Network monitoring

And wireless

Exercise 1

- Install a new service on the Ubuntu box (f.x. FTP or similar)
  - `apt-get install vsftpd`
- Add it to the monitoring in the nagios

Exercise 1 walk-through

- Try out the ftp tester plugin for Nagios:

```
/usr/lib/nagios/plugins/check_ftp -H localhost
```

- Then add the service to the Nagios monitoring:

```
sudo nano /etc/nagios3/conf.d/services_nagios2.cfg

define service {
  hostgroup_name                  ssh-servers
  service_description             FTP
  check_command                   check_ftp
  use                                     generic-service
  notification_interval           0
}
```

- Check that the Nagios config doesn't contain errors

```
sudo nagios3 -v /etc/nagios3/nagios.cfg
```

- And restart Nagios

```
sudo service nagios3 start
```

Exercise 2

- Try to add monitoring to the Object identifier 1.3.6.1.2.1.1.8.0 (this find timeticks)

- You can use the check_snmp for that

Exercise 2 walk-through

- Use the Nagios check_snmp plugin to test (1 line)

```
/usr/lib/nagios/plugins/check_snmp -H 127.0.0.1 -o
1.3.6.1.2.1.1.8.0 -P 2c -C recorded/linksys-system
```

- Create a customized command to this snmp OID in the file snmp.cfg

```
sudo nano  /etc/nagios-plugins/config/snmp.cfg

define command{
        command_name    snmp_ticks
        command_line    /usr/lib/nagios/plugins/check_snmp -H
'$HOSTADDRESS$' -C '$ARG1$' -o 1.3.6.1.2.1.1.8.0 -P 2c
        }
```

Exercise 2 walk-through

- Then add the service to the Nagios monitoring:

```
sudo nano /etc/nagios3/conf.d/services_nagios2.cfg

define service {
  hostgroup_name                  ssh-servers
  service_description             TimeTicks
  check_command                   snmp_ticks!recorded/linksys-system
  use                             generic-service
  notification_interval           0 ; set > 0 if you want to be renotified
}
```

- Check that the Nagios config doesn't contain errors

```
sudo nagios3 -v /etc/nagios3/nagios.cfg
```

- And restart Nagios

```
sudo service nagios3 start
```

Options for creating snmp users

- noAuthNoPriv
  - No authorisation and no encryption, basically no security at all!
- authNoPriv
  - Authorisation is required but collected data sent over the network is not encrypted.
- authPriv
  - The strongest form. Authorisation required and everything sent over the network is encrypted.

Installing snmp-agent on ubuntu

```
sudo apt-get install snmpd

sudo mv /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.org

sudo nano /etc/snmp/snmpd.conf
```

- Add the following to the file

```
#
createUser user1
createUser user2 MD5 user2password
createUser user3 MD5 user3password DES user3encryption
#
rouser user1 noauth 1.3.6.1.2.1.1
rouser user2 auth 1.3.6.1.2.1
rwuser user3 priv 1.3.6.1.2.1
```

Installing snmp-agent on ubuntu

```
sudo nano /etc/default/snmpd
```

- Comment out the following line, by adding # to it

```
#SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux,mteTrigger,mteTriggerConf -p
/run/snmpd.pid'
```

- Add the following line to the file:

```
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux,mteTrigger,mteTriggerConf -p
/run/snmpd.pid -c /etc/snmp/snmpd.conf‘
```

- Then you restart the service

```
sudo service snmpd restart
```

Walking the Ubuntu agent

- Now walk the snmp with the created user (this is 1 line)

```
snmpwalk -v3 -l authPriv -u user3 -a MD5 -A "user3password" -x DES -X "user3encryption"
localhost
```

- Install the mibs for Ubuntu to make it more readable.

```
sudo apt-get install snmp-mibs-downloader
sudo download-mibs
```

```
sudo nano /etc/snmp/snmp.conf
```

- Enable using the mibs by changing the line

```
mibs :
```

- to

```
# mibs :
```

Walking the Ubuntu agent

- Now walk the snmp with the created user again (this is 1 line)

```
snmpwalk -v3 -l authPriv -u user3 -a MD5 -A "user3password" -x DES -X
"user3encryption" localhost
```

- You can see the installed MIB structure by running

```
snmptranslate -Tp
```

Exercise

- Now select a value you would like to monitor and add it to Nagios.
- Have a look at the guides from last time, but this time do it using SNMP v3

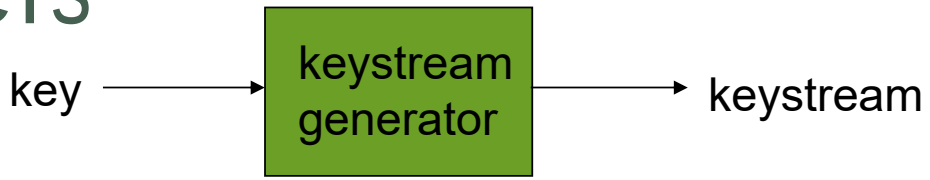
Wifi Security

WEP design goals

- symmetric key crypto
  - confidentiality
  - end host authorization
  - data integrity

- self-synchronizing: each packet separately encrypted
  - given encrypted packet and key, can decrypt; can continue to decrypt packets when preceding packet was lost (unlike Cipher Block Chaining (CBC) in block ciphers)
- Efficient
  - implementable in hardware or software

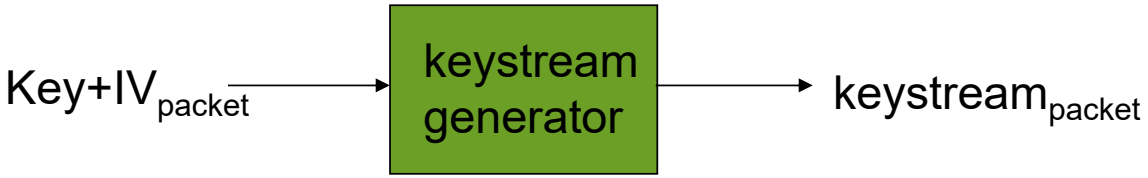
Review: symmetric stream ciphers



- *combine each byte of keystream with byte of plaintext to get ciphertext:*
  - m(i) = ith unit of message
  - ks(i) = ith unit of keystream
  - c(i) = ith unit of ciphertext
  - c(i) = ks(i) ⊕ m(i) (⊕ = exclusive or)
  - m(i) = ks(i) ⊕ c(i)
- WEP uses RC4

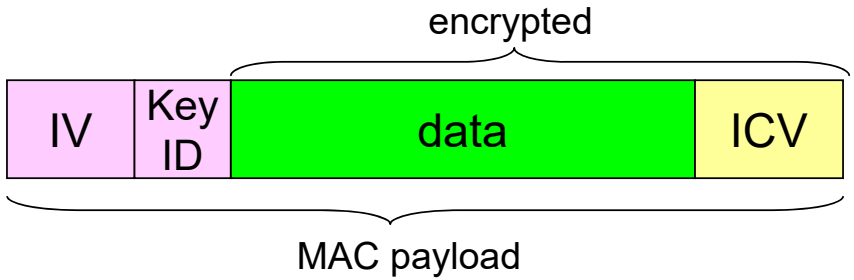
Stream cipher and packet independence

- recall design goal: each packet separately encrypted
- if for frame n+1, use keystream from where we left off for frame n, then each frame is not separately encrypted
  - need to know where we left off for packet n
- WEP approach: initialize keystream with key + new IV for each packet:

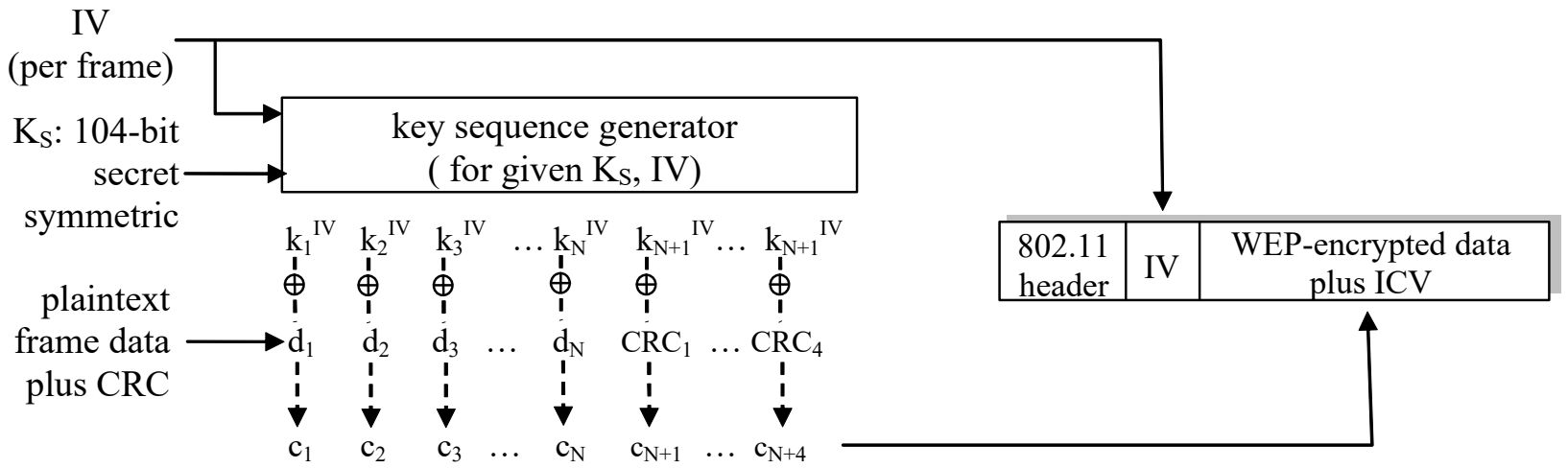


WEP encryption (1)

- sender calculates Integrity Check Value (ICV) over data
  - four-byte hash/CRC for data integrity
- each side has 104-bit shared key
- sender creates 24-bit initialization vector (IV), appends to key: gives 128-bit key
- sender also appends keyID (in 8-bit field)
- 128-bit key inputted into pseudo random number generator to get keystream
- data in frame + ICV is encrypted with RC4:
  - B\bytes of keystream are XORed with bytes of data & ICV
  - IV & keyID are appended to encrypted data to create payload
  - payload inserted into 802.11 frame

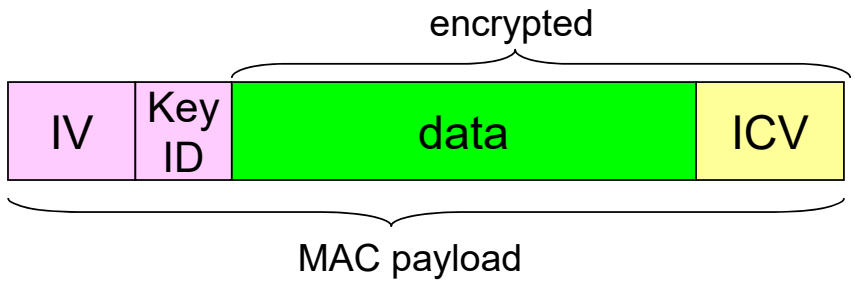


WEP encryption (2)



new IV for each frame

WEP decryption overview

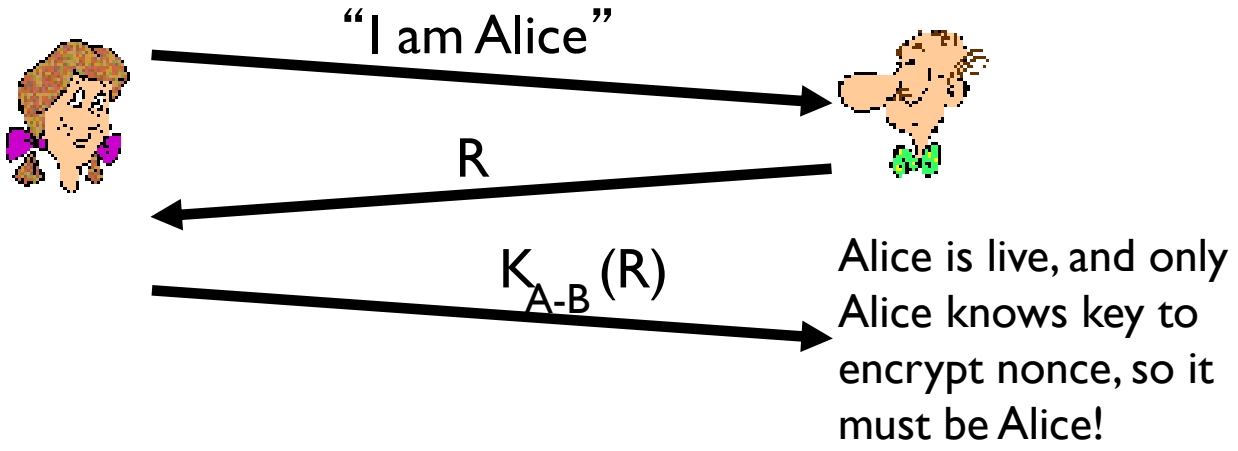


- receiver extracts IV
- inputs IV, shared secret key into pseudo random generator, gets keystream
- XORs keystream with encrypted data to decrypt data + ICV
- verifies integrity of data with ICV
  - note: message integrity approach used here is different from MAC (message authentication code) and signatures (using PKI).

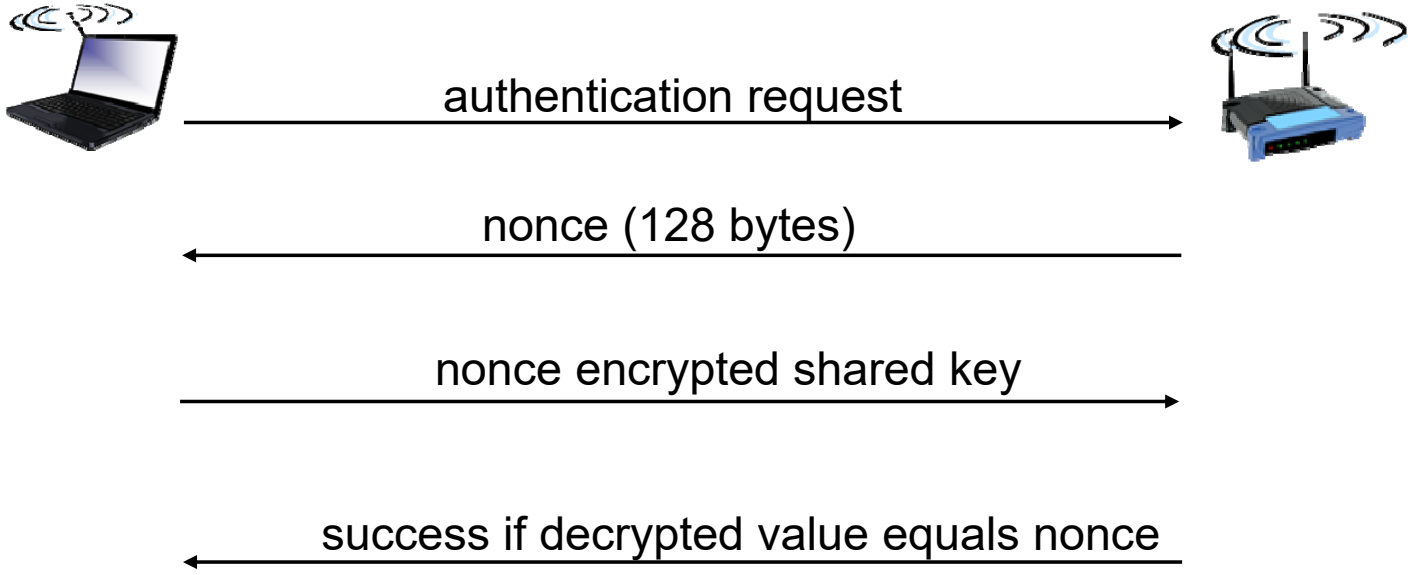
End-point authentication w/ nonce

Nonce: number (R) used only *once –in-a-lifetime*

How to prove Alice "live": Bob sends Alice *nonce*, R. Alice must return R, encrypted with shared secret key



WEP authentication



Notes:

- not all APs do it, even if WEP is being used
- AP indicates if authentication is necessary in beacon frame
- done before association

Breaking 802.11 WEP encryption

security hole:

- 24-bit IV, one IV per frame, -> IV' s eventually reused
- IV transmitted in plaintext -> IV reuse detected

attack:

- Trudy causes Alice to encrypt known plaintext $d_1\ d_2\ d_3\ d_4$ …
- Trudy sees: $c_i = d_i$ XOR k_i^{IV}
- Trudy knows $c_i\ d_i$, so can compute k_i^{IV}
- Trudy knows encrypting key sequence $k_1^{IV}\ k_2^{IV}\ k_3^{IV}$ …
- Next time IV is used, Trudy can decrypt!

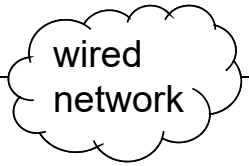
802.11i: improved security

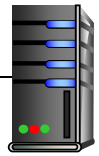
- numerous (stronger) forms of encryption possible
- provides key distribution
- uses authentication server separate from access point

802.11i: four phases of operation

STA: client station

AP: access point





AS: Authentication server

1 Discovery of security capabilities

2 STA and AS mutually authenticate, together generate Master Key (MK). *AP serves as "pass through"*

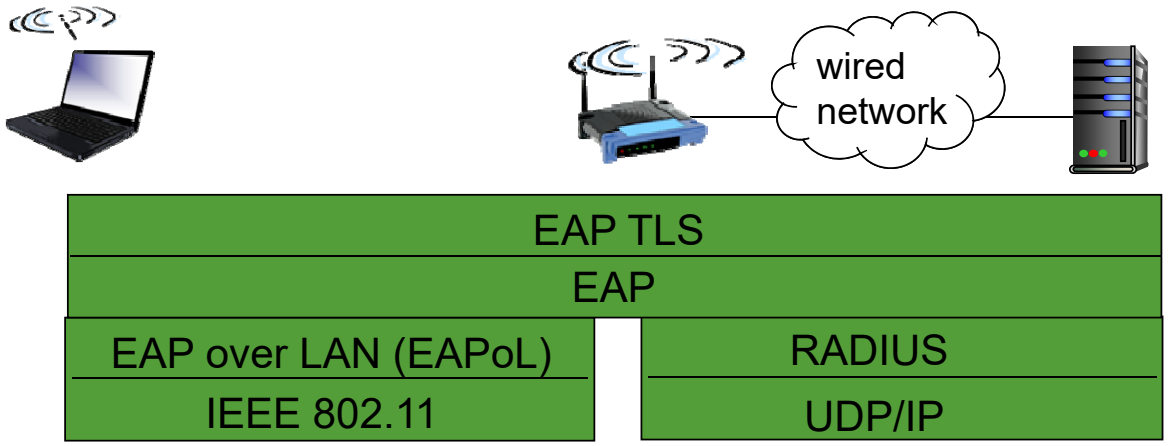
3 STA derives Pairwise Master Key (PMK)

3 AS derives same PMK, sends to AP

4 STA, AP use PMK to derive Temporal Key (TK) used for message encryption, integrity

EAP: extensible authentication protocol

- EAP: end-end client (mobile) to authentication server protocol
- EAP sent over separate "links"
  - mobile-to-AP (EAP over LAN)
  - AP to authentication server (RADIUS over UDP)



Ways to attack Wifi

- Create rogue APs
- Disassociate users from the real AP
- Attacks the encryption

Hirte Attack agains WEP

- Hosts actively scans availiable networks
- If we create a rogue AP with the same SSID, then the client will automatically try to associate to it
- Try it using this guide
  - https://pentestlab.blog/tag/airbase-ng/

Exam

- Draw a question with no preparation. Question covers a topic
- Try to discuss the topic, and use practical examples
- Exam is 30 minutes in total, including pulling the question and grading
- Count on being able to present talk for about 10 minutes
- Prepare material (keywords, examples, exercises, wireshark captures) for different topics so that you can use it to help you at the exam

Topics

- Packet injection (scapy)
- Network scanning
- ARP (poisoning)
- DNS spoofing
- HTTPS (sslstrip)
- NETFLOW
- IDS/IPS
- Network Architecture
- VPN (ipsec, openvpn)
- NET management (snmp v2, v3 nagios)