Guía de Ejercicios - 8

Seguridad en Redes de Computadoras

1. Responder las siguientes preguntas:

    a. Explicar los conceptos de **confidencialidad**, **autenticación**, **integridad y disponibilidad.**
    b. ¿Cuál es la diferencia entre un algoritmo de encriptación simétrico y asimétrico? ¿Cuáles son las ventajas y desventajas entre ambas formas de encriptar?
    c. ¿Qué conceptos del ejercicio a. pueden proveer los dos tipos de encriptación del ejercicio b.?
    d. ¿Qué es una **firma digital**? ¿Con qué conceptos del ejercicio a. podemos relacionar las firmas digitales?
    e. ¿Qué es un **firewall**? ¿Qué servicios del ejercicio a. puede proveer un firewall?

2. Supongamos que un intruso tiene un mensaje encriptado y también la versión no encriptada del mismo. ¿Podría el intruso realizar un *ataque de texto cifrado*? ¿Podría realizar un *ataque de texto plano conocido*? ¿Y un *ataque de texto plano elegido*?

3. Dados los alfabetos:

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
A1	i	p	e	x	o	g	h	m	n	l	y	s	q	c	z	v	w	f	t	j	r	d	b	u	k	a
A2	t	q	f	d	i	c	v	z	b	p	o	l	r	s	x	a	h	e	m	n	u	w	g	y	j	k

Escriba un programa para encriptar el mensaje:

'*Alan Turing fue un matemático, lógico, informático teórico, criptógrafo, filósofo y biólogo teórico británico. Está considerado uno de los padres de la ciencia de la computación y precursor de la informática moderna.*'

Utilice la técnica de encriptación polialfabética siguiendo la secuencia:

A1 -> A2 -> A1 …

¿Qué problemas tiene este método de encriptación? ¿Cómo se podría romper el cifrado?

4. Si tenemos un *cifrado por bloques* de 8 bits ¿Cuántos posibles bloques de input (es decir, texto plano que debe ser cifrado) podríamos tener? Si el cifrado que es utilizado *mapea* cada bloque a un bloque distinto, ¿cuántos posibles *mapeos* diferentes podrían haber?

5. Considere el siguiente escenario, un usuario utiliza la página web de un banco para acceder a sus datos. Para ello realiza un request HTTP, que tiene la siguiente forma:

   'GET / HTTP1.1 …'

   El banco le responde a través de HTTP con el archivo html correspondiente, es decir un mensaje:

   'HTTP/1.1 200 …'

   a) ¿Sería buena idea utilizar un cifrado por bloques de 8 bits? ¿Qué problemas podrían surgir?
   b) ¿Se solucionarían aumentando la cantidad de bits del cifrado? ¿Qué desventajas traería esto?

6. Supongamos que tenemos un sistema de N personas, en el que cada una se quiere comunicar con las otras N-1 personas usando *encriptación simétrica*, y cada mensaje enviado desde una persona a otra es recibido también por todas las demás personas. ¿Cuántas claves son necesarias para lograr que ninguna persona pueda desencriptar un mensaje que es para otra persona? Si para la comunicación se utilizara un sistema de clave pública, ¿cuantas claves serían necesarias?

7. Carlos le quiere mandar el mensaje '*Hola soy Carlos'* a Paula utilizando el método RSA. Explique paso a paso qué debe hacer Carlos para lograrlo. ¿Qué información debe conocer de antemano? ¿Cómo hace Paula para desencriptar el mensaje?

a. Suponga que para calcular *m* simplemente se elige el número correspondiente de cada letra en el alfabeto, por ejemplo el mensaje 'abc' sería *m* =123 ¿Generaría esta técnica que RSA sea más vulnerable a ataques estadísticos?

8. ¿Qué problemas surgen de las comunicaciones que utilizan esquemas de encriptación de clave pública/clave privada?, ¿A qué clase de ataques son susceptibles? ¿Cómo sería llevado a cabo un ataque de ese estilo?

9. Suponga que Bob quiere establecer una conexión TCP con Alice. Sin embargo Trudy se está haciendo pasar por ella.
   a) ¿En qué momento del handshake TLS Bob se da cuenta qué no está hablando con Alice? ¿De qué manera se da cuenta?
   b) Una vez que descubre que Trudy se estaba haciendo pasar por Alice, cierra esa conexión y abre una con la verdadera Alice. Explicar cómo se utilizan las distintas contraseñas intercambiadas.

10. A partir de la siguiente tabla de firewall responder las siguientes preguntas:

Action	ip src	ip dst	proto	puerto src	puerto dst	flags
allow	123.10/16	fuera de 123.10/16	TCP	>1023	80	--
allow	Fuera de 123.10/16	123.10/16	TCP	80	>1023	ACK
allow	123.10/16	fuera de 123.10/16	UDP	>1023	53	*
allow	Fuera de 123.10/16	123.10/16	UDP	53	>1023	*
allow	123.10/16	fuera de 123.10/16	TCP	>1023	25	--
allow	Fuera de 123.10/16	123.10/16	TCP	25	>1023	ACK
deny	*	*	*	*	*	*

   a) ¿Qué servicios está dejando pasar este firewall?
   b) Indicar qué haría la tabla de firewall con los siguientes paquetes

1)

- Ip Source: 123.10.121.32
- Ip dst: 49.23.120.9.
- Prot: TCP
- Puerto src: 2421
- Puerto dest: 80

- Flags: ACK

2)
- Ip Source: 123.11.25.10
- Ip dst: 100.2.0.88.
- Prot: UDP
- Puerto src: 2421
- Puerto dest: 53
- Flags: --

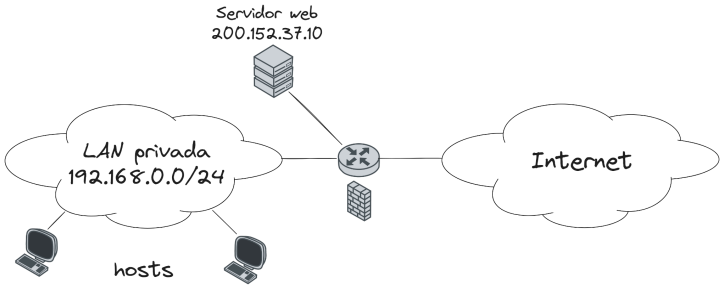
3)
- Ip Source: 71.1.2.1
- Ip dst: 123.10.200.9
- Prot: UDP
- Puerto src: 25
- Puerto dest: 2421
- Flags: --

4)
- Ip Source: 71.1.2.1
- Ip dst: 123.10.200.9
- Prot: TCP
- Puerto src: 53
- Puerto dest: 8888
- Flags: –

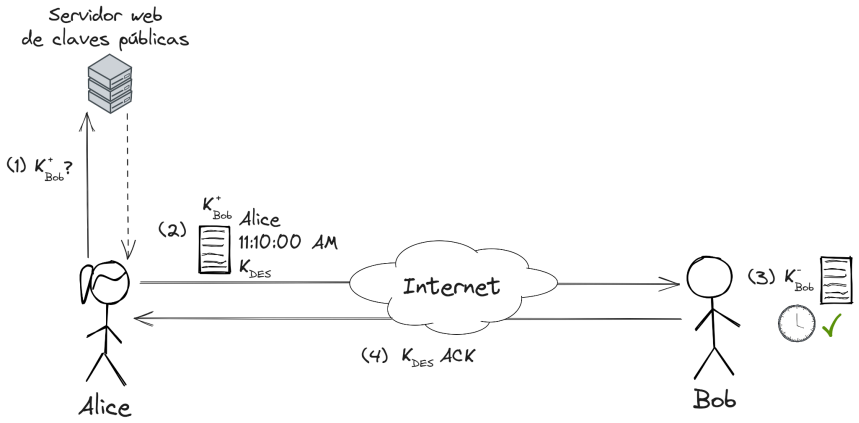
11. Cierta organización decidió estructurar una LAN para su negocio de acuerdo a la figura mostrada a la derecha. Las computadoras de los empleados se conectan a una subnet IP privada, mientras que un servidor web seguro alojando un portal de acceso público dispone de una dirección IP pública específicamente asignada por el ISP. El gateway que ofrece salida a Internet implementa además un firewall stateful que regula el tráfico entrante y saliente hacia la LAN y hacia el servidor web. Por directivas de la empresa, los empleados sólo deben tener permitido acceder a servicios web seguros desde sus

estaciones de trabajo. Por otro lado, por razones de seguridad, no se desea permitir conexiones entrantes desde el exterior a la LAN privada.

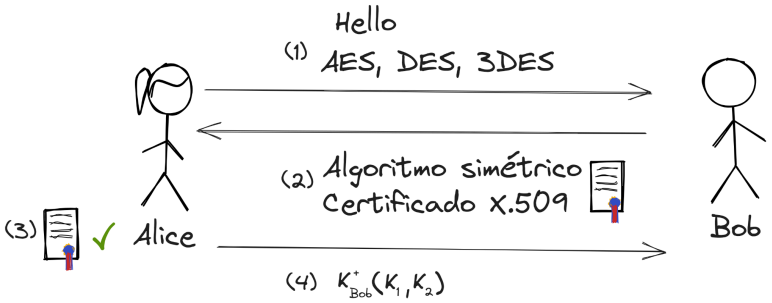
a. Proponer una configuración de reglas para el firewall que permita satisfacer estos requisitos.
b. El administrador de la red de la organización planea instalar un sniffer de tráfico en el gateway con el objetivo de definir reglas de firewall adicionales a partir del contenido de los requests provenientes de Internet hacia el servidor web interno. Analizar si esto es posible, indicando cómo implementarlo en caso de que sí lo sea o argumentando las razones en caso contrario.



12. Consideremos el siguiente protocolo diseñado por Alice para entablar comunicaciones seguras con Bob a través de Internet. En primer lugar, Alice envía un request HTTP a un servidor web remoto que oficia de repositorio global de claves públicas de RSA. El servidor web le transmite la clave pública de Bob, K^{+}_{Bob}. Acto seguido, Alice ensambla un mensaje compuesto por su identidad, la hora actual y una secuencia de bits aleatoria con el objeto de emplearlos como clave de sesión para la comunicación segura de datos. Alice encripta este mensaje con K^{+}_{Bob} y transmite el resultado a Bob, quien procede a desencriptar el mensaje recibido con su clave privada y validar posteriormente que la hora de la solicitud esté dentro de una ventana de tolerancia de un minuto. Por último, en caso de satisfacer dicha validación, Bob encripta un mensaje de confirmación con el algoritmo simétrico DES utilizando la clave de sesión provista por Alice y lo transmite a su interlocutora. Una vez finalizado este *handshake*, ambas partes están en condiciones de enviar datos de aplicación cifrados vía DES (en modo *counter*) con la misma clave de sesión. Identificar al menos tres problemas en el protocolo de Alice e indicar cómo podrían ser explotados por un atacante para vulnerar la seguridad de la comunicación.

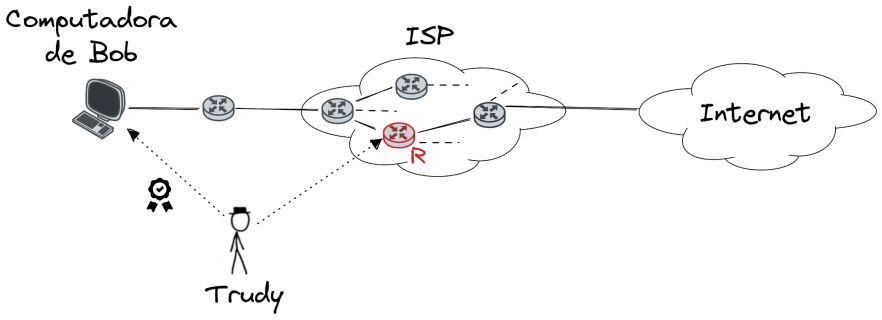


13. Luego de un fracaso rotundo en el diseño de su primer protocolo seguro, Alice decidió cursar TD4 junto a Bob con el objeto de robustecer sus mecanismos de comunicación en Internet. Mientras estudiaban para el parcial, Alice propuso un nuevo protocolo en el que, en primer lugar, envía a Bob un mensaje *Hello* donde lista qué algoritmos de encriptación simétricos está dispuesta a utilizar: AES, DES o 3DES. Bob responde dicho mensaje indicando el algoritmo simétrico seleccionado y proveyendo su certificado X.509 correctamente respaldado por una CA confiable. Posteriormente, Alice valida el certificado y, en caso de comprobar su autenticidad, extrae de allí la clave pública RSA de Bob, K^+_{Bob}. Luego, genera dos secuencias de bits aleatorias, K_1 y K_2, para emplear como claves de sesión unidireccionales durante la comunicación de datos. Alice ensambla un mensaje con dichas claves, lo cifra con K^+_{Bob} y lo envía a su interlocutor, quien podrá desencriptarlo valiéndose de su clave privada. Una vez culminado este *handshake*, los interlocutores podrán intercambiar datos cifrados con el algoritmo simétrico seleccionado y las claves de sesión generadas por Alice.
    a. A pesar de los esfuerzos de Alice y Bob, un actor malicioso aún podría recuperar los datos enviados en ambos sentidos luego del *handshake* sin ser detectado. Explicar cómo podría realizarse este ataque.
    b. Indicar qué datos contiene el certificado enviado por Bob y cómo ejecuta Alice las validaciones en el punto (3) del *handshake*.



14. En un taller de sniffing de TD4, Alicia y Beto lograron capturar el handshake TLS entre un proceso de su máquina y el servidor HTTPS ubicado en el dominio www.utdt.edu. Inspeccionando en Wireshark los paquetes, ambos detectan y observan el paquete conteniendo el certificado enviado por el servidor.
    a. Alicia sostiene que dicho certificado se encuentra encriptado, mientras que Beto asegura que esta información viajó en texto plano. ¿Quién de los dos tiene razón? Justificar.
    b. Explicar cómo el cliente TLS en el proceso de la máquina de Alicia y Beto ejecuta las validaciones sobre el certificado para garantizar la autenticidad del servidor.
    c. Alicia decide realizar el siguiente experimento: accede a su navegador web, elimina de él todos los certificados root preconfigurados y acto seguido intenta navegar hacia https://www.utdt.edu, https://www.google.com y finalmente hacia http://httpforever.com. Explicar qué resultado obtendrá Alicia en cada una de las tres pruebas del experimento.

15. Supongamos que, en el escenario mostrado en la figura, Trudy logra comprometer la computadora de Bob a través de un *malware* e instala en su sistema un certificado malicioso. Más tarde, Bob accede a la web de su home banking vía HTTPS. Asumiendo que Trudy tiene control total sobre el router R en la red del ISP de Bob, y que todo el tráfico TCP y UDP desde y hacia el host de Bob pasa por dicho router, diseñar un ataque sobre TLS para que Trudy pueda leer los datos de las transacciones bancarias de Bob sin que éste se dé cuenta de ello.



Ejercicios *hands-on*

1. Responder las siguientes preguntas a partir de la implementación de RSA en Sage estudiada en clase:
    a. ¿Cómo se eligen los primos *p* y *q* y el exponente *e*?
    b. ¿Cómo se calcula el exponente *d*?
    c. ¿Cómo podemos operar con tiras de bytes arbitrarias?
    d. ¿Qué ocurre si el mensaje a encriptar termina siendo más grande que el módulo *n*?
    e. Investigar por qué no es conveniente utilizar RSA como un *block cipher*.

2. A partir del ejercicio anterior,
    a. Investigar en qué consiste el algoritmo de exponenciación binaria e implementarlo en Python para calcular exponenciaciones modulares.
    b. Investigar e implementar en Python el algoritmo de Euclides extendido para calcular inversas modulares.
    c. Reemplazar en el script visto en la clase la función `pow` y la función `xgcd` por las implementaciones desarrolladas para encriptar/desencriptar y para calcular el exponente *d*, respectivamente.

3. Responder las siguientes preguntas a partir del script de encriptación simétrica vía AES estudiado en clase:
    a. ¿Cómo se configura el modo de encriptación?
    b. ¿Qué pasa si se encriptan 8 bytes en modo CBC?
    c. ¿Qué pasa si se encriptan en modo CTR?

d. Investigar qué otros modos de encriptación de *block ciphers* soporta la librería PyCryptodome. Modificar el script para operar con al menos uno de ellos.

4. Implementar en Python los modos de encriptación ECB, CBC y CTR. Una vez hecho esto, revisitar el script del ejercicio anterior de forma tal de utilizar estas implementaciones para cifrar datos con AES.

5. Investigar cómo utilizar funciones de hash criptográficas en la librería PyCryptodome. Desarrollar un script similar al del ejercicio anterior para calcular hashes de datos arbitrarios con una función de hash parametrizable por línea de comando (tomar como opciones válidas MD5, SHA-1, SHA-256 y SHA-512).

6. Investigar en qué consiste el esquema de Merkle-Damgård para construir funciones de hash criptográficas. Implementar este esquema en Python y luego utilizarlo para implementar la función de hash SHA-1.

7.
   a. Utilizar la librería `openssl` para generar un certificado X.509 autofirmado. ¿Qué información debemos suministrar?
   b. Inspeccionar los certificados de tu navegador web:
      i. ¿Cuáles son las CAs que emitieron los certificados *root* de confianza?
      ii. ¿Cómo es la cadena de certificados proveniente de `https://www.utdt.edu`?
      iii. ¿Qué pasa si agregamos nuestro certificado anterior a la lista de certificados root?

8. Implementar en Python un servidor HTTPS que ofrezca el certificado X.509 autofirmado del ejercicio anterior.
   a. Ejecutarlo para que escuche en el puerto 8443 del `localhost`. ¿Qué sucede si accedemos a esta dirección desde un navegador web local?
   b. Agregar el certificado autofirmado al repositorio de certificados de tu sistema y volver a repetir el experimento.

c. Regenerar el certificado configurando un período de validez que no abarque la fecha actual. ¿Qué sucede si accedemos ahora a nuestro servidor HTTPS desde el navegador?