FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

U.PORTO

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Network Security Intro

Redes de Computadores

2021/22

Pedro Brandão

References

- Slides are by Mark Stamp "Information Security: Principles and Practice" 2nd edition (Wiley 2011).
- Some other slides from Dr Lawrie Brown (UNSW@ADFA) for "Computer Security: Principles and Practice", 1/e, by William Stallings and Lawrie Brown
- With adaptations/additions by Pedro Brandão

Driving questions…

- *How to secure protocols?*
- *What are the key objectives of securing protocols?*
- *Are security protocols hard or brittle?*
- *How to use crypto operations to provide those objectives?*
- *What is the path of a packet in the kernel?*
- *Can change it in that path?*

RCom 21/22 - NetSec intro

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

U.PORTO

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

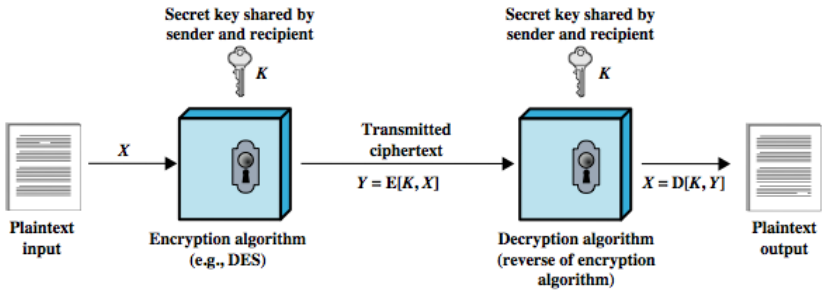
Crypto refresh

Network Security Intro

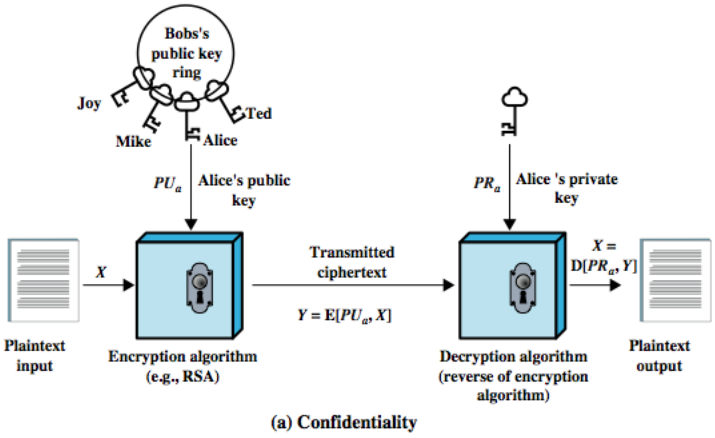
How to Speak Crypto

- A cipher or cryptosystem is used to encrypt the plaintext
- The result of encryption is ciphertext
- We decrypt ciphertext to recover plaintext
- A key is used to configure a cryptosystem
- A symmetric key cryptosystem uses the same key to encrypt as to decrypt
- A public key cryptosystem uses a public key to encrypt and a private key to decrypt
- Nonce == number used once

Symmetric key

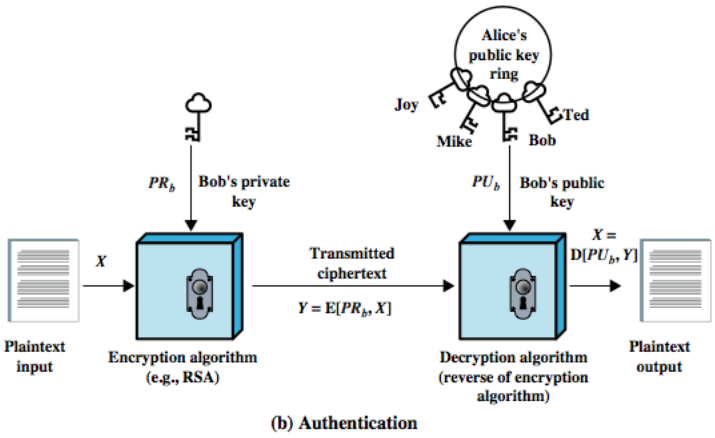


Public Key Encryption



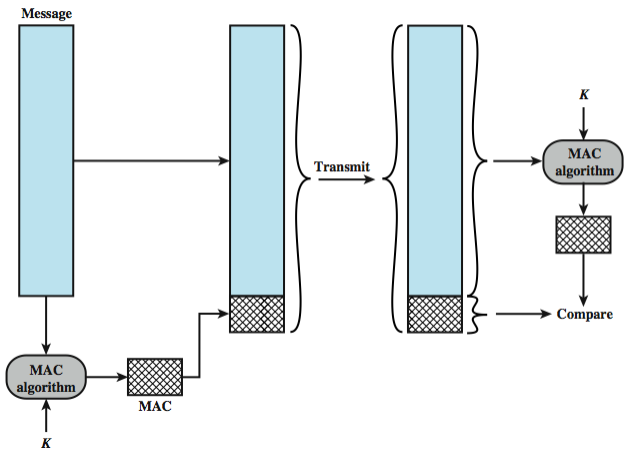
(a) Confidentiality

Public Key Authentication

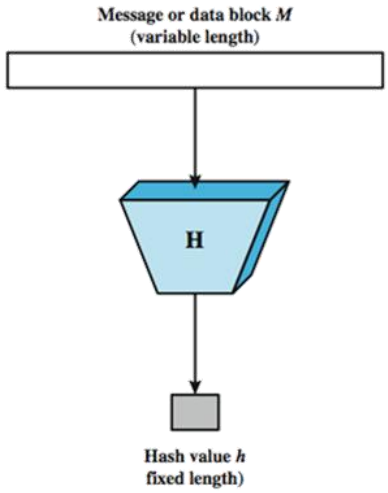


(b) Authentication

Message Authentication Codes



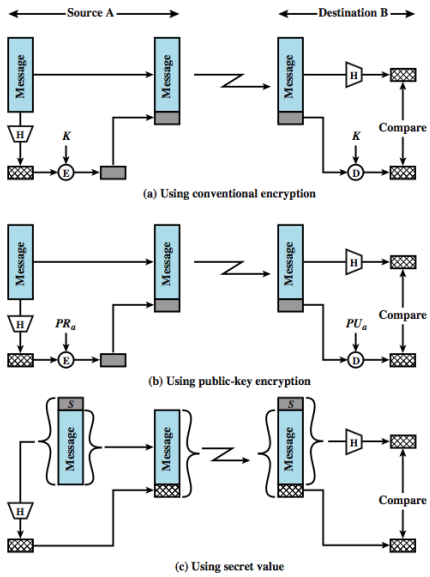
Secure Hash Functions



RCom 21/22 - NetSec intro

U.PORTO

Message Auth



11

RCom 21/22 - NetSec intro

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

U.PORTO

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Internet security protocols

Network security Intro

Protocols

- Protocol flaws can be very subtle
- Several well-known security protocols have significant flaws
  - Including WEP, GSM, and even IPsec, SSL (POODLE and its extension to TLS, DROWN, ROBOT)
- Implementation errors can occur
  - Such as IE implementation of SSL (CVE-2002-0862), OpenSSL (HeartBleed, FREAK, HEIST)
- Not easy to get protocols right…

More attacks on TLS

Security Protocols - Features

- Authentication
  - Of machines, users, i.e. communication end-points
  - Mutual
- Confidentiality
  - Exchanged communications are non-readable/understandable by others
  - Flow of information is confidential
    - Usually hard to achieve
- Crypto integrity
  - Data handled has not been tampered with
    - Discard tampered data
- Non-repudiation
  - Communicators may not denying sending messages
    - Message is associated with peer

Ideal Security Protocol

- Satisfies security requirements
  - Requirements must be precise
- Efficient
  - Small computational requirement
  - Small bandwidth usage, network delays…
- Robust
  - Works when attacker tries to break it
  - Works even if environment changes
- Easy to use & implement, flexible…
- Difficult to satisfy all of these!

RCom 21/22 - NetSec intro

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

U.PORTO

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Simple Security Protocols

Network security Intro

Secure Entry to NSA

1. Insert badge into reader
2. Enter PIN
3. Correct PIN?
   - **Yes?** Enter
   - **No?** Get shot by security guard

ATM Machine Protocol

1. Insert ATM card
2. Enter PIN
3. Correct PIN?
   - **Yes?** Conduct your transaction(s)
   - **No?** Machine (eventually) eats card

Identify Friend or Foe (IFF)

Russian
SU-34

Homs

USAF
F-35
K

2. E(N,K)

1. N

Hama
K

SU in the Middle

USAF
F-35
K

3. N

4. E(N,K)

Homs

2. N

5. E(N,K)

Russian
SU-34

6. E(N,K)

1. N

Hama
K

Authentication Protocols

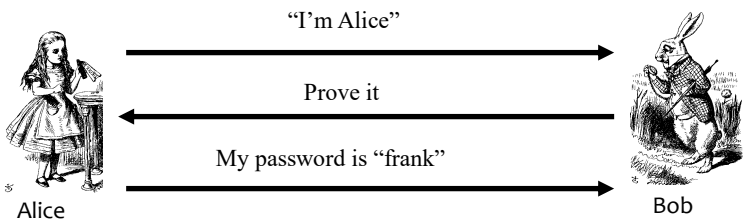
Authentication

- Alice must prove her identity to Bob
  - Alice and Bob can be humans or **computers**
- May also require Bob to prove he's Bob (mutual authentication)
- Probably need to establish a **session key**
- May have other requirements, such as
  - Use public keys
  - Use symmetric keys
  - Use hash functions
  - Anonymity, plausible deniability, etc., etc.

Authentication

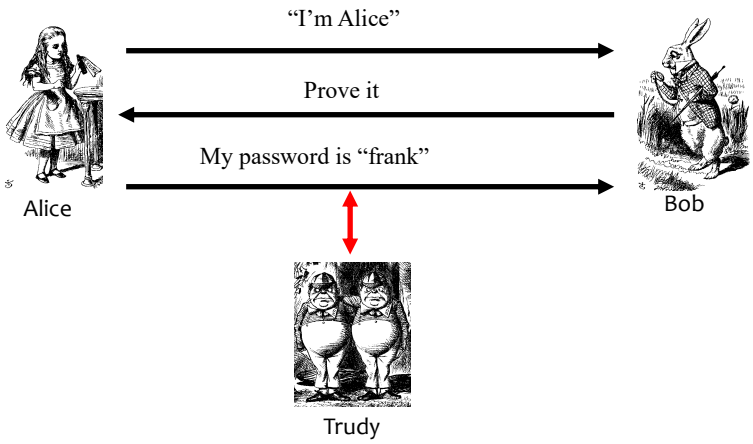
- Authentication on a stand-alone computer is relatively simple
  - Hash password with salt, etc.
  - "Secure path," attacks on authentication software, keystroke logging, etc., are issues
- Authentication over a network is challenging
  - Attacker can passively observe messages
  - Attacker can replay messages
  - Active attacks possible (insert, delete, change)

Simple Authentication

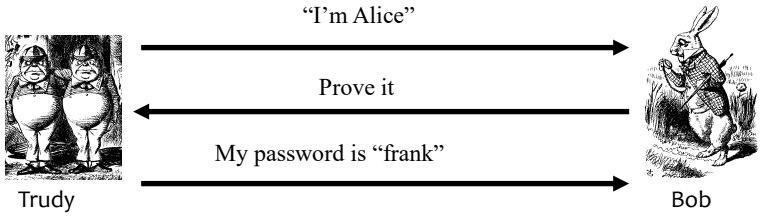


- Simple and may be OK for standalone system
- But insecure for networked system
  - Subject to a replay attack (next 2 slides)
  - Also, Bob must know Alice's password

Authentication Attack

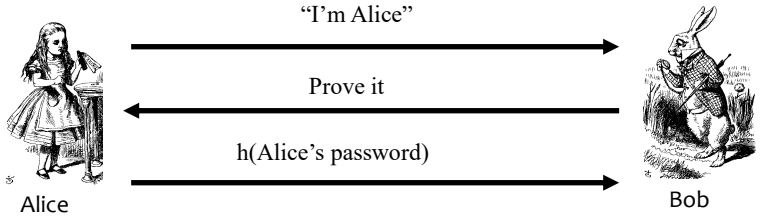


Authentication Attack



- This is an example of a replay attack
- How can we prevent a replay?

Better Authentication



- Better since it hides Alice's password
  - From both Bob and Trudy
- Subject to replay?
  - YES

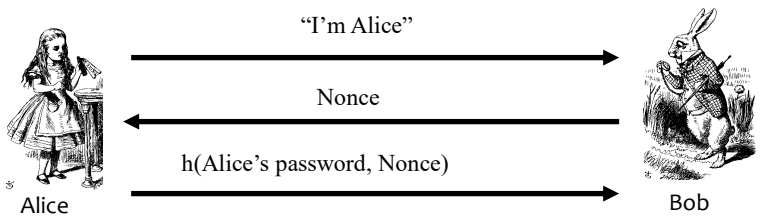
Challenge-Response

- To prevent replay, use challenge-response
  - Goal is to ensure "freshness"
- Suppose Bob wants to authenticate Alice
  - Challenge sent from Bob to Alice
- Challenge is chosen so that
  - Replay is not possible
  - Only Alice can provide the correct response
  - Bob can verify the response

Nonce

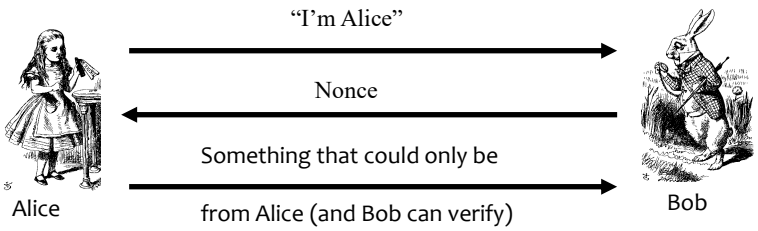
- To ensure freshness, can employ a nonce
  - Nonce == number used once
- What to use for nonces?
  - That is, what is the challenge?
- What should Alice do with the nonce?
  - That is, how to compute the response?
- How can Bob verify the response?
- Should we rely on passwords or keys?

Challenge-Response



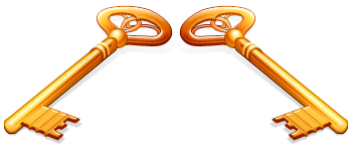
- Nonce is the challenge
- The hash is the response
- Nonce prevents replay, ensures freshness
- Password is something Alice knows
- Bob must know Alice's pwd to verify

Generic Challenge-Response



- In practice, how to achieve this?
- Hashed pwd works…
- Encryption (using keys) is better here (Why?)

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO
U.PORTO
U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO
RCom 21/22 - NetSec intro



Symmetric Keys

Authentication protocols

Network Security Intro

Symmetric Key Notation

- Encrypt plaintext P with key K

  `C = E(P,K)`

- Decrypt ciphertext C with key K

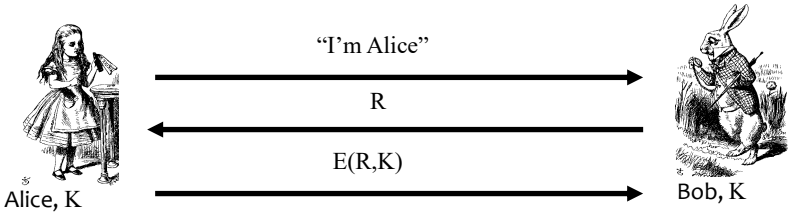
  `P = D(C,K)`

- Here, we are concerned with attacks on protocols, **not** attacks on crypto
  - We assume crypto algorithms secure

Authentication: Symmetric Key

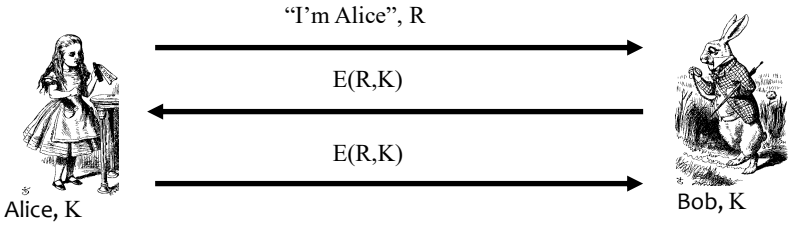
- Alice and Bob share symmetric key K
- Key K known only to Alice and Bob
- Authenticate by proving knowledge of shared symmetric key
- How to accomplish this?
  - Must not reveal key, must not allow replay (or other) attack, must be verifiable, …

Authentication with Symmetric Key



- Secure method for Bob to authenticate Alice
- Alice **does not** authenticate Bob
- So, can we achieve mutual authentication?

Mutual Authentication?

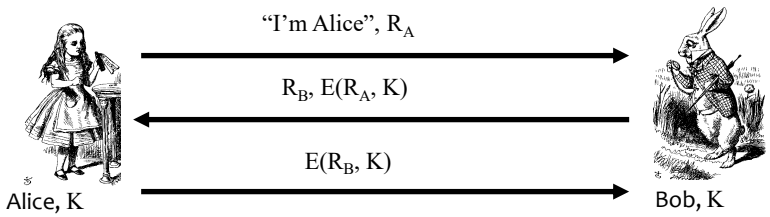


- What's wrong with this picture?
- "Alice" could be Trudy (or anybody else)!

Mutual Authentication

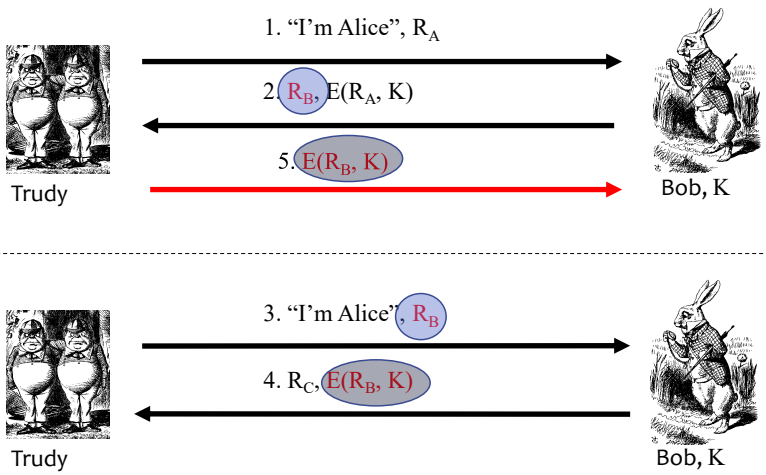
- Since we have a secure one-way authentication protocol…
- The obvious thing to do is to use the protocol twice
  - Once for Bob to authenticate Alice
  - Once for Alice to authenticate Bob
- This has got to work…

Mutual Authentication



- This provides mutual authentication…
- … or does it?

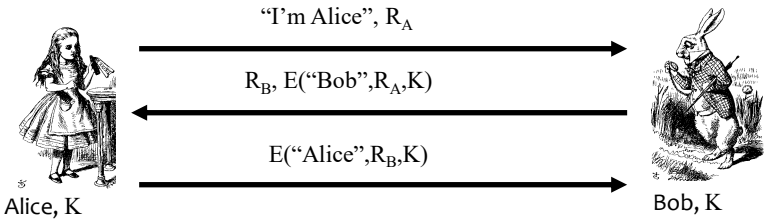
Mutual Authentication Attack



Mutual Authentication

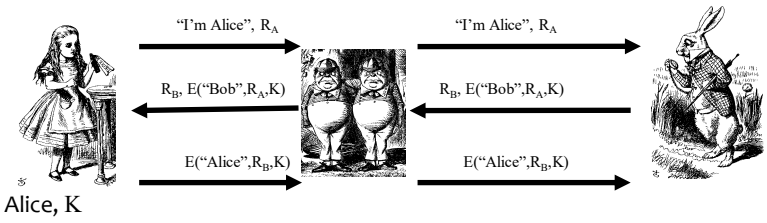
- Our one-way authentication protocol is not secure for mutual authentication
  - Protocols are subtle!
  - The "obvious" thing may not be secure
- Also, if assumptions or environment change, protocol may not be secure
  - This is a common source of security failure
  - For example, Internet protocols

Symmetric Key Mutual Authentication



- *Do these "insignificant" changes help?*
  - Yes!

Symmetric Key Mutual Authentication



- *Is this MiTM?*
- *What else is needed to thwart this?*
  - Session key

Public Keys

Authentication protocols

Network Security Intro

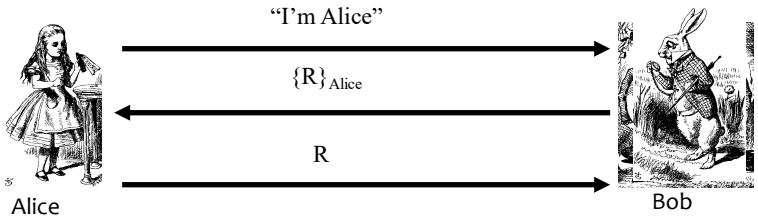
RCom 21/22 - NetSec intro

Public Key Notation

- Encrypt M with Alice's public key: $\{M\}_{Alice}$
- Sign M with Alice's private key: $[M]_{Alice}$
- Then
  - $[\{M\}_{Alice}]_{Alice} = M$
  - $\{[M]_{Alice}\}_{Alice} = M$
- **Anybody** can use Alice's **public key**
- Only **Alice** can use her **private key**

U.PORTO

Public Key Authentication



- Is this secure?
- Trudy can get Alice to decrypt anything!
  - So, should have two key pairs

Public Key Authentication

- Is this secure?
- Trudy can get Alice to sign anything!
  - Same as previous — should have two key pairs

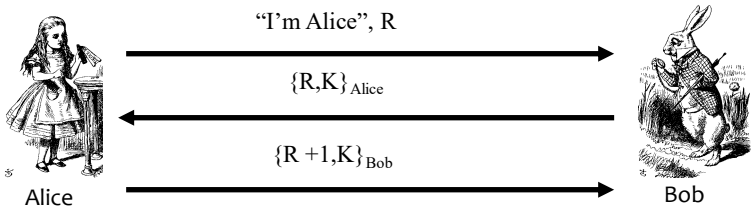
Public Keys

- Generally, a bad idea to use the same key pair for encryption and signing
- Instead, should have…
  - … one key pair for encryption/decryption…
  - … and a different key pair for signing/verifying signatures

Session Key

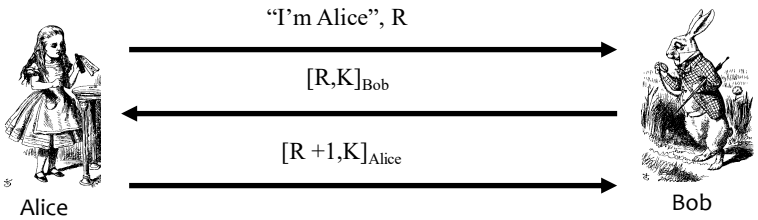
- Usually, a **session key** is required
  - I.e., a symmetric key for a particular session
  - Used for confidentiality and/or integrity
- How to authenticate and establish a session key (i.e., shared symmetric key)?
  - When authentication completed, want Alice and Bob to share a session key
  - Trudy cannot break the authentication…
  - … and Trudy cannot determine the session key

Authentication & Session Key



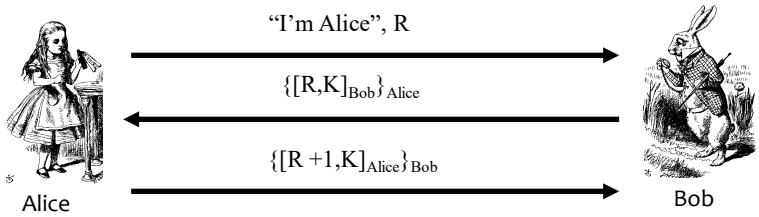
- Is this secure?
  - Alice is authenticated and session key is secure
  - Alice's "nonce", R, useless to authenticate Bob
  - The key K is acting as Bob's nonce to Alice
- No mutual authentication

Public Key Authentication and Session Key



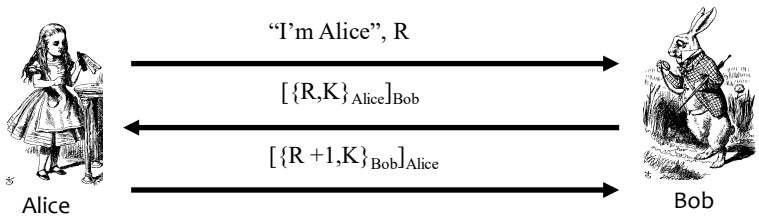
- Is this secure?
  - Mutual authentication (good), but…
  - … session key is not secret (very bad)

Public Key Authentication and Session Key



- Is this secure?
- Seems to be OK
- Mutual authentication and session key!

Public Key Authentication and Session Key



- Is this secure?
- Seems to be OK
  - Anyone can see $\{R,K\}_{Alice}$ and $\{R+1,K\}_{Bob}$

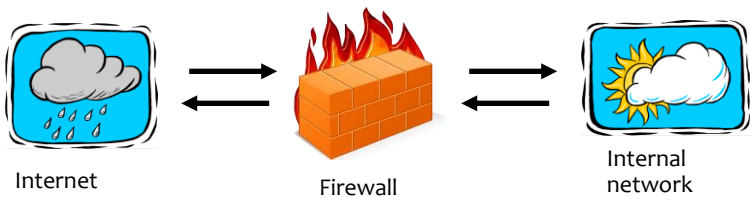
Iptables and netfilter

Firewalling

Network Security Intro

Firewalls

- Firewall must determine what to let in to internal network and/or what to let out
- Access control for the network



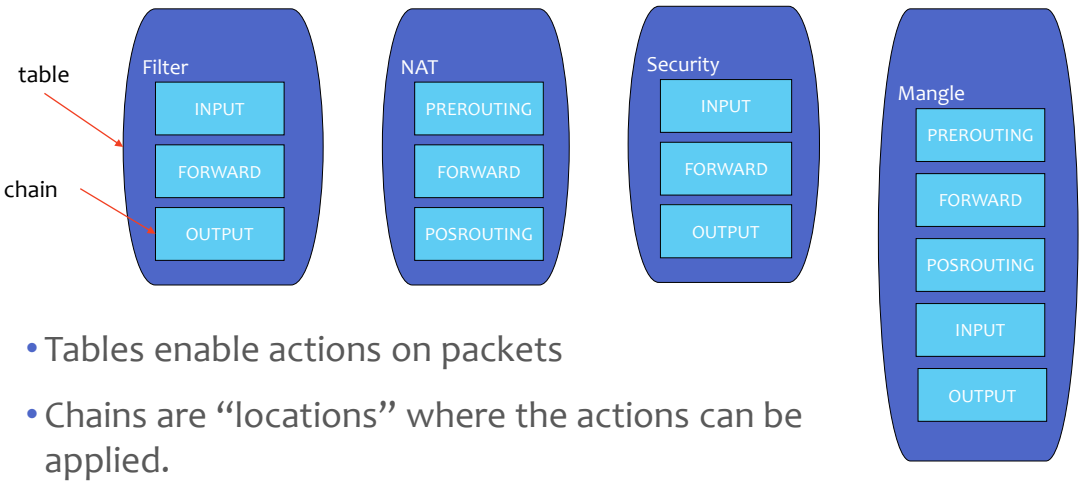
Iptables and netfilter

$> man iptables

`administration tool for IPv4/IPv6 packet filtering and NAT`

- From netfilter.org
  - "provides packet filtering software for the Linux 2.4.x and later kernel series. The netfilter project is commonly associated with iptables and its successor nftables."
  - `nftables` replaces the popular `{ip,ip6,arp,eb}tables`.

Tables contain chains

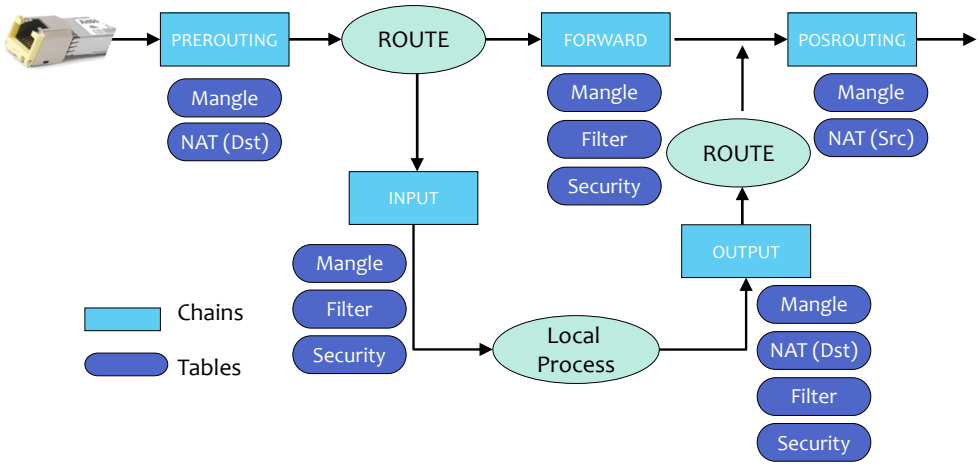


- Tables enable actions on packets
- Chains are "locations" where the actions can be applied.

RCom 21/22 - NetSec intro

U.PORTO

iptables – path of an IP packet on Netfilter



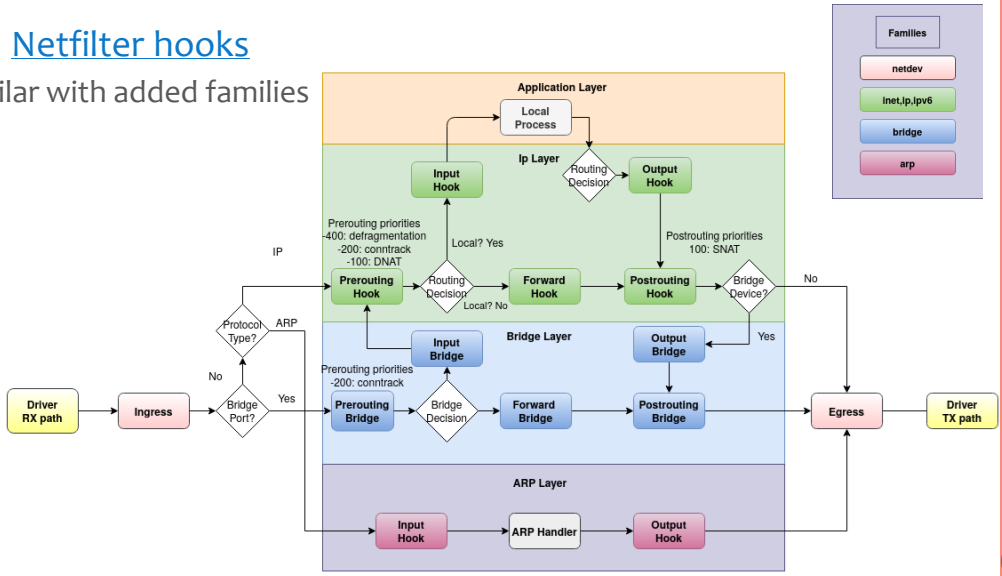
57

RCom 21/22 - NetSec intro

U.PORTO

nftables view

- From Netfilter hooks
  - Similar with added families



58

Iptables and nftables examples

- iptables:

```
## Change source addresses to 1.2.3.4.
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4
```

table: nat — chain: POSTROUTING — target: SNAT

- nftables:

```
## Change source addresses to 1.2.3.4.
nft add rule ip nat POSTROUTING oifname "eth0" counter snat to 1.2.3.4
```

table: nat — chain: POSTROUTING — target: snat

See Moving from iptables to nftables

Summary

- Secure protocols, what are they good for…
  - Authentication protocols
  - Symmetric, public key
- iptables and path of a packet in netFilter

Homework

1. Review slides

2. Read from Tanenbaum
   - Section 8.7.1 –Authentication Based on a Shared Secret Key

3. Answer questions at Moodle

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

U.PORTO

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

End of Network Security Intro