

Módulo 06 Seguridad en Redes (Pt. 2)



Redes de Computadoras
Depto de Cs. e Ing. de la Comp.
Universidad Nacional del Sur



Copyright

- Copyright © 2010-2023 A. G. Stankevicius
- Se asegura la libertad para copiar, distribuir y modificar este documento de acuerdo a los términos de la **GNU Free Documentation License**, versión 1.2 o cualquiera posterior publicada por la Free Software Foundation, sin secciones invariantes ni textos de cubierta delantera o trasera
- Una copia de esta licencia está siempre disponible en la página <http://www.gnu.org/copyleft/fdl.html>
- La versión transparente de este documento puede ser obtenida de la siguiente dirección:

<http://cs.uns.edu.ar/~ags/teaching>

Contenidos

- Introducción a la seguridad en redes
- Principios de la criptografía
- Autenticación
- Integridad
- Distribución de claves y de certificados
- Seguridad multinivel
- Sistemas de detección de intrusos
- Vectores de ataque y contramedidas

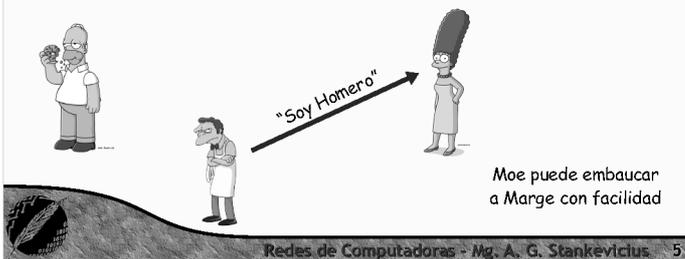
Autenticación

- La autenticación tiene por objeto verificar la identidad de los eventuales interlocutores
 - Volviendo a nuestro ejemplo, la primer aproximación sería que Homero le diga a Marge "Soy Homero"
- Protocolo v1.0:



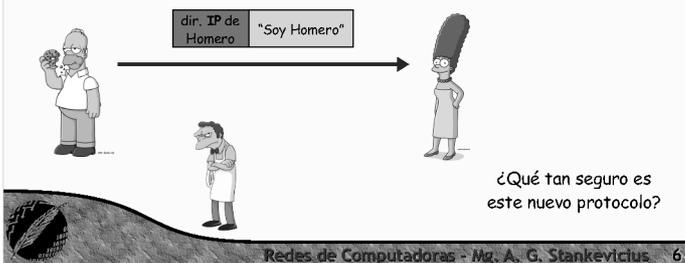
Engaño

- Debemos tener en cuenta que en una red los interlocutores suelen no estar cara a cara
 - Al protocolo v1.0 se lo puede hacer fallar con suma facilidad:



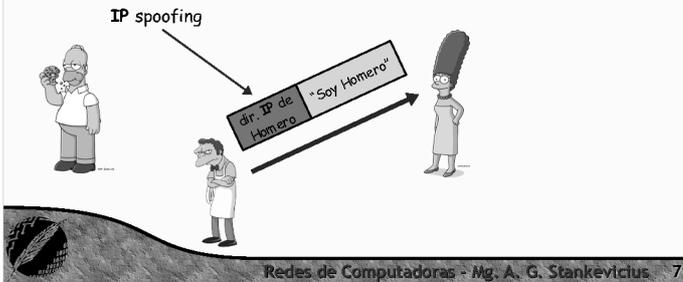
Autenticación de la fuente

- El protocolo v1.0 se lo puede refinar, revisando que el mensaje venga dentro de un datagrama que se origine en la dirección IP de Homero
- Protocolo v2.0:



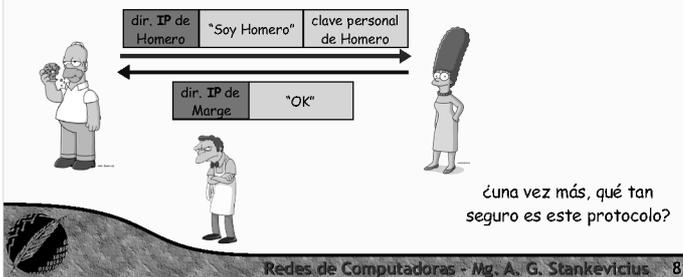
Adulteración de la fuente

- Moe puede usar una aplicación que adultere la dirección IP de origen de los datagramas
→ Esto permite hacer fracasar al protocolo v2.0:



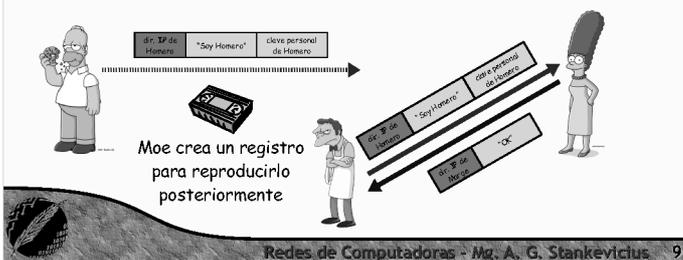
Uso de una clave personal

- Claramente el protocolo es demasiado simple, quizás expandiendo la interacción se corrijan las falencias identificadas:
- Protocolo v3.0:



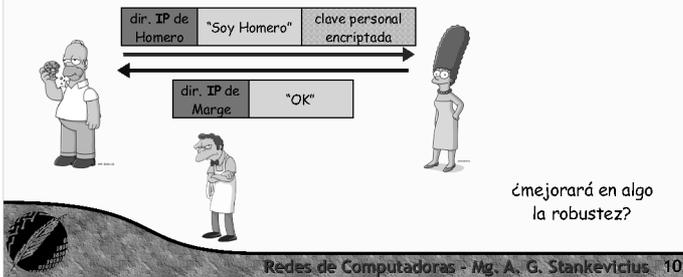
Ataque por reproducción

- Moe puede interceptar el datagrama inicial y reproducirlo cuando sea necesario
→ Este tipo de ataque, denominado "ataque por reproducción", permite quebrar al protocolo v3.0:



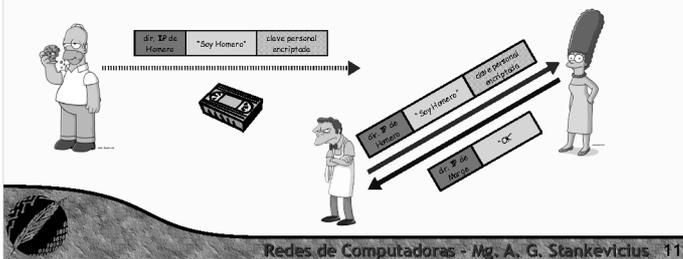
Clave personal encriptada

- Una alternativa para hacer más robusto al protocolo es enviar la clave personal pero esta vez encriptada:
- Protocolo **v3.1**:



Ataque por reproducción

- El ataque por reproducción también registra y reproduce la clave encriptada
- El mismo "ataque por reproducción" también hace fracasar al protocolo **v3.1**:



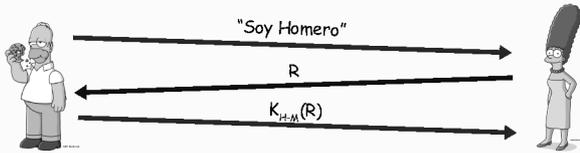
Ataque por reproducción

- Para desactivar los ataques por reproducción basta con incorporar un componente dinámico:
- Este mecanismo se lo conoce como de "desafío-respuesta" (challenge-response)
- Denominaremos nonce (**n**umber used **o**nce) a un número al azar que ha de ser usado sólo una vez
- Protocolo **v4.0** (con desafío-respuesta):
- Homero para comprobar su identidad recibe un nonce generado por Marge, el cual meramente se agrega al mensaje que le quiere enviar originalmente



Intercambio de nonces

- Intercambio de mensajes de acuerdo al protocolo v4.0:



¿resiste el ataque por reproducción?
 ¿qué se le puede criticar a este protocolo?

Marge verifica que el nonce sea el correcto haciendo uso del secreto compartido con Homero

Uso de una clave pública

- ¿Se podrá adaptar la criptografía de clave pública al proceso de autenticación?
- Protocolo v5.0:

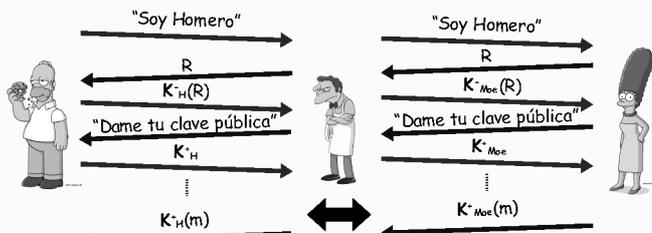


a prueba de hackers...
 ¿lo no??

Marge comprueba si $R = K_H(K_{H,R})$
 ya que sólo Homero cuenta con K_H

Ataque "man in the middle"

- El protocolo v5.0 también resulta vulnerable:



Moe primero recupera m usando su clave privada y luego usa la clave pública de Homero para volver a codificar el mensaje de Marge

Ataque "man in the middle"

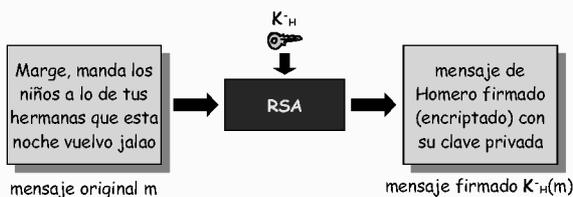
- El ataque "man in the middle" consiste en interceptar y adulterar los mensajes en tránsito
 - El atacante gana acceso a la totalidad de la información intercambiada a través del canal
- Resulta bastante difícil de detectar:
 - En el ejemplo Marge recibe absolutamente todo lo que Homero envía (es decir, si posteriormente se juntan podrán recordar lo conversado)
 - El problema es que Moe también accede a la totalidad de esta información

Firma digital

- La firma digital es una técnica criptográfica que brinda una funcionalidad análoga a la firma manuscrita
 - Por caso, al firmar digitalmente un documento queda perfectamente claro quién fue su autor
 - Más aún, considerando que la firma digital no es falsificable, cualquier persona puede probar a un tercero que aquel que firmó digitalmente un cierto documento es en efecto su autor o responsable

Firma digital

- La criptografía de clave pública permite llevar adelante un modelo directo de firma digital
 - Por caso, Homero para firmar digitalmente un mensaje m sólo tiene que encriptarlo con su clave K_H



Análisis

- Supongamos que Marge recibe el mensaje m y su firma digital $K_H(m)$
 - Marge puede comprobar dos cosas, que el mensaje no haya sido adulterado y que de hecho provenga de Homero
 - Para esto, aplica la clave pública de Homero K_H^+ a la firma digital $K_H(m)$ para verificar si $m = K_H^+(K_H(m))$
 - De ser así, quien firmó digitalmente necesariamente tuvo acceso a la clave privada de Homero (es decir, fue Homero, salvo mediar alguna situación anómala)

Digesto de un mensaje

- El mecanismo de firma digital propuesto sólo es viable para encriptar mensajes cortos
 - Recordemos que el proceso de encriptado es computacionalmente costoso
- Para el caso de los mensajes de gran tamaño, sólo se firma digitalmente el digesto del mismo
 - El digesto de un mensaje viene a ser la “huella digital” (fingerprint) del mismo
 - Es en realidad una función fácil de computar, que retorna un resultado de tamaño fijo denominado hash

Función hash

- La función hash debe cumplir un conjunto de requisitos:
 - Usualmente múltiples mensajes pueden compartir el mismo digesto
 - Da como resultado un digesto del mensaje de tamaño fijo denominado “huella digital” (fingerprint)
 - Por último, dados un mensaje m y su correspondiente función hash H , tiene que ser computacionalmente inviable encontrar otro mensaje m' con el cual colisione, esto que, que $H(m) = H(m')$

MD5

- En la actualidad la función hash **MD5** (Message Digest v5) es una de las más difundidas:
 - Se define formalmente en el **RFC 1321**
 - Ideada por Rivest para reemplazar la ya obsoleta función hash **MD4**
 - Computa un digesto de mensaje de 128 bits en un proceso de cuatro fases de 16 operaciones
 - Para un digesto arbitrario **d** aparentaba ser bastante difícil construir un mensaje **m** cuyo hash **MD5** sea **d**
 - Hoy en día se sabe vulnerable a ataques de colisión

SHA-1 y SHA-2

- También está disponible la función hash alternativa **SHA-1**:
 - Se basa en el mismo principio de **MD4** y **MD5**
 - Se trata de un estándar sancionado por el **NIST** (National Institute of Standards and Technology)
 - Computa un digesto de mensaje de 160 bits en un proceso de cuatro etapas
 - De requerir una función hash más robusta (esto es, más difícil colisionar por fuerza bruta), se puede optar por **SHA-2**, cuyo tamaño de digesto en bits es de 256 o 512

Intermediarios confiables

- Tanto la criptografía de clave simétrica como asimétrica se beneficiarían al disponer de un tercero que haga de intermediario confiable
- En la criptografía de clave simétrica la dificultad consiste en compartir la clave privada
 - El centro de distribución de claves (**KDC**) se encarga de esto
- En cambio, en la criptografía de clave pública la dificultad consiste en compartir esa clave
 - La autoridad de certificación (**CA**) se encarga de esto

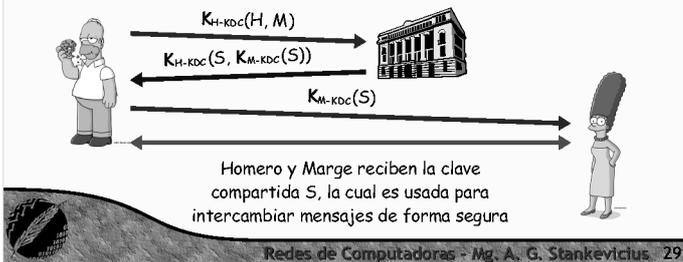
Key Distribution Center

- Supongamos que Homero y Marge necesitan acordar una nueva clave compartida
 - El centro de distribución de claves mantiene diferentes claves personales, una para cada uno de sus usuarios registrados
 - De manera análoga, los usuarios también conocen sus respectivas claves personales



Key Distribution Center

- El KDC facilita el proceso de distribución de una nueva clave compartida
 - Cuando Homero y Marge quieran comunicarse de manera segura, el KDC les genera una nueva clave compartida, válida sólo durante esa interacción

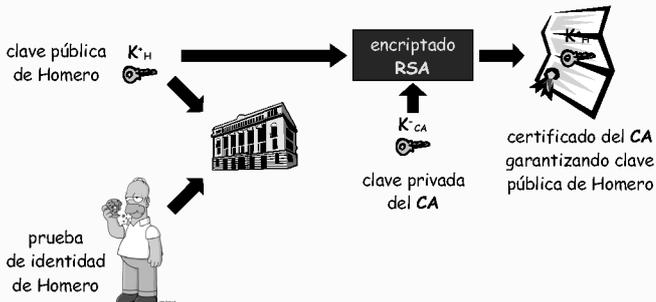


Certification Authority

- El CA se encarga de la distribución de las claves públicas al usar criptografía de clave pública
 - El CA vincula una cierta entidad E a su clave pública
 - La entidad E (por caso, una persona, un server, etc.) suministra al CA una prueba de identidad
 - De resultar aceptable, el CA vincula a E con su clave pública a través de un certificado
 - Este certificado es en esencia la clave pública de E firmada digitalmente por el CA

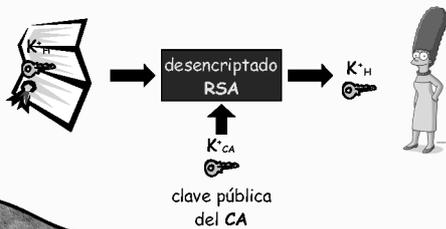


Certification Authority



Certification Authority

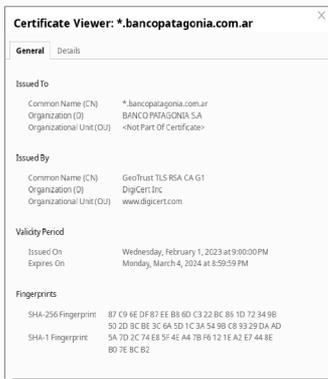
- Cualquier interesado puede extraer la clave pública contenida en un certificado
- Por caso, Marge a partir de la certificación de clave pública de Homero puede recuperar la clave con la certeza de que no ha sido adulterada



Estructura de un certificado

- El certificado suele contener información extra aparte de la clave pública:

- Un número de serie, único para cada CA
- Otros datos relativos al dueño del certificado



¿Preguntas?
