Security on the air: 802.11

Guido Bolognesi
guido@kill-9.it

Presentazione

Relatore

Partecipanti

Domanda: wifi?

Domanda: wifi security?

Il punto di vista Internet

Una ricerca su Google per “wifi” restituisce

3.240.000 hit

Una ricerca per “wi-fi security”...

1.360.000 hit

Perché wifi è insicuro?

Wifi = ethernet + radio

Quindi

Problemi wifi = problemi network

+

Problemi radio

Perché wifi è insicuro?

+

Problemi della “colla” implementativa

Possibili attacchi

Layer 1: DOS radio

Algoritmi e meccanismi di
- Autenticazione
- Cifratura
- Integrità

Alcuni sistemi di protezione

- WEP
- EAP – TLS
- EAP – TTLS
- LEAP
- PEAP
- WPA

Alcuni sistemi di protezione

- WEP
- EAP – TLS
- EAP – TTLS
- LEAP
- PEAP
- WPA

Tipologie di attacco

Gli attacchi possono essere di due tipi:

- passivi (sniffing puro)
- attivi (association request, traffic injection, ...)

Discovery degli AP

Strumenti passivi:

- kismet
- Airsnort
- wellenreiter

Strumenti attivi: netstumbler

Attrezzatura necessaria

Cosa mi serve:

- Un **laptop/PDA**
- un **sistema operativo** moderno
- una **scheda** (che possibilmente vada in monitor mode su tutti i canali)
- una **antenna**
- [un **inverter**]

Cifratura

block **vs** stream **cipher**

Block

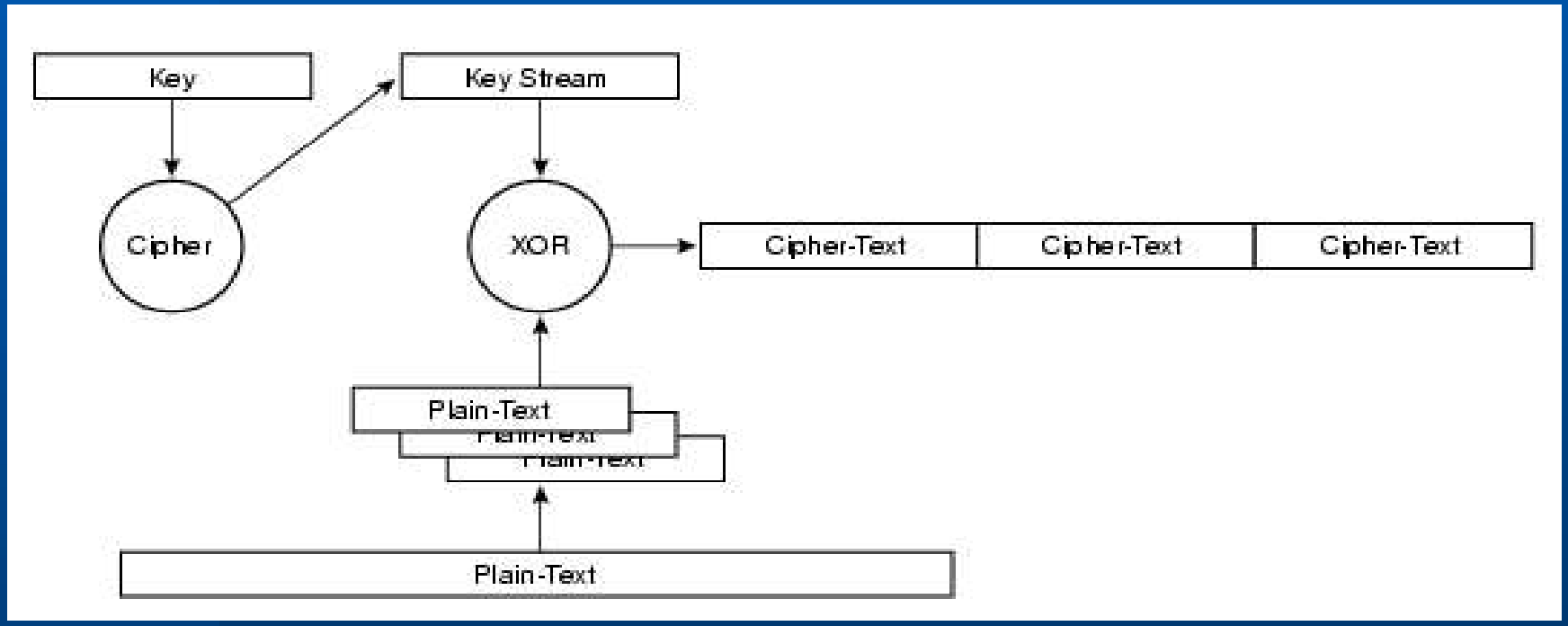
- dovendo crittare con un algoritmo che generi blocchi da 16 un frame da 34

16 + 16 + 2

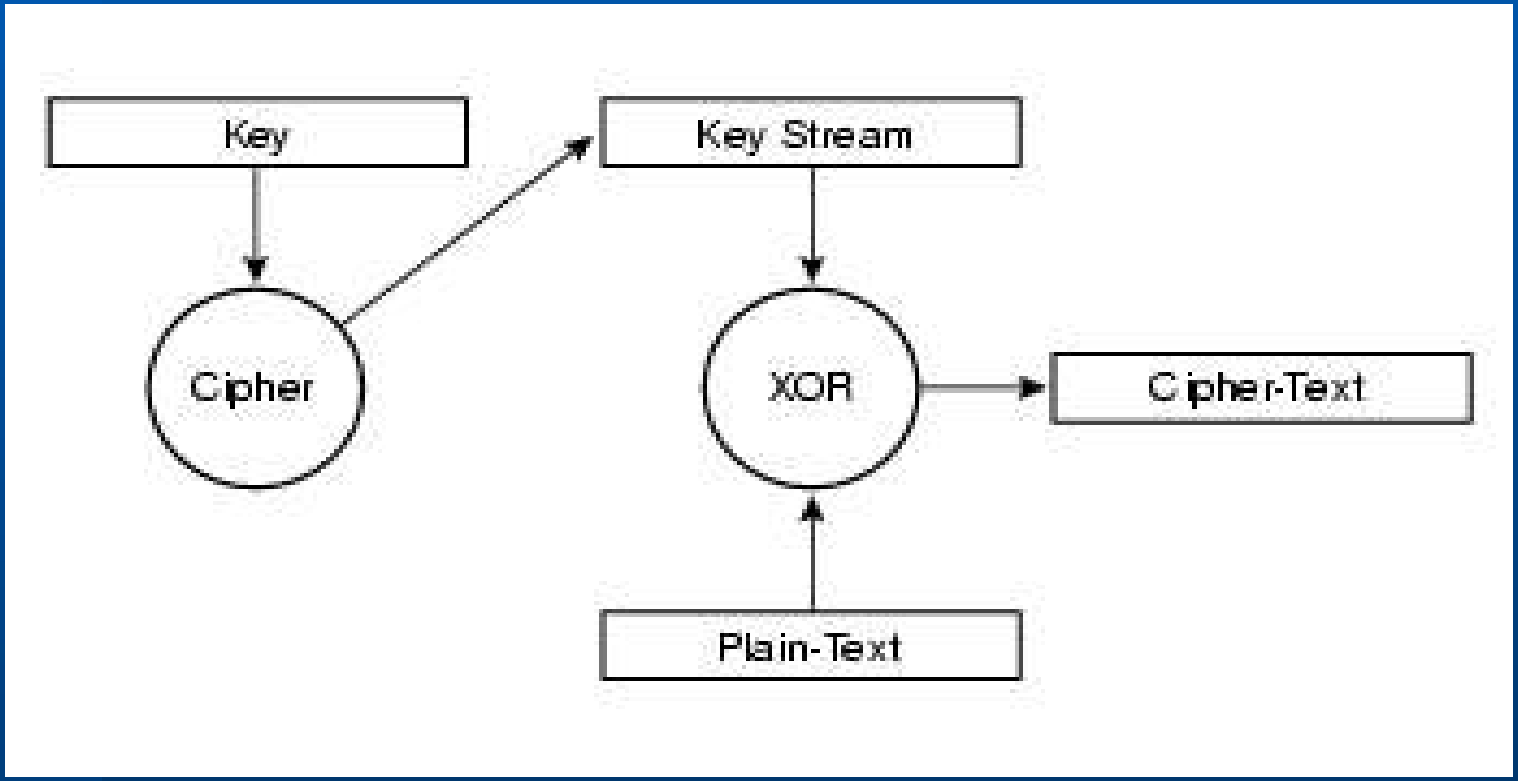
Stream

- Genera un flusso continuo, adatto quindi a ogni tipo di dato

Block cipher



Stream cipher



Come cifro?

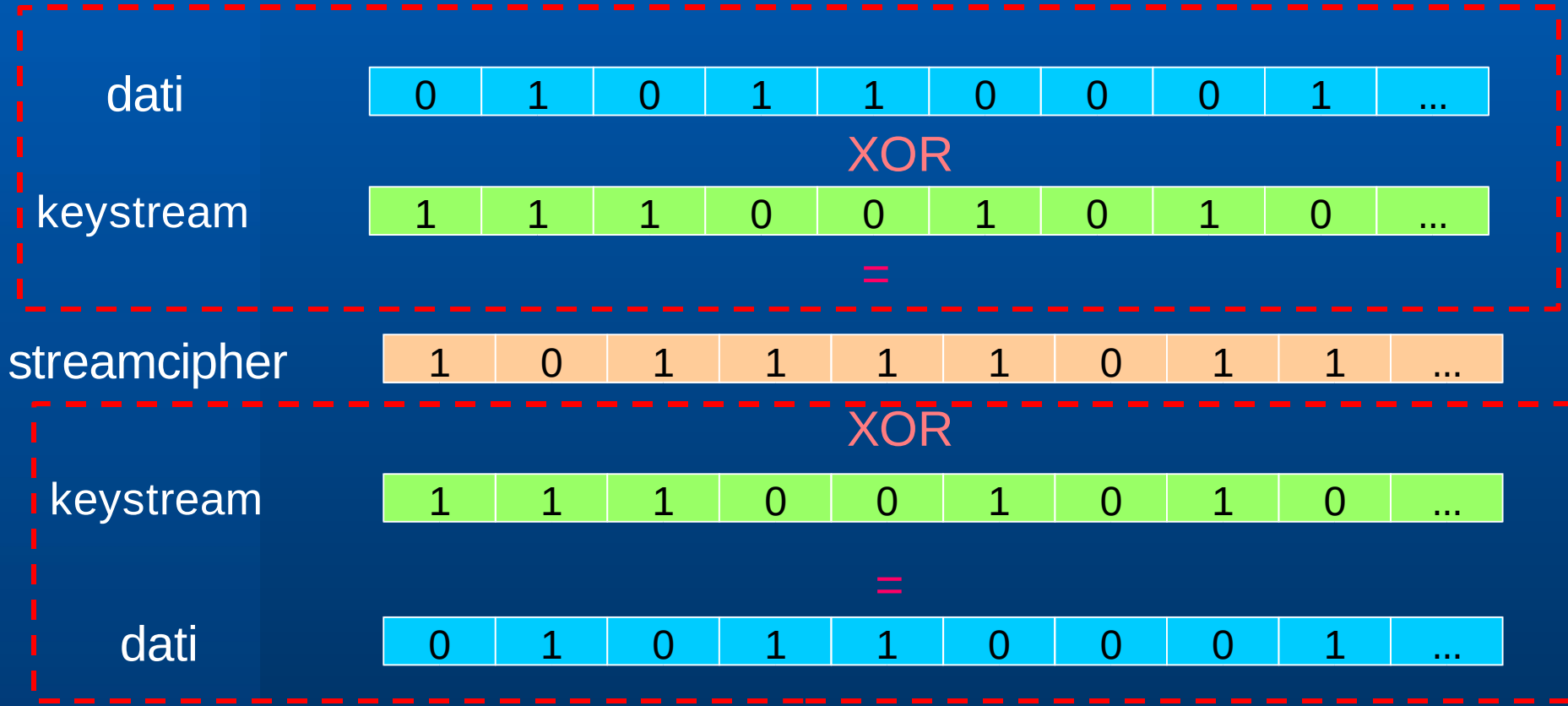
Basta combinare il testo in chiaro con una “portante” cifrata

WEP utilizza RC4 per ottenere il ciphertext

keystream XOR plaintext = cipher

Meccanismo a chiave simmetrica

Come cifro? (2)


dati
0 1 0 1 1 0 0 0 1 ...
XOR
keystream
1 1 1 0 0 1 0 1 0 ...
=
streamcipher
1 0 1 1 1 1 0 1 1 ...
XOR
keystream
1 1 1 0 0 1 0 1 0 ...
=
dati
0 1 0 1 1 0 0 0 1 ...

Cracking WEP

XOR significa che se conosco lo stream posso ottenere il payload.

MA ANCHE IL CONTRARIO

802.11 ha un LLC, i frame hanno un header SNAP fisso (0xAA): ho il primo byte del keystream.

Cracking WEP

Il ricevente non ha certezza dell'integrità del frame, dato che viene calcolata solo tramite CRC32

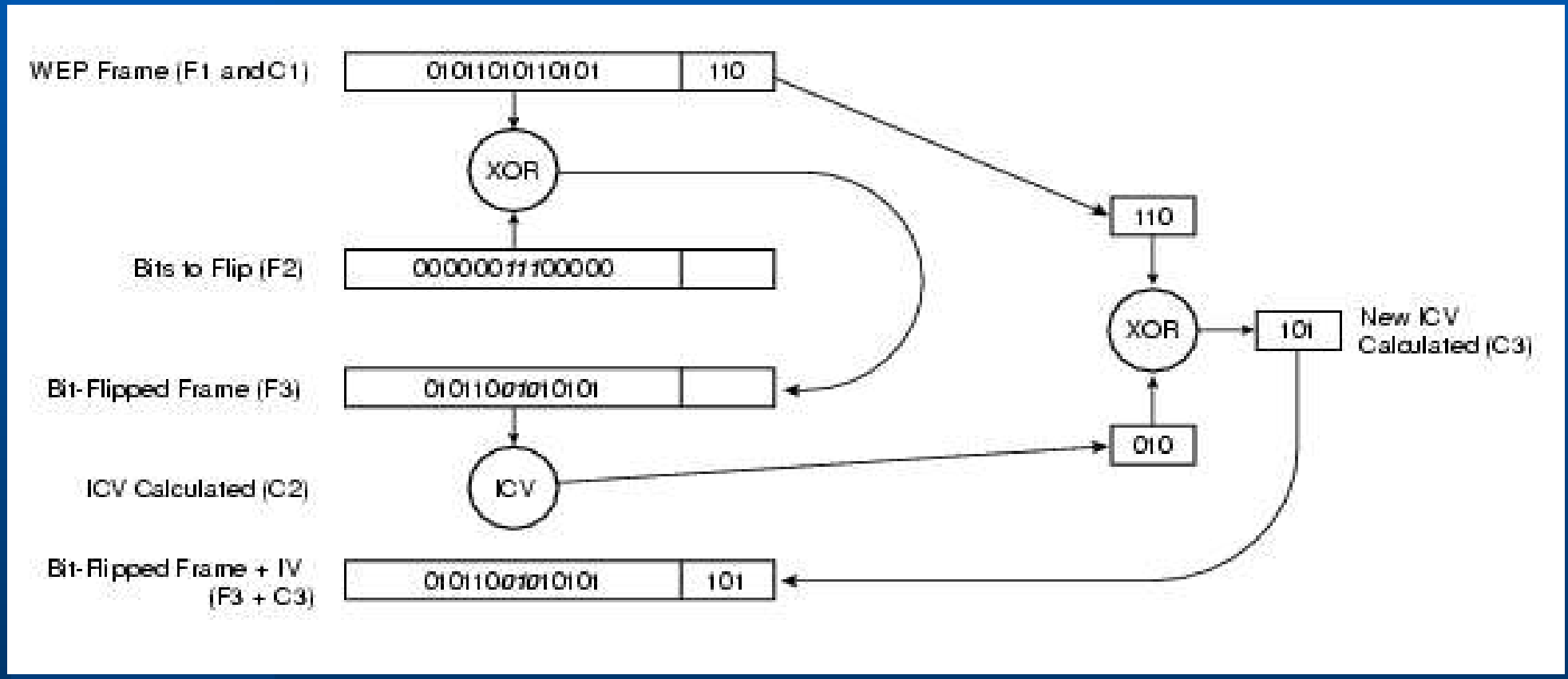
Attacchi attivi:

- Bit flipping
- Replay attack
- Corruzione frame layer 2

Cracking WEP: bitflip

- catturo un frame cifrato
- flippo bit random nel payload e rigenero il CRC32
- lo mando, viene decapsulato correttamente (CRC32 corretto)
- genera un errore a layer3
- cerco il messaggio di errore (conosciuto) e derivo il keystream

Cracking WEP: bitflip



Cracking WEP: replay attack

- testo in chiaro conosciuto mandato sulla wifi (ARP, ICMP, STP...)
- sniff in cerca del ciphertext
- una volta trovato, derivo il keystream
- Riutilizzo la wep key + IV per generare un altro ciphertext

Cracking WEP: replay attack

ho il keystream

10 genero un pacchetto più grande di 1 byte del keystream che ho (ad es. ICMP)

20 lo inietto in rete finché non ottengo risposta (è solo 1 byte = 256 tentativi)

GOTO 10

Cracking WEP: reinjection

- reinj.c
- Accelera il traffico a livello 2 reiniettando frame ARP o TCP SYN
- Diventa possibile recuperare la chiave WEP in meno di 60 minuti

Cracking WEP: altri problemi

- Nessuna gestione dinamica delle chiavi, difficoltà di manutenzione
- Autenticazione monodirezionale: l'AP viene considerato sicuro
  - Chi mi sta dando accesso alla rete?
  - Di chi è il MAC address?
  - Chi mi sta autenticando?

Monkey_Jack

- L’attaccante lancia un attacco DOS
- La scheda della vittima cerca un nuovo AP
- La vittima si associa ad un AP “finto” sulla macchina dell’attaccante
- La macchina dell’attaccante si associa con il vero AP
- Ora la macchina dell’attaccante può passare i frame tra il client e l’AP

Monkey_Jack

```
#./monkey_jack
Monkey Jack: Wireless 802.11(b) MITM proof of concept.

Usage: ./monkey_jack -b <bssid> -v <victim mac> -C <channel number> [ -c <channel number> ]
        [ -i <interface name> ] [ -I <interface name> ] [ -e <essid> ]

       -a:  number of disassociation frames to send (defaults to 7)
       -t:  number of deauthentication frames to send (defaults to 0)
       -b:  bssid, the mac address of the access point (e.g. 00:de:ad:be:ef:00)
       -v:  victim mac address.
       -c:  channel number (1-14) that the access point is on, defaults to current.
       -C:  channel number (1-14) that we're going to move them to.
       -i:  the name of the AirJack interface to use (defaults to aj0).
       -I:  the name of the interface to use (defaults to eth1).
       -e:  the essid of the AP.

#./monkey_jack -b 00:40:96:5b:37:af -v 00:07:85:92:db:a9 -c 1 -C 8 -i aj0 -I eth1 -e "l3p3r0us"
Starting Monkey in the Middle Attack:

victim: 00:07:85:92:db:a9
bssid:  00:40:96:5b:37:af

configuring airjack device...done.
forcing ourselves in the middle...done.
configuring lucent card...done.
coercing our card to associate as the victim...done.

layer 1 insertion complete.
```

WEP: sommario

NON
UTILIZZATE
WEP

Anche se onestamente ci sono implementazioni moderne fatte meglio

802.1x + EAP = 802.11i

Framework di autenticazione

Composto da 3 elementi

- **supplicant** - sul client wifi
- **authenticator** - sull'access point
- **authentication server** - tipicamente un server RADIUS

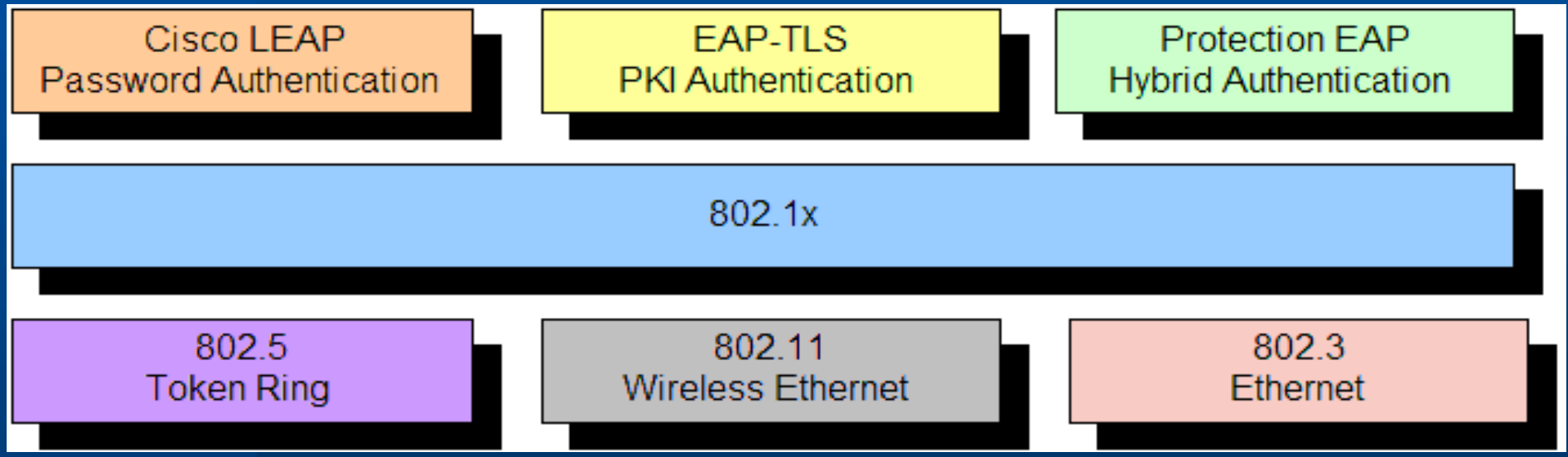
EAP

Extensible authentication Protocol

- EAP-TLS - Transport Layer Security (RFC2716) successore SSLv3 – ma non autentica l'AP
- EAP-SIM - Subscriber Identity Module
- LEAP - soluzione Cisco proprietaria
- PEAP - Protected EAP un tunnel TLS per autenticare il server, dopo di che un altro per autenticare il client

EAP

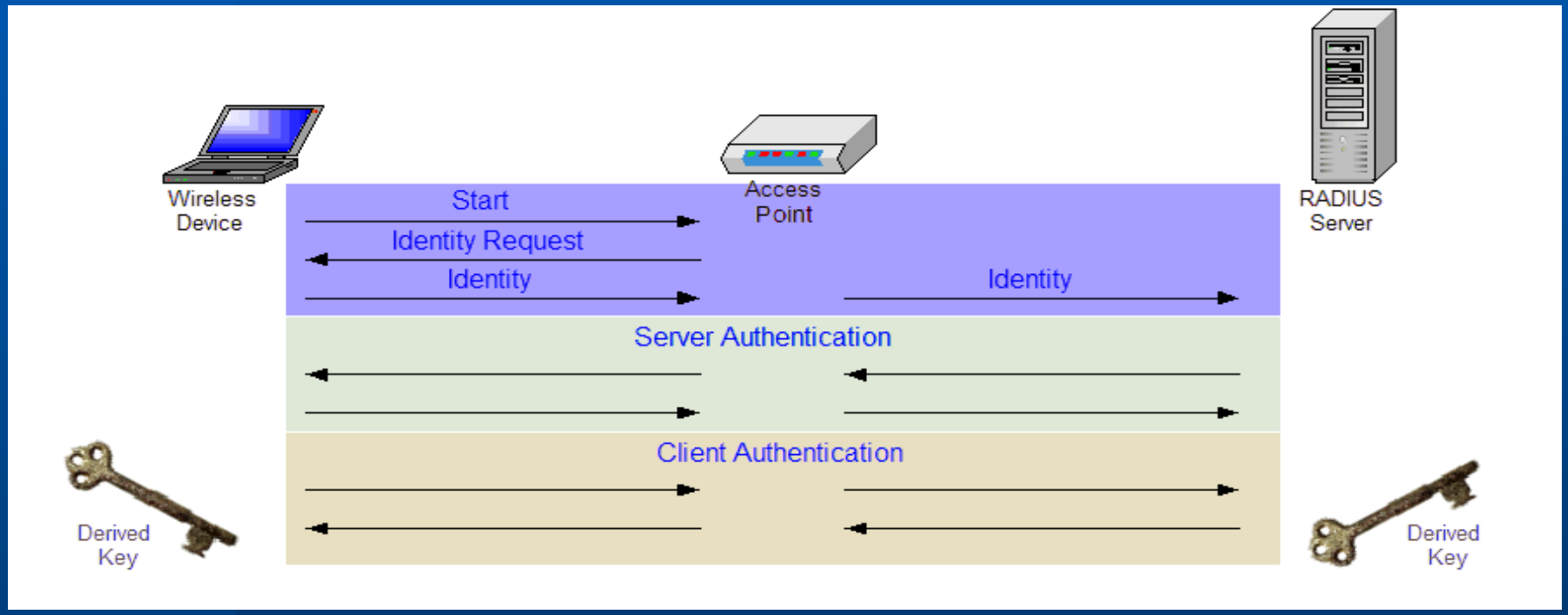
802.1x fa autenticazione *port-based*



Dinamica generale

- il client tenta il collegamento all'AP
- L’AP chiede autenticazione
- Il client la fornisce -> AP -> forward all'authentication server
- [radius autentica il client]
- [client autentica il server]
- AP manda la chiave al client cifrata con la chiave di sessione.

Dinamica generale



Vantaggi

- Gestione dinamica delle chiavi
- Mutua autenticazione
- Nessun accesso alla rete fino ad autenticazione completa
- Autenticazione dell’ **utente**, non dell’**apparato**
- Cifratura ed integrità dei pacchetti migliore

Cracking LEAP

- LEAP utilizza un protocollo molto simile a MS-CHAPv2 per la parte di challenge/authentication
- Per autenticarsi su un server RADIUS utilizza la password dell’utente
- La password di LEAP viene data in pasto a RC4 per generare un hash da 16 byte

Cracking LEAP

- I 16 byte vengono portati a 21 aggiungendo **5 null (!!)**
- Il risultato viene diviso in 3 parti da 7 byte (stessa lunghezza di una chiave DES56)
- Ogni parte viene utilizzata per cifrare un challenge e unita alle altre (ma **senza fare CBC**)

Cracking LEAP

Ora siamo in questa situazione:

Siano A, B, C le parti da 7 byte

$F_a(\text{challenge})+F_b(\text{challenge})+F_c(\text{challenge})$

Ma $F_c(\text{challenge})$ ha solamente

7-5 = **2 byte variabili!**

Quindi solamente 2^{16} combinazioni, ~65k

Lo spazio delle chiavi viene ridotto a $1/2^{16}$ del possibile

Cracking LEAP

- Questo tipo di attacco dipende ovviamente dalla correlazione tra gli ultimi due byte della password e ciò che li precede
- Cisco document id: 44281
- Rilascio del tool: Aprile 2004 (6 mesi dopo la disclosure)
- Attenzione: **rilascio**, non **scrittura…**

Cracking WPA? 4/11/03

Wireless Protected Access

Per-packet keying, Temporal Key Integrity Protocol…

- WPA + 802.1x
- WPA + chiavi statiche (PSK)
  - Numero 256bit
  - Passphrase 8-16 byte

Robert Moskowitz – ICSA Security Lab

http://wifinetnews.com/archives/002452.html

Cracking WPA? 4/11/03

Dallo standard 802.11i:

"A passphrase typically has about 2.5 bits of security per character, so the passphrase of n bytes equates to a key with about 2.5n + 12 bits of security. Hence, it provides a relatively low level of security, with keys generated from short passwords subject to dictionary attack. Use of the key hash is recommended only where it is impractical to make use of a stronger form of user authentication. **A key generated from a passphrase of less than about 20 characters is unlikely to deter attacks**."

Cracking WPA? 4/11/03

- Pochi utenti sono disposti ad usare passphase da 20 o più caratteri
- Offline dictionary attack sulle preshared già fattibile
- **Occorre sempre un utilizzo cosciente e responsabile delle tecnologie**

Altri tool interessanti

AirSnarf, per gli hotspot

- HostAP+Dns hijacking+Web Server = sniff the hotspot
- http://airsnarf.shmoo.com/

Fake AP

- genera milioni di pacchetti con SSID randomici, chiavi wep randomiche, mac address randomici
- blocca il discovery di reti da parte di client (richiede impostazione manuale di ssid)
- http://www.blackalchemy.to/Projects/fakeap/fake-ap.html

Ulteriori attacchi

AP MAC-address spoof

Flood di messaggi di disassociazione – deautenticazione (tutte le stazioni si devono riautenticare/riassociare)

... usate la fantasia!

Grazie

Fabio “naif” Pietrosanti

Yvette “vodka” Agostini

Kay Sommers

Shon Harris

Cisco.com

KoAn

La comunità dell’ OpenSource