la obtención de su inverso, se realizan gracias al algoritmo de Euclides extendido, algoritmo computacionalmente polinómico.

El usuario dispone ya de todos los elementos necesarios para sus claves:

Clave pública: (n_i, e_i)

Clave privada: d_i

Ejemplo 9. *Sean* $p = 103$ *y* $q = 199$ *números primos, con* $n = 20497$. *Se calcula* $\phi(n) = 102 \cdot 198 = 20196$, *y se elige* $e = 8207$, *este es coprimo con* $\phi(n)$. *Ya está completa la clave pública, ahora es necesario calcular* d, *tal que* $d \cdot e \equiv 1 \ (mod\ \phi(n))$. *Se obtiene* $d = 3455$ *con lo que se completa la clave privada.*

La trampa radica en que $\phi(n_i)$ es fácil de calcular conociendo la factorización de n (es decir, $\phi(n_i) = (p_i - 1)(q_i - 1)$), pero difícil si tal factorización no se conoce. Además, la clave privada d_i no puede conocerse a partir de e_i sin conocimiento de $\phi(n_i)$.

Los mensajes tanto en claro, como de descifrado, deben previamente identificarse con elementos del conjunto de clases residuales $\mathbb{Z}/n_i\mathbb{Z}$.

Cifrado: $\mathbb{Z}/n_i\mathbb{Z} \to \mathbb{Z}/n_i\mathbb{Z}; M \mapsto C \equiv M^{e_i} \ (mod\ n_i)$

a) Se obtiene la clave pública (n_i, e_i).

b) Representar el mensaje x como una sucesión de enteros $x_1, x_2, ..., x_t$ en el intervalo $[0, n_i - 1]$.

c) Calcular $c_i \equiv x_i^e \pmod{n_i}$, $i = 1, 2, ..., t$.

d) Enviar el texto cifrado $c = c_1, c_2, ..., c_t$.

Descifrado: $\mathbb{Z}/n_i\mathbb{Z} \to \mathbb{Z}/n_i\mathbb{Z}; C \to C^{d_i} \equiv M^{e_i d_i} \equiv M \ (mod\ n_i)$

a) Usar la clave privada d y calcular $x_i \equiv c_i^d \ (mod\ n)$ $1 \leq i \leq t$, para recuperar x.

Lema 2. *Con las notaciones anteriores, se verifica* $M^{e_i d_i} \equiv M \ (mod\ n_i)$.

Demostración. Distingamos dos casos:

Caso 1: $mcd(M, n_i) = 1$. Este es el caso en el que M puede considerarse como un elemento de $(\mathbb{Z}/n_i\mathbb{Z})^*$, grupo de orden $\phi(n_i)$. Dado que $e_i d_i \equiv 1 \ (mod\ \phi(n_i))$, el resultado es evidente.

Caso 2: $mcd(M, n_i) \neq 1$. Puesto que $n_i = p_i q_i, M$ debe ser divisible por uno de los primos p_i, q_i, pero no por ambos.

Supongamos que $p_i \mid M$ y $q_i \nmid M$ Obviamente se verifica $M^{e_i d_i} \equiv M \equiv 0 \ (mod\ p_i)$. También se tiene $M^{e_i d_i} \equiv M \ (mod\ q_i)$; es decir, M puede considerarse como un elemento de $(\mathbb{Z}/q_i\mathbb{Z})^*$, y siendo que $e_i d_i \equiv 1 \ (mod\ \phi(n_i)) = (p_i - 1)(q_i - 1)$, también $e_i d_i \equiv 1 \ (mod\ q_i - 1)$, de donde se sigue el resultado.

Si consideramos el caso contrario, llegamos a que $M^{e_i d_i} \equiv M \ (mod\ p_i)$.

Así pues, las dos congruencias, $M^{e_i d_i} \equiv M \ (mod\ p_i)$ y $M^{e_i d_i} \equiv M \ (mod\ q_i)$, implican $M^{e_i d_i} \equiv M \ (mod\ n_i)$. □

Los procesos de cifrado y descifrado se basan en una exponenciación modular, para la cual existe un algoritmo de complejidad polinómica.

7.2. Mensajes en claro y mensajes cifrados

Este criptosistema presenta el inconveniente de que el espacio de mensajes en claro $\mathcal{M}_i$, que como hemos dicho es el $\mathbb{Z}/n_i\mathbb{Z}$, es diferente para cada usuario i, situación indeseable por motivos prácticos. Lo mismo sucede para el espacio $\mathcal{C}_i$ de mensajes cifrados. Veamos cómo hacer iguales todos los $\mathcal{M}_i$ y análogamente todos los $\mathcal{C}_i$; por el contrario, ambos resultarán diferentes entre sí. Simultáneamente veremos cómo identificar los mensajes originales con elementos de $\mathbb{Z}/n_i\mathbb{Z}$.

Supongamos que el alfabeto de partida tiene cardinal N. Elijamos $k, l \in \mathbb{N}$, tales que $k < l$ y N^k, N^l sean de la magnitud requerida (aproximadamente 200 dígitos decimales). Se toman entonces:

$\mathcal{M}$ = Bloques de k letras $\sim$ Números con, a lo sumo, k dígitos en base N = Números naturales menores que N^k

$\mathcal{C}$ = Bloques de l letras $\sim$ Números con, a lo sumo, l dígitos en base N = Números naturales menores que N^l

En general, el mensaje a cifrar tendrá eventualmente más de k letras. Bastará entonces dividirlo en bloques de tamaño k (si el último de estos bloques queda incompletos, se completará insertando signos convenidos); a efectos del criptosistema cada bloque se considera un mensaje diferente.

Cada usuario debe elegir su pareja de primos p_i, q_i tales que $N^k < n_i = p_i q_i < N^l$. Ello permite, para todo mensaje $M \in \mathcal{M}$, identificarlo con un elemento de $\mathbb{Z}/n_i\mathbb{Z}$ (pues $N^k < n_i$); análogamente, como $C = c(M) \equiv M^{e_i} \ (mod\ n_i) \in \mathbb{Z}/n_i\mathbb{Z}$ y dado que $n_i < N^l$, puede considerarse que $C \in \mathcal{C}$.

Ejemplo 10. *En un alfabeto con $N = 26$ letras en su orden natural, identificamos A con 0, B con 1, ..., Z con 26. Elegimos $k = 5, l = 6$ y los números primos $p = 3851, q = 6607$. Vemos que*

$$11881376 = 26^5 < n = p \cdot q = 25443557 < 26^6 = 308915776$$

y que $\phi(n) = 3850 \cdot 6606 = 25433100$. Si se elige como clave pública $e = 8651341$, se tiene como clave privada $d \equiv e^{-1} \ (mod\ \phi(n)) = 4899061$.

El mensaje $M = VENDE$, se identifica con

$$21 \cdot 26^4 + 4 \cdot 26^3 + 13 \cdot 26^2 + 3 \cdot 26 + 4 = 9675670.$$

Su cifrado es

$$C = 9675670^{8651341} (mod\ 25443557) = 15989266 =$$
$$1 \cdot 26^5 + 8 \cdot 26^4 + 25 \cdot 26^3 + 18 \cdot 26^2 + 19 \cdot 26 + 20 \sim BIZSTU$$

y su descifrado

$$15989266^{4899061} \equiv 9675670 (mod\ n)$$

7.3. Seguridad del sistema RSA

La seguridad del sistema RSA radica en la imposibilidad computacional de factorizar un número de 200 cifras, ya que, con los algoritmos actuales y las mejores computadoras requeriría siglos. Cuando se señala que un problema es computacionalmente difícil, ello no excluye que instancias particulares del mismo sean fáciles. Para nuestro algoritmo, n producto de dos números primos, es necesario adoptar ciertas precauciones en la elección de dichos primos, pues en algunos casos los algoritmos de factorización existentes son muy eficientes. Hay que tener en cuenta que:

1. Los primos p y q no deben ser próximos entre sí, ya que, ambos serían próximos a $\sqrt{n}$, y mediante el algoritmo de factorización de Fermat no sería complicado descubrirlo. Así pues, tomar como p, q una pareja de primos gemelos, primos que se diferencian en dos unidades, sería la peor elección posible.

2. $p-1$ y $q-1$ no deben tener todos sus factores primos pequeños; de esta forma no se podrá aplicar la factorización mediante el método $p-1$ de Pollard.

3. $p+1$ y $q+1$ no deben tener todos sus factores primos pequeños; de esta forma no se podrá aplicar la factorización mediante el método $p+1$ de Pollard.

El intento de factorización de n no es el único ataque posible al RSA. En general, para cualquier criptosistema, el ataque del criptoanalista puede adoptar formas inesperadas. Vamos a describir una "debilidad potencial"de este criptosistema.

7.4. Debilidad potencial

Si el propósito de un sistema criptográfico es esconder el mensaje, es obvio que la situación en que el mensaje cifrado resulta ser igual al mensaje en claro es altamente indeseable. Sin embargo, para el RSA, esta situación se produce al menos para los mensajes $M = 0$ y $M = 1$ (y si la clave pública e es impar, también para el caso $M = -1$). Si estos valores de M fuesen los únicos, tal situación no sería preocupante ya que, en la práctica, el número total de mensajes es del orden de 10^{200}.

Consideremos el siguiente caso particular. Si

$$p = 7, q = 13, e = 13$$

absolutamente todos los mensajes permanecen inalterados (es decir, $M^{13} \equiv M \ (mod\, 7 \cdot 13)$ para todo M, $0 < M < 7 \cdot 13$), lo que implica que el hecho de cifrar un mensaje con este sistema sería absurdo. Necesitaríamos pues una fórmula para el número N de mensajes inalterados, que nos permita conocer a priori el riesgo de que esto ocurra. Tal fórmula viene dada por el siguiente resultado.

Proposición 1. *El número de mensajes, N, que permanecen inalterados al cifrarlos con el criptosistema RSA, definido por los números primos p, q y la llave pública e, es*

$$N = (1 + mcd(e-1, p-1))(1 + mcd(e-1, q-1))$$

Demostración. En virtud del teorema chino de los restos, el número de soluciones de la ecuación en congruencias $M^e \equiv M \ (mod\, pq)$, es el producto de cada una de las ecuaciones

$$M^e \equiv M \ (mod\ p),\ M^e \equiv M \ (mod\ q)$$

lo que a su vez equivale a

$$M^{e-1} \equiv 1 \ (mod\ p) \qquad \text{ó} \qquad M \equiv 0 \ (mod\ p)$$

$$M^{e-1} \equiv 1 \ (mod\ q) \qquad \text{ó} \qquad M \equiv 0 \ (mod\ q)$$

Dado que una congruencia del tipo $X^d \equiv 1 \ (mod\ p)$, tiene $mcd(d, p-1)$ soluciones, se deduce el resultado. □

A raíz de la proposición anterior, el sistema puede considerarse seguro si los valores $mcd(e-1, p-1)$, $mcd(e-1, q-1)$ son pequeños. En caso contrario, no se considera seguro.

Ejemplo 11. *Para $p = 5, q = 7$, la elección $e = 3$ conduce, según la fórmula de la proposición anterior, a $N = 9$. En cambio, para $e = 5$, se tiene que $N = 15$.*

Un criptoanálisis al criptosistema RSA se logrará si de alguna forma se consigue conocer cuál es el valor de $\phi(N)$

- Supongamos que el valor de $\phi(N)$ es conocido y que un intruso logra interceptar algún mensaje, del cual tiene conocimiento que fue cifrado con dicho criptosistema y conjunto de claves. Como los valores de N y e son públicos, el intruso sólo tendría que calcular el valor de d dado por $de \equiv 1 \ (mod \ \phi(N))$ para poder descifrar el texto interceptado al utilizar $x \equiv c^d \ (mod \ N)$ y así habrá tenido éxito al realizar un criptoanálisis al criptosistema.

- Por otro lado, supongamos que se logra la factorización de N, es decir $N = pq$. Como es conocido el valor de p y q se puede calcular el valor de $\phi(N)$ y así estaríamos en el caso anterior, donde se logró un criptoanálisis al suponer que se conocía el valor de $\phi(N)$.

El problema de calcular el valor de $\phi(N)$ es equivalente a factorizar N ya que, si se logra la factorización de N se podría calcular fácilmente el valor de $\phi(N)$. Por otro lado, si suponemos que se conoce el valor de $\phi(N)$, la factorización de N se lograría al resolver el siguiente sistema de ecuaciones que se forma:

$$\begin{aligned} N &= pq \\ \phi(N) &= (p-1)(q-1) \end{aligned}$$

De la primera ecuación del sistema se despeja $q = N/p$ y se sustituye en la segunda ecuación del sistema, con lo que se obtendría una ecuación cuadrática en términos de p

$$p^2 - (N - \phi(N) + 1)p + N = 0$$

Las raíces de esta ecuación son los valores de p y q mediante los cuales se logra la factorización de N.

Ejemplo 12. *Supongamos que $\phi(N) = 84754668$ es conocido y que el valor de $N = 84773093$ también lo es. Con esta información se logra obtener la ecuación:*

$$p^2 - 18426p + 84773093 = 0$$

resolviendo la ecuación se encuentran las dos raíces del sistema con $p = 9539$ y $q = 8887$ las cuales son los factores primos de N. Con lo que se habría logrado la factorización de N.

Este es solo un ejemplo de porqué los números que se utilizan en el criptosistema RSA son de gran magnitud. Por ejemplo, para la factorización del número conocido como RSA-768, que consta de 768 dígitos binarios, se emplearon 80 procesadores únicamente para seleccionar ciertos polinomios que se utilizarían en la factorización. La selección de los polinomios tardó aproximadamente medio año. Para el proceso de factorización se emplearon varios cientos de computadoras y la factorización requirió casi dos años, adicionales al medio año en que se seleccionaron los polinomios. Se estima que si solamente se empleara una computadora con procesador AMD a 2.2 GHz. con 2 GB de RAM se requerirían aproximadamente 1500 años para la factorización.

Mencionar la posibilidad de que se rompa este sistema si existieran ordenadores cuánticos, ya que estos están preparados para realizar la factorización de un número en tiempo real. Esto destruye todo criptosistema basado en factorización.

Los ordenadores cuánticos se basan en el uso de qubits en lugar de bits, y da lugar a nuevas puertas lógicas que hacen posibles nuevos algoritmos. La idea de computación cuántica surge en 1981, cuando Paul Benioff expuso su teoría para aprovechar las leyes cuánticas en el entorno de la computación. En lugar de trabajar a nivel de voltajes eléctricos, se trabaja a nivel de cuanto, valor mínimo que puede tomar una determinada magnitud en un sistema físico. En la computación digital tradicional, un bit solo puede tomar dos valores: 0 ó 1. En cambio, en la computación cuántica, intervienen las leyes de la mecánica cuántica, y la partícula puede estar en superposición coherente: puede ser 0, 1 y puede ser 0 y 1 a la vez. Eso permite que se puedan realizar varias operaciones a la vez, según el número de qubits. El número de qubits indica la cantidad de bits que pueden estar en superposición. Con los bits convencionales, si teníamos un registro de tres bits, había ocho valores posibles y el registro solo podía tomar uno de esos valores. En cambio, si tenemos un vector de tres qubits, la partícula puede tomar ocho valores distintos a la vez gracias a la superposición cuántica. Así, un vector de tres qubits permitiría un total de ocho operaciones paralelas. Así pues, el número de operaciones es exponencial con respecto al número de qubits.

Uno de los obstáculos principales para la computación cuántica es el problema de la decoherencia cuántica, que causa la pérdida del carácter unitario de los pasos del algoritmo cuántico. La decoherencia cuántica explica cómo un sistema físico, bajo ciertas condiciones específicas, deja de exhibir efectos cuánticos y pasa a exhibir un comportamiento típicamente clásico, sin los efectos contraintuitivos típicos de la mecánica cuántica. Otro de los problemas es la escalabilidad, especialmente teniendo en cuenta el considerable incremento en qubits necesario para cualquier cálculo que implica la corrección de errores. Para ninguno de los sistemas actualmente propuestos es trivial un diseño capaz de manejar un número lo bastante alto de qubits para resolver problemas computacionalmente interesantes hoy en día.

7.5. Elección de los primos p y q

Puesto que el sistema RSA se basa en la elección, por parte de cada usuario del sistema, de un par de números primos de tamaño adecuado, es obvio que el sistema solo sería factible en la práctica si tal elección es fácil, es decir, dada por algoritmos de complejidad polinomial en el tamaño de los datos.

Algoritmos polinómicos para decidir si un número es primo o compuesto existen y son conocidos como *tests de primalidad*. Se trata de test probabilísticos que no demuestran, en el sentido matemático, que un número es primo, pero garantizan dicha primalidad con probabilidad tan alta como se desee.

Para elegir los primos p y q, basta pues elegir a y b, números impares arbitrariamente del tamaño requerido, y someterlos a un test probabilístico. Si son primos puede tomarse, en principio, como los p, q buscados. Si a no fuese primo, se sustituiría por $a+2, a+4$, etc. (respectivamente con la b) hasta obtener los primos deseados.

Anexo A

Anexo I

A.1. Teoría de grupos

Los esquemas criptográficos presentados en este trabajo se construyen a partir de una de las estructuras algebraicas más estudiadas: la estructura de grupo. A continuación se presentan unas nociones elementales de teoría de grupos.

Definición 13. *Dado un conjunto G y una operación interna $\cdot : G \times G \to G$, se dice que el par $(G, \cdot)$ es un grupo si cumple las propiedades:*

- *Asociativa: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$, para cada $a, b, c \in G$.*
- *Existencia de elemento neutro: existe un elemento $e \in G$ que cumple que $a \cdot e = e \cdot a = a$ para cualquier $a \in G$, que llamaremos elemento neutro.*
- *Existencia de elemento simétrico: para cada $a \in G$ existe un elemento $a' \in G$ tal que $a \cdot a' = a' \cdot a = e$, que llamaremos elemento simétrico de a.*

Definición 14. *Sea el grupo $(G, \cdot)$ lo llamaremos grupo abeliano si, además de cumplir las propiedades de grupo, cumple la propiedad conmutativa. Es decir, para cada $x, y \in G$ se verifica que $x \cdot y = y \cdot x$.*

Definición 15. *Dado un grupo G se denomina orden de G, denotado por $|G|$, al cardinal del conjunto subyacente.*

Definición 16. *Sean $(G, \cdot)$ y $(H, *)$ grupos. Una aplicación $f : G \to H$ se dice que es un homomorfismo si $f(a \cdot b) = f(a) * f(b), \forall a, b \in G$.*

Definición 17. *Un homomorfismo inyectivo, respectivamente suprayectivo, se denomina monomorfismo, respectivamente epimorfismo.*

Definición 18. *Un homomorfismo biyectivo se denomina isomorfismo y, si se establece de un grupo en sí mismo, se denomina automorfismo.*

Teorema 4. *El conjunto de los automorfismos de un grupo G dado, tiene a su vez estructura de grupo con la operación composición de aplicaciones y se denota $Aut(G)$.*

Definición 19. *Sea G un grupo y $x \in G$, la aplicación $f_x : G \to G$ definida por $f_x(g) = xgx^{-1}$ es un automorfismo de G y se denomina automorfismo interno.*

Definición 20. *Sea G un grupo, un subconjunto no vacío $H \subset G$ se dice subgrupo de G si es estable respecto la ley interna de G y tiene a su vez estructura de grupo con la restricción de la operación de G.*

Definición 21. *Sea A un conjunto y sean $+$ y $\cdot$ dos operaciones binarias. Se dice que la terna $(A, +, \cdot)$ es un anillo conmutativo y con unidad si se cumplen las siguientes propiedades:*

a) $(A, +)$ *es un grupo abeliano.*

b) $(A, \cdot)$ *tiene las propiedades asociativa, conmutativa, tiene elemento neutro y es distributiva respecto a $+$.*

Ejemplo 13. *La terna $(\mathbb{Z}, +, \cdot)$ es un anillo conmutativo y con unidad, cuyo elemento neutro para la suma es el 0 y para el producto el 1.*

Definición 22. *Un cuerpo es un conjunto F provisto de dos operaciones internas, $(F, +, \cdot)$, de modo que tanto $(F, +)$ como $(F \setminus \{0\}, \cdot)$ son grupos abelianos, y el producto es distributivo respecto a la suma.*

Definición 23. *Un cuerpo finito es un cuerpo con un número finito de elementos. Se suele escribir $\mathbb{F}_q$ para indicar un cuerpo finito con q elementos (orden q).*

Teorema 5. *Un subconjunto no vacío H de G es un subgrupo si $\forall s, t \in H$ se tiene que $s^{-1} \in H$ y $st \in H$.*

Definición 24. *Un subgrupo invariante por los automorfismos internos de G, es decir, tal que $xHx^{-1} = H, \forall x \in G$, se denomina normal o invariante.*

Teorema 6. *Todo grupo G tiene al menos dos subgrupos normales, él mismo y el formado por el elemento neutro e_G. Además, estos dos subgrupos se dicen impropios y cualquier otro subgrupo se dice subgrupo propio de G.*

Definición 25. *Al menor subgrupo normal de G contiene a X se le denomina subgrupo normal generado por X y se escribe como $< X >^G$.*

Definición 26. *Si $X = \{g\}$, para algún $g \in G$, el grupo $< X >$ se denota por $< g >$ y se dice que es un grupo cíclico. Se denomina orden de g y se escribe $|g|$, al orden del grupo generado por g.*

A.2. Cuerpos finitos

Si p es primo, cada entero no divisible por p tiene inverso módulo p; por tanto, $\mathbb{Z}_p$ es un cuerpo. El cuerpo $\mathbb{Z}_p$ desempeña un papel fundamental en la teoría de cuerpos finitos.

Teorema 7. *Sea p un número primo y $m \in \mathbb{Z}_{>0}$ existen cuerpos de cardinal p^n y los denotamos como $\mathbb{F}_p$. Además, la extensión de cuerpos $\mathbb{F}_p \subset \mathbb{F}_{p^m}$ es algebraico de grado m.*

Definición 27. *Se denomina característica del cuerpo, al número primo p mencionado en el teorema anterior.*

Teorema 8. *Sea $\mathbb{F}$ un cuerpo de orden $q = p^m$, entonces cualquier subconjunto finito $\mathbb{F}^*$, es cíclico. En particular, $\mathbb{F}^*$ es cíclico de orden $q - 1$.*

Bibliografía

Anshel, I., Anshel, M., Goldfeld, D.: An algebraic method for public-key cryptography. Mathematical Research Letters, Vol. 6, 287-291, (1999).

Beckman, B.: Códigos secretos. Piramide, Madrid (1990).

Beth, T.: El estado de la Criptografía de Clave Pública a la vista de las posibilidades de la Computación Cuántica. Actas de la VI Reunión Española sobre Criptología y Seguridad de la Información, VI RECSI, 39-50, (2001).

Blackley, G. R., Borosh, I.: RSA public key cryptosystems do not always concel messages. Comput. Math. Appl., Vol. 5, 169-178, (1979).

Douglas, R., Stinson.: Cryptography: Theory and practice. Chapman and Hall/Crc, 3rd edition (2006).

Hoffman, K., Kunze, R.: Linear Algebra. Prentice-Hall, New Jersey (1971).

Mollin, R. A.: RSA and public-key cryptography. Chapman and Hall/CRC Boca Raton. Florida (2003).

Munuera, C., Tena, J.: Codificación de la información. Secretario de Publicaciones e Intercambio, Universidad de Valladolid, 205-240 (1997).

Navarro, G.: Un curso de álgebra, Universitat de València, (2002).

Peña Sénchez de Rivera, D.: Estadística. Modelos y métodos, Vol. 2. Alianza Editorial S. A., 308-313(1987 - 1989).

Rotman, J. J.: An Introduction to the Theory of Groups. Graduate Texts in Mathematics. Springer, Vol. 148, (1999).

Singh, S.: Los códigos secretos: El arte de la ciencia de la criptografía desde el antiguo Egipto a la era de Internet, Debate. Madrid (2000).

Vicent Francés, J. F., Propuesta y análisis de criptosistemas de clave pública

basados en matrices triangulares superiores por bloques. Tesis Doctoral.