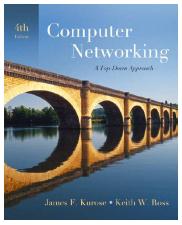
Introducción a las Redes de Computadoras

Capítulo 8
Seguridad en la Red



Computer Networking: A Top Down Approach, 4th edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2007.

Nota acerca de las transparencias del curso:

Estas transparencias están basadas en el sitio web que acompaña el libro y han sido modificadas por los docentes del curso.

Capítulo 8: Seguridad en la Red

Objetivos:

- Entender los principios de la seguridad en la red:
  - criptografía y sus varios usos más allá de la "confidencialidad"
  - integridad de los mensajes
  - autenticación
- seguridad en la práctica:
  - Seguridad en las capas de aplicación, transporte, red, enlace
  - firewalls e *Intrusion Detection Systems*

Capítulo 8: Agenda

8.1 ¿Qué es seguridad en la red?
8.2 Principios de la criptografía
8.3 Integridad de los mensajes
8.4 Autenticación del *end point*
8.5 e-mail seguro
8.6 Asegurando las conexiones TCP: SSL
8.7 Seguridad en la capa de red: IPsec
8.8 Asegurando las redes LAN inalámbricas
8.9 Seguridad Operacional: *firewalls* e IDS

Terminología y definiciones

~~Encriptar~~ Cifrar

- "Disfrazar" la información para transmitirla por un canal inseguro

~~Desencriptar~~ Descifrar

- Obtener el mensaje original a partir de la información "disfrazada"

¿Qué es seguridad en la red?

Confidencialidad: sólo el emisor y el receptor deberían "entender" el contenido del mensaje
- el emisor cifra el mensaje
- el receptor descifra el mensaje

Autenticación: el emisor y el receptor buscan confirmar la **identidad** del otro

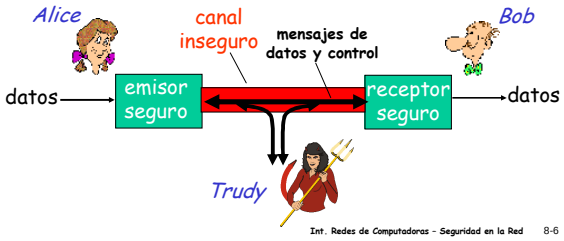
Integridad del mensaje: el emisor y el receptor buscan asegurar que el mensaje no ha sido alterado (en tránsito o después) sin ser detectado

Disponibilidad: los servicios deben estar disponibles para los usuarios

No repudio: no poder negar la autoría de un mensaje

Amigos y enemigos: *Alice, Bob, Trudy*

- muy bien conocidos en el mundo de la seguridad en redes
- *Bob, Alice* (¡amantes!) se quieren comunicar de manera "segura"
- *Trudy* (intruso) podría interceptar, borrar, agregar mensajes



Ataques pasivos y activos

- Pasivos
  - Escuchar el tráfico
- Activos
  - Desviar el tráfico
  - Desviar el tráfico, modificarlo y enviarlo al destino
  - Inyectar tráfico

¿Quiénes pueden ser Bob y Alice?

- Web *browser/server* para transacciones electrónicas (p.e., compras *on-line*)
- Cliente/servidor en banca *on-line*
- servidores DNS
- Cliente/Servidor DNS
- routers intercambiando actualizaciones de tablas de *routing*
- ...

¡Hay chicos malos (y chicas)!

P: ¿Qué puede hacer un "chico malo"?
R: imucho!, por ejemplo:

- *eavesdrop:* interceptar mensajes
- *insert:* mensajes en la conexión
- *impersonation:* puede falsificar (*spoof*) la dirección origen de un paquete (o cualquier campo de un paquete)
- *session hijacking:* secuestrar una sesión asumiendo el rol del emisor o del receptor
- *denial of service*: evitar que el servicio se utilizado por sus usuarios legítimos (p.e., sobrecargando los recursos)
- *replay attack*: escuchar una secuencia de mensajes (puede ser inentendible para el intruso) y luego repetirla

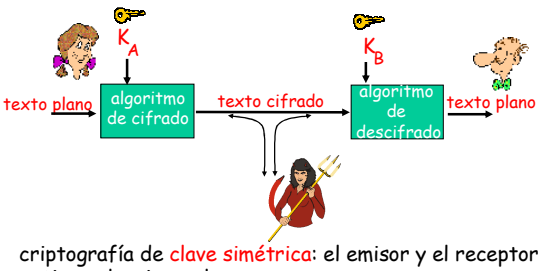
Capítulo 8: Agenda

8.1 ¿Qué es seguridad en red?
8.2 Principios de la criptografía
8.3 Integridad de los mensajes
8.4 Autenticación del *end point*
8.5 e-mail seguro
8.6 Asegurando las conexiones TCP: SSL
8.7 Seguridad en la capa de red: IPsec
8.8 Asegurando las redes LAN inalámbricas
8.9 Seguridad Operacional: *firewalls* e IDS

Conceptos

- Criptografía
  - "Escribir secreto"
- Criptoanálisis
  - Técnicas para conocer la información cifrada
  - Lo ideal, descubrir la clave involucrada
- Criptología = Criptografía + Criptoanálisis

El lenguaje de la criptografía



criptografía de clave simétrica: el emisor y el receptor tienen la misma clave

criptografía de clave pública: clave de cifrado (*pública*), clave de descifrado *secreta* (privada)

Más conceptos

- Criptosistema
  - Quíntupla
    - **M**: conjunto de todos los mensajes sin cifrar (texto claro o texto plano)
    - **C**: conjunto de todos los posibles mensajes cifrados (texto cifrado)
    - **K**: conjunto de claves involucradas
    - **E**: conjunto de transformaciones de cifrado o funciones de cifrado que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente para cada valor posible de *k*
    - **D**: conjunto de transformaciones de descifrado
- Dos tipos de criptosistema: simétricos o de clave privada, asimétricos o de clave pública

Más conceptos (II)

- En la práctica, se suelen utilizar combinados (simétricos y asimétricos), pues los asimétricos son más costosos computacionalmente
- Asimétricos para codificar la clave simétrica a utilizar
- A veces el tiempo hacer perder relevancia a la información que protegemos
  - Por ejemplo: una primicia

Criptografía de clave simétrica (los códigos)

cifrado por sustitución: sustituir unos símbolos por otros

cifrado monoalfabético: sustituye una letra por otra (César)

texto plano: **abcdefghijklmnopqrstuvwxyz**

texto cifrado: **mnbvcxzasdfghjklpoiuytrewq**

P.e.: texto plano: *bob. i love you. alice*

texto cifrado: *nkn. s gktc wky. mgsbc*

cifrado polialfabético: varios monoalfabéticos y un patrón a seguir

cifrado por transposición: cambia el orden de los símbolos dentro del texto. n columnas, se escribe en horizontal y permutación a seguir para cifrar

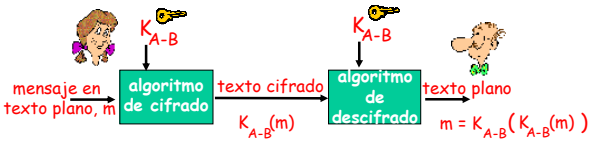
Criptoanálisis

- Posibles ataques
  - Sólo texto cifrado
    - El intruso conoce sólo una porción del texto cifrado
    - Análisis estadístico
  - Texto cifrado conocido
    - El intruso conoce una porción del texto plano y su correspondiente texto cifrado
  - Texto plano elegido
    - El intruso elige texto plano y ve la salida (texto cifrado)
  - Fuerza bruta

Un poco de historia

- Origen de la criptografía: contexto militar
- Ahora también el ámbito civil (investigadores universitarios)
  - Impactos comerciales importantes
    - Comercio electrónico
    - Telefonía celular
- La seguridad basada en la oscuridad (*security through obscurity*), por lo menos en lo que respecta a los algoritmos de cifrado, no ha sido una buena elección

Criptografía de clave simétrica



criptografía de clave simétrica: Bob y Alice comparten la misma clave (simétrica): K_{A-B}

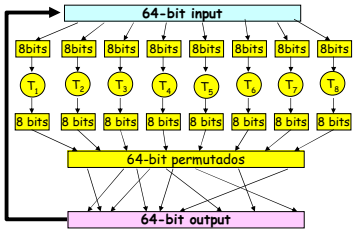
- En los de sustitución como el de César no hay clave, sólo hay un algoritmo secreto
- P: ¿cómo Bob y Alice acuerdan un valor de clave?

Clases de cifrado

- Cifrado por bloques (*block cipher*)
  - Procesa el mensaje en bloques de largo fijo (en general, de 64 ó 128 bits)
- Cifrado por flujo (*stream cipher*)
  - Trata al mensaje a procesar como si fuera una fuente continua de bits

Cifrado en Bloques

loop para **n** pasadas



- Una sola pasada: 1 bit de entrada afecta 8 bits de salida
- Múltiples pasadas: cada bit de entrada afecta a (casi) todos los bits de salida
- Ejemplos de algoritmos de cifrado por bloques: DES, 3DES, AES

Criptografía de clave simétrica
DES: *Data Encryption Standard*

- estándar de cifrado de USA (1977 – FIPS 46)
- clave simétrica de 56 bits; texto plano de entrada de 64 bits, 16 rondas, cada una con una clave distinta
- ¿Qué tan seguro es DES?
  - Desafío DES: frase descifrada con fuerza bruta
    - 1977: 4 meses, 1988: 41 días, 1988: 56 horas, 1999: 22 horas
- "haciendo DES más seguro": DES múltiple (1979)
  - usar 3 claves secuencialmente (triple-DES) en cada dato

Criptografía de clave simétrica
AES: *Advanced Encryption Standard*

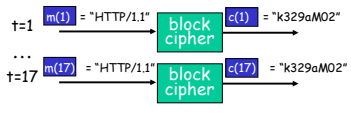
- nuevo (Nov. 2001) estándar del NIST (FIPS PUB 197) de clave simétrica, reemplazando a DES
- Llamado público internacional: algoritmo ganador Rijndael ("reindal")
- procesa datos en bloques de 128 a 256 bits en saltos de a 32 bits
- claves de 128, 192, o 256 bits
- Descifrado de fuerza bruta que tome 1 seg. en DES con clave de 56 bits, toma 149 trillones de años en AES con clave de 128 bits

Modos de operación para algoritmos de cifrado por bloques

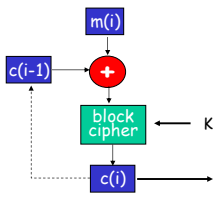
- Para mensajes más largos que un bloque
  - ECB: *Electronic Code Book*
  - CBC: *Cipher Block Chaining*
  - CFB: *Cipher FeedBack*
  - OFB: *Output FeedBack*

Cipher Block Chaining (CBC)

- ECB: si el bloque de entrada es repetido, produce el mismo texto de cifrado:



- XOR del bloque de entrada i-ésimo, **m(i)**, con el bloque previo del texto cifrado, **c(i-1)**
  - **c(0)** = **IV**, transmitido en claro
  - ¿qué ocurre en el escenario "HTTP/1.1" de más arriba?



Criptografía de clave pública

criptografía de clave *simétrica*

- Requiere que tanto el emisor como el receptor conozcan la clave secreta compartida
- P: ¿Cómo pueden acordar la clave si por ejemplo nunca se vieron?

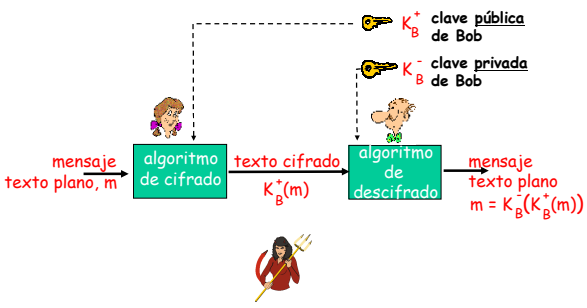
criptografía de clave *pública*

- enfoque radicalmente distinto (Diffie-Hellman, 1976; RSA, 1978)
- el emisor y el receptor *no* comparten una clave secreta
- Cada uno tiene dos claves, la clave ***pública*** la conocen *todos*, la clave ***privada*** la conoce *sólo su dueño*

Aplicaciones de algoritmos asimétricos

- Cifrado de la información con la clave pública, para que viaje en canales inseguros, sin tener que transmitir nunca la clave de descifrado
- Autenticación de mensajes, con la clave privada, a partir de funciones que nos permiten obtener una firma o resumen del mensaje

Criptografía de clave pública



Algoritmos de cifrado de clave pública

Requerimientos:

1. necesitamos $K_B^+(\cdot)$ y $K_B^-(\cdot)$ tal que
   $K_B^-(K_B^+(m)) = m$
2. Dada la clave pub K_B^+, debería ser imposible computar la clave privada K_B^-, aún conociendo el algoritmo

Algoritmo RSA: Rivest, Shamir, Adleman

RSA: números primos y aritmética modular

- Un entero N es primo **sii** sólo es divisible entre N y 1.
- Todo entero se puede descomponer en un producto único de **factores primos**
- **Fortaleza de RSA**: dificultad para factorizar números grandes

RSA

- Las claves pública y privada se calculan a partir de un número que se obtiene como producto de dos números primos grandes
- El atacante, si quiere recuperar un texto claro a partir del texto cifrado y de la clave pública, se enfrentará a un **problema de factorización**. No se conoce públicamente otro mecanismo
- Debería conocer los nros. primos, lo cual es un problema computacionalmente *imposible*

RSA: Seleccionando las claves

1. Seleccione dos números primos largos ***p***, ***q***. (p.e., de 512 o 1024 bits cada uno)
2. Calcule *n = pq* y ***z*** *= (p-1)(q-1)*
3. Seleccione *e* (con ***e*** *< **n***) que no tenga factores comunes con ***z***. (*e* y *z* son "primos relativos").
4. Seleccione *d* tal que ***ed-1*** es exactamente divisible entre *z* (en otras palabras: ***ed*** **mod** ***z*** *= **1***).
5. La clave *pública* es $\underbrace{(n,e)}_{K_B^+}$. La clave *privada* es $\underbrace{(n,d)}_{K_B^-}$.

RSA: Cifrado, descifrado

0. Dados (n,e) y (n,d) calculados como vimos antes
1. Para cifrar un patrón de bits, ***m***, calcular
   $c = m^e \bmod n$ (resto de cuando m^e es dividido entre ***n***)
2. Para descifrar un patrón de bits recibido, ***c***, calcular
   $m = c^d \bmod n$ (resto de cuando c^d es dividido por ***n***)

$$m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$$

RSA ejemplo:

Bob selecciona *p=5, q=7*. Entonces *n=35, z=24*.
e=5 (entonces ***e*** *y* ***z*** son primos relativos).
d=29 (entonces ***ed-1*** es exactamente divisible entre ***z***).

cifrado:

letra	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

descifrado:

c	c^d	$m = c^d \bmod n$	letra
17	481968572106750915091411825223071697	12	l

RSA: ¿Por qué funciona? $m = (m^e \bmod n)^d \bmod n$

Teoría de números: Si p,q son primos y $n = pq$, entonces:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = \boxed{m^{ed} \bmod n}$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

$$= m^1 \bmod n$$

(dado que elegimos ed divisible por $(p-1)(q-1)$ con resto 1)

$$= \boxed{m}$$

Capítulo 8: Agenda

8.1 ¿Qué es seguridad en red?
8.2 Principios de la criptografía
8.3 Integridad de los mensajes
8.4 Autenticación del *end point*
8.5 e-mail seguro
8.6 Asegurando las conexiones TCP: SSL
8.7 Seguridad en la capa de red: IPsec
8.8 Asegurando las redes LAN inalámbricas
8.9 Seguridad Operacional: *firewalls* e IDS

Integridad de los mensajes (autenticación de los mensajes)

Bob recibe un mensaje (cifrado o no) de Alice y quiere asegurarse:

- que el mensaje viene efectivamente de Alice
- que el mensaje no fue cambiado desde que Alice lo envió

Funciones de Hash (resumen) Criptográfico:

- Un hash o *message digest* es una función **unidireccional** que toma una entrada **m** (mensaje), produce un valor de **longitud fija**, y de muchos menos bits, **s=H(m)**. Esta es la propiedad de compresión
- Unidireccional: computacionalmente imposible invertir a la función o sea, dado **s** encontrar **m** tal que **s=H(m)**

Propiedades de las funciones hash

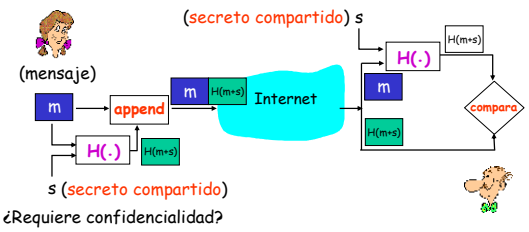
- dado **m**, es fácil calcular **s = H(m)**
- dado **s = H(m)**, (**m** desconocido), es computacionalmente imposible determinar **m**
- es computacionalmente imposible encontrar **m** y **m'** tal que **H(m) = H (m')**
- hashes "randómicas"
  - el **H(m)** debe tener aproximadamente la mitad de sus bits en 1
  - Dadas varias entradas, cada bit de sus hashes debe estar en 1 con probabilidad 0.5
- mensajes muy parecidos deben tener "hashes" muy diferentes. " El cambio en 1 bit debe producir un cambio en aproximadamente la mitad de los bit". Propiedad de difusión

Funciones de hash en la práctica

- Función de hash ampliamente utilizada: MD5
  - *Message Digest* "versión 5"
  - Ron Rivest, RFC 1321
  - Procesa el mensaje de entrada en bloques de 512 bits y produce una salida de 128 bits
  - ataques recientes (2005) a MD5
- SHA-1: también muy utilizada
  - *Security Hash Algorithm*
  - US standard [NIST, FIPS PUB 180-1]
  - Procesa el mensaje de entrada en bloques de 512 bits y produce una salida de 160 bits

Message Authentication Code

Bloque de información que junto con el mensaje permite evaluar su **integridad**. Debe depender del mensaje "m" **y de un secreto "s"** que es la clave de autenticación



¿Requiere confidencialidad?

Ejemplo de MAC: **HMAC** (tanto con MD5 con SHA-1)

Firmas digitales

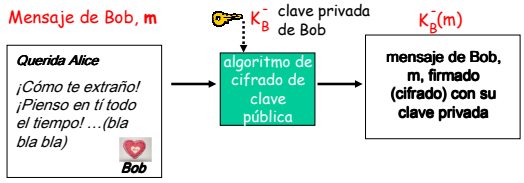
Técnica criptográfica análoga a las firmas manuscritas.

- Combina cifrado asimétrico con funciones hash
- el emisor (Bob) firma digitalmente un documento, estableciéndose que él es el creador/dueño del documento.
- Para ser:
  - verificable: el receptor (Alice) debe convencerse que lo envió Bob;
  - no falsificable: el receptor (Alice) debe convercerse que sólo Bob lo pudo haber enviado
  - no repudiable: el receptor (Alice) debe convencer a otro (un juez) que sólo Bob pudo haber firmado el documento.
- ¿MAC sirve como técnica de firma digital?

Firmas digitales

firma digital simple para un mensaje m:

- Bob "firma" **m** cifrándolo con su clave privada K_B^-, creando un mensaje "firmado", $K_B^-(m)$



Firmas digitales

- suponga que Alice recibe **m** y la firma digital $\mathbf{K_B^-(m)}$
- Alice verifica que m está firmado por Bob aplicando la clave pública de Bob, K_B^+ , a $\mathbf{K_B^-(m)}$; entonces chequea $\mathbf{K_B^+(K_B^-(m)) = m}$.
- si $K_B^+(K_B^-(m)) = m$, quien haya firmado m debe tener la clave privada de Bob.

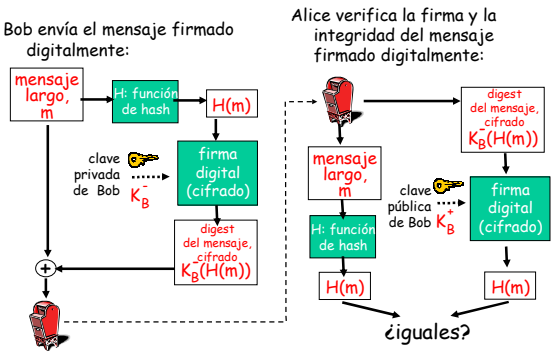
Por lo tanto Alice verifica que:

- Alguien que conoce la clave privada de Bob firmó m.
- Esa persona (Bob) firmó m y no m' (integridad del mensaje)

No repudio:

- ✓ Alice toma m, y la firma $K_B^-(m)$ y puede probar ante la justicia que Bob firmó m.

Firma digital **con hash** (por la crisis… ☺)



Certificación de clave pública

Problema de las claves públicas:

- Cuando Alice obtiene la clave pública de Bob (de sitio web, e-mail, …), ¿cómo ella *sabe* que es la clave pública de Bob y no de Trudy?

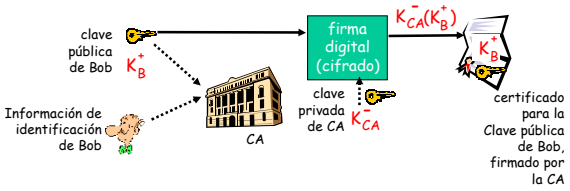
Una solución:

- *Certification Authority* (CA) confiable
  - Validan identidades y gestionan certificados
  - Certificados X.509

Autoridades de Certificación

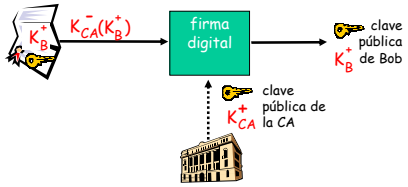
- *Certification Authority* (CA)
  - vinculan una clave pública a una entidad particular, E.
- E registra su clave pública con la CA.
  - E brinda "prueba de identidad" a la CA.
  - CA crea un certificado que es un *binding* de E con su clave pública.
  - El certificado contiene (entre otras cosas) la clave pública de E y está firmado digitalmente por la CA: la CA dice "Esta es la clave pública de E"

Autoridades de Certificación



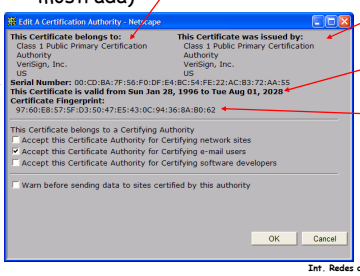
Autoridades de Certificación

- cuando Alice quiere la clave pública de Bob:
  - obtiene el certificado de Bob.
  - aplica la clave pública de la CA al certificado de Bob, chequea su validez y así obtiene la clave pública de Bob



Campos principales de un certificado:

- Número de serie (único para el emisor)
- info acerca del dueño del certificado, incluyendo el algoritmo y el valor de la clave pública (no mostrada)



- info acerca del emisor
- validez
- Firma digital por parte del emisor

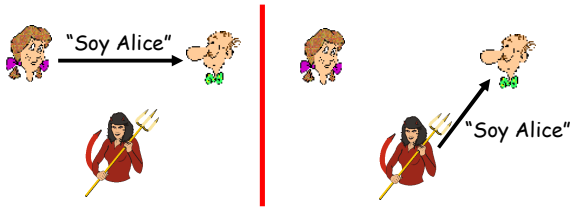
Capítulo 8: Agenda

8.1 ¿Qué es seguridad en red?
8.2 Principios de la criptografía
8.3 Integridad de los mensajes
8.4 Autenticación del *end point*
8.5 e-mail seguro
8.6 Asegurando las conexiones TCP: SSL
8.7 Seguridad en la capa de red: IPsec
8.8 Asegurando las redes LAN inalámbricas
8.9 Seguridad Operacional: *firewalls* e IDS

Autenticación

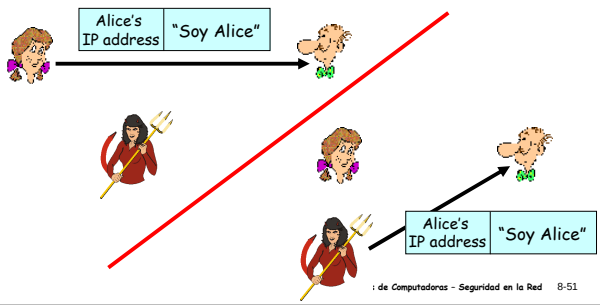
Objetivo: una entidad busca que otra le pruebe su **identidad**

Alice dice "Soy Alice"



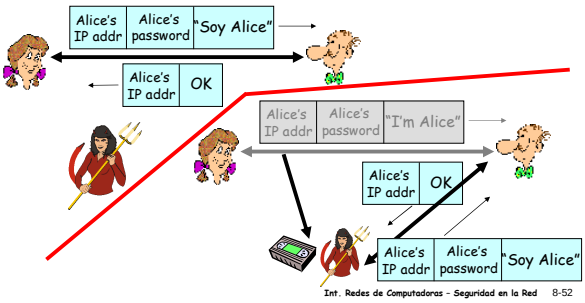
Autenticación: otro intento

Alice dice "Soy Alice" en un paquete IP conteniendo su dirección IP origen



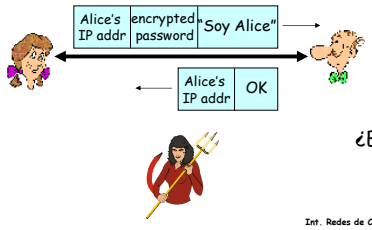
Autenticación: otro intento

Alice dice "Soy Alice" y envía su password secreta para demostrarlo.



Autenticación: otro intento

Alice dice "Soy Alice" y envía su password secreta *cifrada* para probarlo.



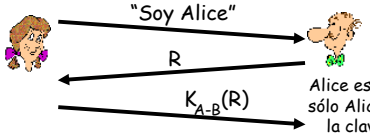
¿Escenario de falla?

Autenticación: otro intento

Objetivo: evitar el *playback attack*

Nonce: número (R) utilizado una sola vez

para probar que Alice "está viva", Bob le envía a Alice un *nonce*, R. Alice debe retornar R, cifrado con la clave secreta compartida



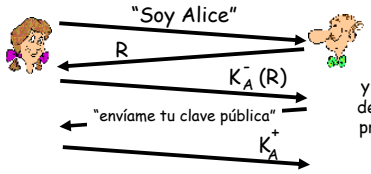
Alice está viva, y sólo Alice conoce la clave para cifrar el *nonce*, entonces debe ser Alice!

Autenticación

Lo visto en el *slide* anterior requiere una clave secreta compartida

- ¿Podemos autenticar utilizando técnicas de clave pública?

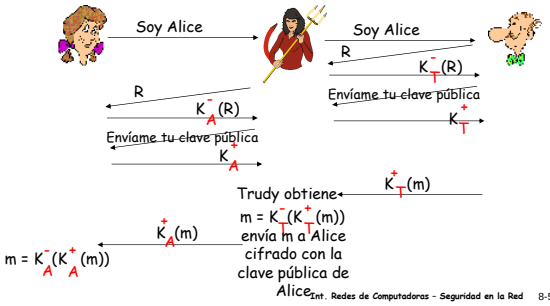
  utilizar *nonce* y criptografía de clave pública



Bob calcula
$K_A^+(K_A^-(R)) = R$
y sabe que sólo Alice debería tener la clave privada, que cifró a R tal que
$K_A^+(K_A^-(R)) = R$

Hueco de seguridad

Ataque ***Man ("Woman") in the Middle***: Trudy (¡ufa!) es Alice para Bob y Bob para Alice



Hueco de seguridad

Ataque ***Man ("Woman") in the Middle***: Trudy (¡ufa!) es Alice para Bob y Bob para Alice

Dificultades para detectarlo:

- Bob recibe todo lo que Alice le envía, y viceversa.
- ¡El problema es que Trudy recibe todos los mensaje también!