Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales
Departamento de Ciencias de la Computación
Seguridad Informática
2023 Maximiliano Cristia Alejandro Hernandez

# Práctica 3: Criptografía y Protocolos

**1.** Usualmente se dice que el método *one-time-pad* es un método irrompible. Piense un posible ataque para one-time-pad teniendo dos textos de la misma longitud cifrados con la misma clave. Si consigue un ataque exitoso, cómo puede ser entonces que el método sea clasificado usualmente como "irrompible"?

**2.** Las claves de DES son cortas para los requerimientos actuales. Una posibilidad es tener dos claves $k_1$ y $k_2$, y encriptar el mensaje usando primero $k_1$, y luego encriptar el resultado usando $k_2$. Proponga cómo funcionaría la desencriptación. Qué vulnerabilidades puede tener este esquema, asumiendo que el atacante tiene mucha memoria disponible?

**3.** Explicar cómo funciona la mejora 3DES para DES. Explicar por qué esta alternativa no tiene los problemas del esquema visto en el ejercicio anterior.

**4.** Si se desea encriptar un archivo de 1M. Cómo utilizaría RSA para encriptarlo?

**5.** Suponga que Alice y Bob tienen claves RSA públicas guardadas en un servidor. Ellos se comunican regularmente usando mensajes autenticados y confidenciales. Eve desea leer los mensajes, pero no puede crackear las claves RSA privadas de ninguno de ellos. Sin embargo, Eve puede hackear el servidor y alterar los archivos donde se guardan las claves públicas de Alice y Bob.

1. Cómo debería alterar el archivo Eve de manera que pueda leer los mensajes confidenciales entre Alice y Bob, y simular mensajes de ambos?
2. Cómo podría Alice y/o Bob darse cuenta de la modificación de las claves públicas?

**6.** Cuál es la diferencia entre los modos de operación ECB y CBC? Cuál recomendaría para encriptar el contenido de una imagen en forma de mapa de bits?

**7.** Resolver el challenge 8 del Set 1 de Matasano (http://cryptopals.com/).

[ De aquí en adelante, basado en ejercicios de la School of Informatics, University of Edinburgh (http://www.inf.ed.ac.uk/teaching/courses/cs) ]

**8.**

Notations

- aenc(k, m) (resp. senc(k, m)) denotes the asymmetric (resp. symmetric) encryption of the message m under the key k.
- sign(k,m) denotes the signature with key k of the message m.
- m1||m2 denotes the concatenation of message m1 with message m2.

An early version of SSL included the following authentication and key agreement protocol:

A $\rightarrow$ B : Hello

B $\rightarrow$ A : pbk(B), CertB

A $\rightarrow$ B : aenc(pbk(B), A||K)

B $\rightarrow$ A : senc(K,N)

A $\rightarrow$ B : senc(K,CertA||sign(pvk(A),N))

where:

- K is a fresh session key generated by A
- N is a fresh nonce generated by B
- CertA and CertB are certificates for A and B's public keys respectively.

a. This protocol is flawed. Describe an attack on the protocol.

b. Fix the protocol by changing it as little as possible.

**9.** Write the following statements in BAN logic:

- a. "*Alice believes that Bob believes they share a secret key. Bob has told Alice the key.*" Can we conclude that Alice and Bob share a secret key? Why or why not?
- b. "*Bob believes everything a trusted server says. Alice also trusts the server but only to make keys for her. The server sends a fresh shared key to Alice and Bob.*"

**10.** Answer the following questions concerning the use of BAN.

a. The BAN jurisdiction rule is:

$$\frac{P \overset{\texttt{cree}}{\Rightarrow} Q \overset{\texttt{tjs}}{\Rightarrow} X \qquad P \overset{\texttt{cree}}{\Rightarrow} Q \overset{\texttt{cree}}{\Rightarrow} X}{P \overset{\texttt{cree}}{\Rightarrow} X}$$

We've seen that this rule can be used for trusted servers that do key generation. Give another example of how this might be used in real-life computing.

b. BAN logic assumes that all parties communicating are honest. This is an unrealistic assumption for the real-life use of protocols. How does BAN logic deal with it to prevent lacking effectiveness?

c. If in the BAN system of logic everyone is always honest, why is it possible for one party to see another say something and not automatically believe them? (Hint: think on what else the jurisdiction rule asks for)

**11.** The Kerberos protocol is a key exchange protocol that uses a trusted server. It has the steps:

$$\begin{array}{lcl} A \to S & : & A, B \\ S \to A & : & \{Ts, L, Kab, B, \{Ts, L, Kab, A\}_{Kbs}\}_{Kas} \\ A \to B & : & \{Ts, L, Kab, A\}_{Kbs}, \{A, Ta\}_{Kab} \\ B \to A & : & \{Ta + 1\}_{Kab} \end{array}$$

We can rewrite this in idealized protocol as:

$$\begin{array}{lcll} S \to A & : & \{Ts, (A \longleftrightarrow^{Kab} B), \{Ts, (A \longleftrightarrow^{Kab} B)\}_{Kbs}\}_{Kas} & \\ A \to B & : & \{Ts, (A \longleftrightarrow^{Kab} B)\}_{Kbs}, \{Ta, (A \longleftrightarrow^{Kab} B)\}_{Kab} & \text{from A} \\ B \to A & : & \{Ta, (A \longleftrightarrow^{Kab} B)\}_{Kab} \quad \text{from B} & \end{array}$$

a. What are the key differences between the normal way in which security protocols are written and idealized protocol? Give short reasons for leaving out the parts of the Kerberos protocol that are left out of the idealized version.

b. The assumptions made by the Kerberos protocol can be summarized in BAN logic as:

$$
\begin{array}{lcl}
A & \models & A \longleftrightarrow^{Kas} S \\
S & \models & A \longleftrightarrow^{Kas} S \\
S & \models & A \longleftrightarrow^{Kab} B \\
B & \models & B \longleftrightarrow^{Kbs} S \\
S & \models & B \longleftrightarrow^{Kbs} S \\
A & \models & (S \models\!\Rightarrow A \longleftrightarrow^{K} B) \\
B & \models & (S \models\!\Rightarrow A \longleftrightarrow^{K} B) \\
A & \models & \sharp(Ts) \\
B & \models & \sharp(Ts) \\
B & \models & \sharp(Ta)
\end{array}
$$

Briefly, explain in Spanish (i.e. in natural language) what the assumptions are (hint: do this by breaking the assumptions down into 3 groups).