Práctica 1 (PARTE 2) Cifrado / Firmas

Ejercicio 11

Cifrado XOR con clave temporal autogenerada

- Resolver el reto alojado en el puerto **11015** del sitio **ic.catedras.linti.unlp.edu.ar**

Sistemas de cifrado asimétricos (También llamados de clave pública)
En un algoritmo de cifrado asimétrico (por ejemplo, RSA), se utilizan 1 par de claves o llaves: la clave pública y la clave privada. Las claves públicas pueden darse a conocer libremente a otras personas, mientras que las claves privadas sólo deberían ser conocidas por su propietario.

Lo que se encripta con la clave pública, se desencripta con la clave privada correspondiente. Del mismo modo, lo que se encripta con la privada, se desencripta con la pública.

Si una persona que emite un mensaje a un destinatario, usa la clave pública del destinatario para cifrar el mensaje; sólo el destinatario, usando su clave privada, podrá descifrar dicho mensaje. El mensaje sólo podría ser descifrado por el destinatario dado que, es el único que debería conocer la clave que se necesita. Esto permite garantizar la confidencialidad del mensaje. Nadie salvo el destinatario puede descifrarlo. Cualquiera, usando la clave pública del destinatario, puede cifrar mensajes que solo podrán ser descifrados por el destinatario usando su clave privada.

Si el propietario del par de claves usa su clave privada para cifrar un mensaje, cualquiera puede descifrarlo utilizando la clave pública del primero. En este caso se consigue certificar el origen del mensaje, puesto que si todos lo podemos descifrar usando una clave pública, el origen es quien tiene la clave privada correspondiente. *Esta idea es el fundamento de lo que se conoce como firma digital.*

Los sistemas de cifrado asimétricos, permiten abordar el problema de la distribución de la clave, al no ser necesario un intercambio de claves como lo es en los sistemas de cifrado simétricos. La seguridad de los sistemas de cifrado asimétrico reside en la dificultad computacional de descubrir la clave privada a partir de la pública. La generación de un nuevo par de claves (una clave pública y su correspondiente clave privada) utilizan funciones matemáticas que son computacionalmente viable para la generación de ambas claves, pero las propiedades matemáticas de las mismas hace que sea muy difícil o computacionalmente imposible calcular la clave privada a partir de la clave pública.

RSA

Es un sistema criptográfico de clave pública desarrollado en 1977 por Rivest, Shamir y Adleman. Es el primer y más utilizado algoritmo de este tipo y es válido tanto para cifrar como para firmar mensajes digitales. La seguridad de este algoritmo radica en el problema de la factorización de números enteros muy grandes (propiedad matemática que hace que sea muy difícil calcular la clave privada a partir de la clave pública).

Los mensajes enviados se representan mediante números, y el funcionamiento se basa en el producto, conocido, de dos números primos grandes elegidos al azar y mantenidos en secreto. Actualmente estos primos son del orden de 10^200, y se prevé que su tamaño crezca con el aumento de la capacidad de cálculo de los ordenadores.

Como en todo sistema de clave pública, cada usuario posee dos claves de cifrado: una pública y otra privada. Cuando se quiere enviar un mensaje, el emisor busca la clave pública del receptor, cifra su mensaje con esa clave, y una vez que el mensaje cifrado llega al receptor, este se ocupa de descifrarlo usando su clave privada.

Explicación simplificada del uso de RSA

Proceso de generación de claves (pública y privada)

1. Escoger dos números primos distintos, en este ejemplo tomamos **p = 524287** y **q = 131071**
2. Calcular **n** = **p * q** = 524287 * 131071 = **68718821377**
3. Calcular **phi(n)** = (**p** - 1) * (**q** - 1) = 524286 * 131070 = **68718166020**
4. Elegir **e** tal que 1 < **e** < phi(n) y que sea coprimo con phi(n). Elegimos **e = 65537**
5. Los números **n y e** serán nuestra **clave pública**.
6. Se obtiene el exponente de clave privada **d**, calculando el inverso multiplicativo modular.
   d = modinv(e, phi(n))
   d = modinv(65537, 68718166020)
   d = 59397641569

7. Los números **n = 68718821377** y **d = 59397641569** serán nuestra **clave privada**.
   Definiciones
   coprimo = dos números son coprimos o primos entre sí cuando no tienen divisores en común.
   e = exponente de la clave pública
   d = exponente de la clave privada (se debe mantener en secreto junto con p y q)
   n = módulo de ambas claves, pública y privada
   (n,e) = clave pública
   (n,d) = clave privada
   modinv= inverso multiplicativo modular.

Un mensaje cifrado C se podrá obtener a partir de un mensaje M de la siguiente manera

1. M = hola
2. Representación hexa de M = 686f6c61
3. Representación decimal de M = 1752132705

4. C = (M ^ e) mod n
5. C = (1752132705 ^ 65537) mod 68718821377
6. C = 21543768249

M = mensaje en texto plano en este caso el mensaje que elegimos es “hola”
C = mensaje cifrado en este caso dió “21543768249”

Un mensaje cifrado C podrá ser descifrado utilizando el siguiente procedimiento:

1. M = (C ^ d) mod n
2. M = (21543768249 ^ 56692460753) mod 68718821377
3. M = 1752132705

M decimal = 1752132705
M hexadecimal = 686f6c61 (representación hexa de los bytes de M)
M ascii = \x68 \x6f \x6c \x61 = hola

Ejercicio 12

Revele el mensaje cifrado con RSA:

```
p:14116810449622477004714246307083749256487585440938819 77
q:10254777647391161702320017559629265694898389491212327 67
e:65537
C:24480032935390633635038225308868097264670696263978384433594823408502234840076325655977009553817777036504707 5
```

Ejercicio 13

- Resolver el reto alojado en el puerto **11012** del sitio **ic.catedras.linti.unlp.edu.ar**

Ejercicio 14

Revele el mensaje cifrado con RSA, esta vez no tenemos P ni Q.
Pista: Hay que factorizar o encontrar un buen lugar donde lo hagan…

```
n: 14524491846245356357574490859882044874942222485094938992997 59
e: 65537
C: 12807439447128571430606279699385388519111719501259799450261 52
```

Ejercicio 15

Desencriptar RSA sin p ni q

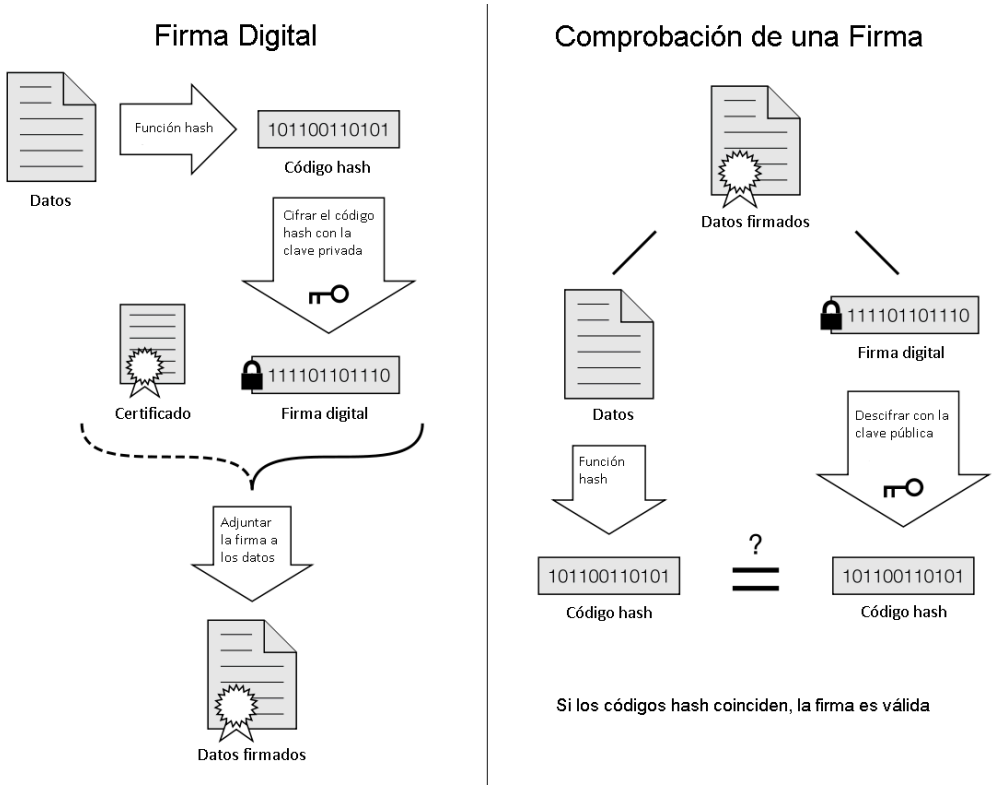
- Resolver el reto alojado en el puerto **11017** del sitio **ic.catedras.linti.unlp.edu.ar**

Firma digital

Utilizando algoritmos asimétricos (clave pública + clave privada) y funciones de hash, es posible crear el concepto conocido como FIRMA DIGITAL.
Una firma digital es un mecanismo criptográfico que permite al receptor de un mensaje firmado digitalmente certificar que dicho mensaje fué enviado por quien lo firmó (autenticación de origen y no repudio), y confirmar que el mensaje no ha sido alterado desde que fue firmado por el originador (integridad).

La firma digital es el cifrado del HASH de un mensaje usando la clave privada. Esto significa que el que envía el mensaje, certifica realmente que fue el, cifrando con su privada el hash del mensaje. Si cifrara todo el archivo, técnicamente sería igual de válido pero computacionalmente no seria muy optimo/performante.

Firma del mensaje(M) = CIFRADO_usando_privada(HASH(M))



Intercambio de claves seguro

Como se vió en la primera parte, la cuestión más vulnerable que tienen los algoritmos de cifrado de clave privada o simétricos es el intercambio de claves.

El mecanismo de intercambio de claves Diffie-Hellman provee una solución a este problema. Este método permite a dos partes, que no tienen conocimiento previo una de otra, establecer un secreto compartido por un canal no seguro.

Explicación simplificada del uso de Diffie-Hellman

Sean Alicia y Bob, dos partes que quieren acordar un secreto en común.

1. Alice y Bob seleccionan un primo **p** grande y un generador **g**, que es un entero más pequeño, ambos valores pueden ser públicos.
2. Alice elige un número entero secreto aleatorio, **a**. Este valor es su clave privada y no la comparte con nadie.
3. Bob elige otro número entero secreto aleatorio, **b**. Este valor es su clave privada y tampoco la comparte con nadie.
4. Alice calcula su clave pública: $A = g^a \pmod p$. Luego, le envía este valor A a Bob.
5. Bob calcula su clave pública: $B = g^b \pmod p$. Luego, le envía este valor B a Alice.

6. Alice recibe la clave pública de Bob, B. Ella calcula el secreto compartido como: $S=B^a \pmod p = (g^b)^a \pmod p = g^{ba} \pmod p$.
7. Bob recibe la clave pública de Alicia, A. Él calcula el secreto compartido como: $S=A^b \pmod p = (g^a)^b \pmod p = g^{ab} \pmod p$.

Ejemplo

Alicia y Bob van a acordar una clave privada.

1. Definen **p = 1999**, y **g = 33**.
2. A elige **a = 47**, calcula $\mathbf{33^{47} \bmod 1999 = 1343}$, y se lo envía a B
3. B elige **b = 117**, calcula $\mathbf{33^{117} \bmod 1999 = 1991}$, y se lo envía a A
4. B recibe **1343** y calcula $\mathbf{1343^{117} \bmod 1999 = 1506}$
5. A recibe **1991** y calcula $\mathbf{1991^{47} \bmod 1999 = 1506}$
6. La clave secreta compartida por A y B será **K = 1506**

Seguridad

La seguridad del intercambio de clave de Diffie-Hellman radica en la imposibilidad computacional a la que se enfrentará el atacante al tener que resolver el problema del logaritmo discreto para encontrar la clave privada que se encuentra en el exponente de la expresión $g^i \bmod p = C$.

Como p y g serán públicos, al capturar el valor C el atacante deberá resolver $i = \log_g C \bmod p$, un problema no polinomial (debido a la operación final dentro del módulo p) que para valores grandes de p (del orden o superior a los 1.000 bits) resulta computacionalmente imposible encontrar su solución.

Ejercicio 16

Diffie-Hellman

- Resolver el reto alojado en el puerto **11018** del sitio **ic.catedras.linti.unlp.edu.ar**

Sistemas de cifrado por bloques

Un sistema de cifrado por bloques es un algoritmo que permite cifrar de a bloques de tamaño fijo. Provee una función E para cifrar que convierte bloques de texto en claro P en bloques de texto cifrado C, utilizando una clave k: **C = E(k,P)**

Los bloques de texto en claro y texto cifrado son secuencias de bits y son del mismo tamaño.

El tamaño de los bloques es determinado por el sistema de cifrado.

Una vez que hemos cifrado los bloques, podemos realizar el proceso inverso usando una función de descifrado D, que toma el bloque de texto cifrado C y la clave k (la misma usada para cifrar ese bloque) como entradas y produce el texto claro original como salida.

P = D(k,C)

Como se puede notar, se usa la misma clave secreta para cifrar y descifrar por lo que los sistemas de cifrado por bloque son cifrados simétricos.

Ejemplo: DES, 3DES, AES

Sistemas de cifrado de flujo (stream cipher)

Para algunas aplicaciones, tales como el cifrado de conversaciones telefónicas, el cifrado en bloques es inapropiado porque los flujos de datos se producen en tiempo real en pequeños fragmentos. En este tipo de sistemas, el flujo de datos se va combinando con un flujo de clave pseudoaleatorio. Las muestras de datos que se van cifrando y descifrando a medida que se transmiten y reciben, pueden ser tan pequeñas como de 8 bits o incluso de 1 bit.

Para ello se puede utilizar un cifrado de flujo, el cual es un tipo de cifrado simétrico donde el flujo de texto en claro es combinado con un flujo de clave pseudoaleatorio (keystream). En este cifrado cada dígito del texto en claro es combinado con el correspondiente dígito del flujo de clave, para dar como resultado un dígito del flujo de texto cifrado.

En la práctica, un dígito es típicamente un bit, y la operación que se utiliza para combinarlos es XOR.

El flujo de clave pseudoaleatorio generalmente es la salida de una función matemática que utiliza una semilla (seed) como entrada.

El valor de la semilla sirve como clave para luego descifrar el flujo de texto cifrado.

Ejemplo: RC4, SEAL, A5

Esteganografía

La esteganografía trata el estudio y aplicación de técnicas que permiten ocultar mensajes u objetos, dentro de otros, llamados portadores, de modo que no se perciba su existencia. Es decir, se procura ocultar mensajes dentro de otros objetos y de esta forma establecer un canal encubierto de comunicación, de modo que el propio acto de la comunicación pase inadvertido para observadores que tienen acceso a ese canal.

Esteganografía en imágenes usando el bit menos significativo

Este es el método moderno más común y popular usado para esteganografía. Consiste en hacer uso del bit menos significativo de los píxeles de una imagen para transportar un mensaje. Hecho así, la distorsión de la imagen en general se mantiene al mínimo (la perceptibilidad es prácticamente nula), mientras que el mensaje es esparcido a lo largo de sus píxeles. Esta técnica funciona mejor cuando el archivo de imagen es grande. En general, los mejores resultados se obtienen en imágenes con formato de color RGB (utilizan tres bytes para la composición de los colores de cada pixel (Red,Green,Blue)).

Ejercicio 17

Utilice la herramienta steghide para encontrar el mensaje oculto en la imagen.

Ejercicio 18

Encuentre el mensaje oculto en el archivo de audio.

PGP (Pretty Good Privacy) es un programa desarrollado por Phil Zimmermann y cuya finalidad es proteger la información distribuida a través de Internet mediante el uso de criptografía de clave pública, así como facilitar la autenticación de documentos mediante el uso de firmas digitales.

Es un criptosistema híbrido que combina técnicas de criptografía simétrica y criptografía asimétrica. Esta combinación permite aprovechar lo mejor de cada uno: El cifrado simétrico es más rápido que el asimétrico o de clave pública, mientras que este, a su vez, proporciona una solución al problema de la distribución de claves en forma segura y garantiza el no repudio de los datos y la no suplantación.

¿Para qué lo podríamos utilizar nosotros?
Podríamos generar un par de claves PGP para proveer a nuestro cliente de correo la posibilidad de enviar y recibir mensajes:

- Cifrados y firmados
- Cifrados solamente
- Firmados solamente

También lo podríamos utilizar para cifrar información personal como pueden ser archivos, carpetas o incluso particiones enteras de disco. Para esto, PGP nos permite utilizar tanto criptografía simétrica (utilizando algoritmos como AES) o criptografía asimétrica (utilizando la clave pública del destinatario).

Red de confianza en PGP
Tanto cuando ciframos mensajes como cuando verificamos firmas digitales, es crucial que la clave pública enviada a alguien o alguna entidad realmente 'pertenezca' al destinatario intencionado. Existen repositorios públicos donde podemos subir nuestra clave pública PGP.

El hecho de descargar una clave pública PGP de algún sitio no nos asegura que podamos confiar en dicha clave.

Para utilizar PGP adecuadamente necesitamos poder asegurarnos que la clave pública de un usuario es efectivamente la que dicho usuario dice ser. Para ello, luego de tener la certeza de que dicha clave pública es la de dicho usuario, deberíamos firmar utilizando nuestra clave privada. De esta forma, nosotros certificamos ese documento que se corresponde con la clave PGP de otro usuario.

Ejercicio 19

> Utilice el diccionario indicado en el desafío publicado en la plataforma CTFd para crackear un archivo encriptado utilizando PGP con criptografía simétrica. Ver challenge **ej20 - symmetric pgp**

Ejercicio 20

> **Encriptar asimétrico:** Encriptar con PGP el archivo "encriptar.txt" con la clave pública dada, realizar el submit del archivo encriptado en formato armor.