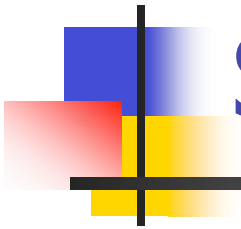
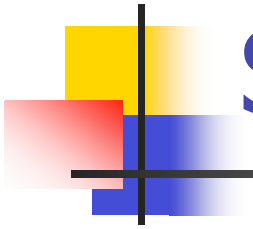


Sicurezza su Reti Wi-Fi

Mario Marcelli

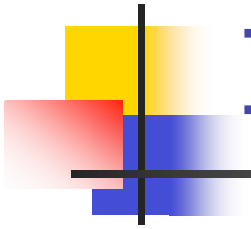
Fabio Faruoli

Claudio Bortone



Sommario

- Introduzione
- Crittografazione del traffico
- Sistemi di autenticazione
- Crittoanalisi del WEP



Introduzione

- La parola inglese “wireless”, letteralmente “senza cavi”, è oggi utilizzata per indicare tutto l’insieme di tecnologie che permettono l’interconnetività di sistemi senza l’utilizzo di cablaggi.
- Da questo punto di vista i nostri stessi telefoni cellulari sono implementazioni “wireless”, sebbene tale termine sia più comunemente utilizzato nel campo più strettamente informatico.

Institute of Electrical and Electronical Engeneers

IEEE

- Standard relativi alle reti *wired* (cablate)
- Dal 1990 ha creato la commissione **802.11** che ha dedicato sette anni di lavoro allo standard per le reti locali in radiofrequenza.

26 giugno 1997 - approvato standard IEEE 802.11 ed è il primo standard internazionale riconosciuto per WLAN (*Wireless* LAN).

Gli Standard Wireless

✓1997- IEEE**802.11** → velocità di 1-2Mbps

↓ Successo e interesse

✓1999- IEEE**802.11a** → 54Mbps – Frequenza 5GHz

↓

✓1999- IEEE**802.11b** → 11Mbps – Frequenza 2,4GHz

802.11b=STANDARD DOMINANTE

802.11b=Wi-Fi (Wireless Fidelity)

↓

Wi Fi™
The Standard for Wireless Fidelity.

✓2003– **802.11g** → 54Mbps – Frequenza 2,4GHz

↓

✓2004– **802.11i** → 54Mbps – Frequenza 5GHz

Dispositivi Wireless

In generale le architetture per sistemi wireless sono basate due tipologie di dispositivi:

- **Access Point (AP);**
- **Wireless Terminal (WT).**

Access Point = *bridge* che collega la sottorete *wireless* con quella cablata.

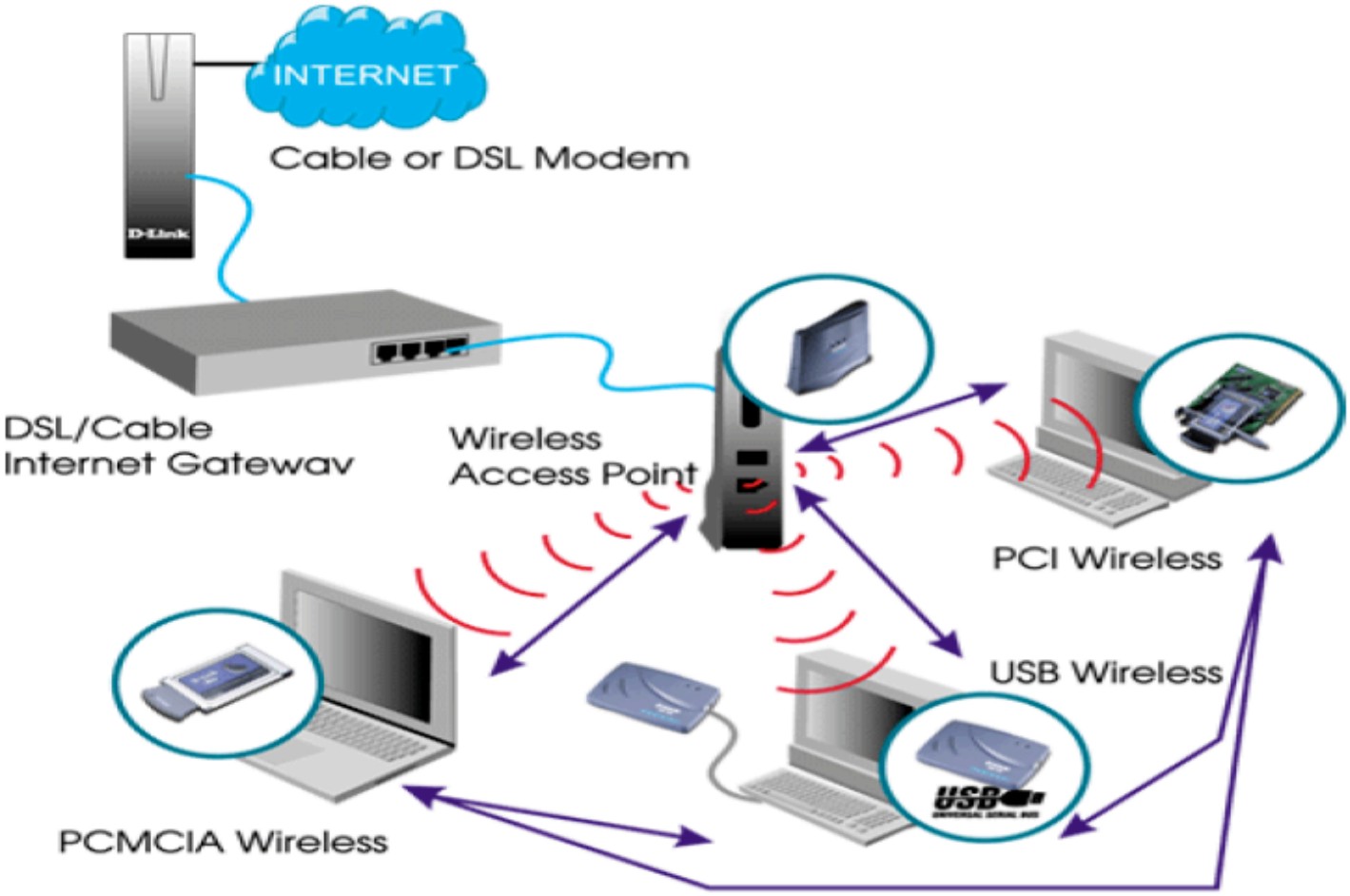
Tablet PC

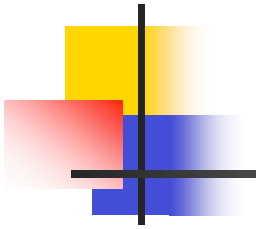
PC portatile

PDA

Smartphone

Scenario di una WLAN

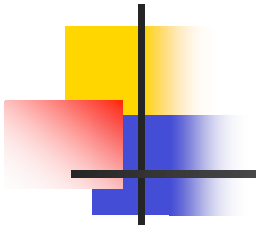




Vantaggi

I vantaggi di questa tecnologia possono essere riassunti come segue:

- **di facile configurazione**
- **di facile installazione**
- **alta mobilità**
- **utile per accrescere la collaborazione tra gli utenti**

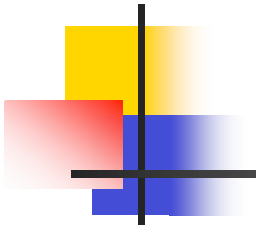


Autenticazione

Attualmente lo standard 802.11b per effettuare l'autenticazione prevede due modalità:

A. **OSA** (Open Systems Authentication), meno sicuro poichè non prevede nessun metodo di autenticazione, pertanto l'accesso è garantito a tutti

B. **SKA** (Shared Key Authentication), prevede invece l'utilizzo di chiavi condivise, rendendo pertanto il sistema di accesso più sicuro.



WEP

WEP (*Wired Equivalent Privacy*) scelto dal comitato IEEE 802.11 come standard per la cifratura:

- algoritmo di cifratura RC4
- chiavi condivise (shared key) di lunghezza variabile da 40 a 104 bit
- soluzioni proprietarie fino a 256 bit
- anche per l'autenticazione con l'AP

Obiettivi del WEP

Autenticazione

- Riconoscimento dell'autore e non ripudio del messaggio

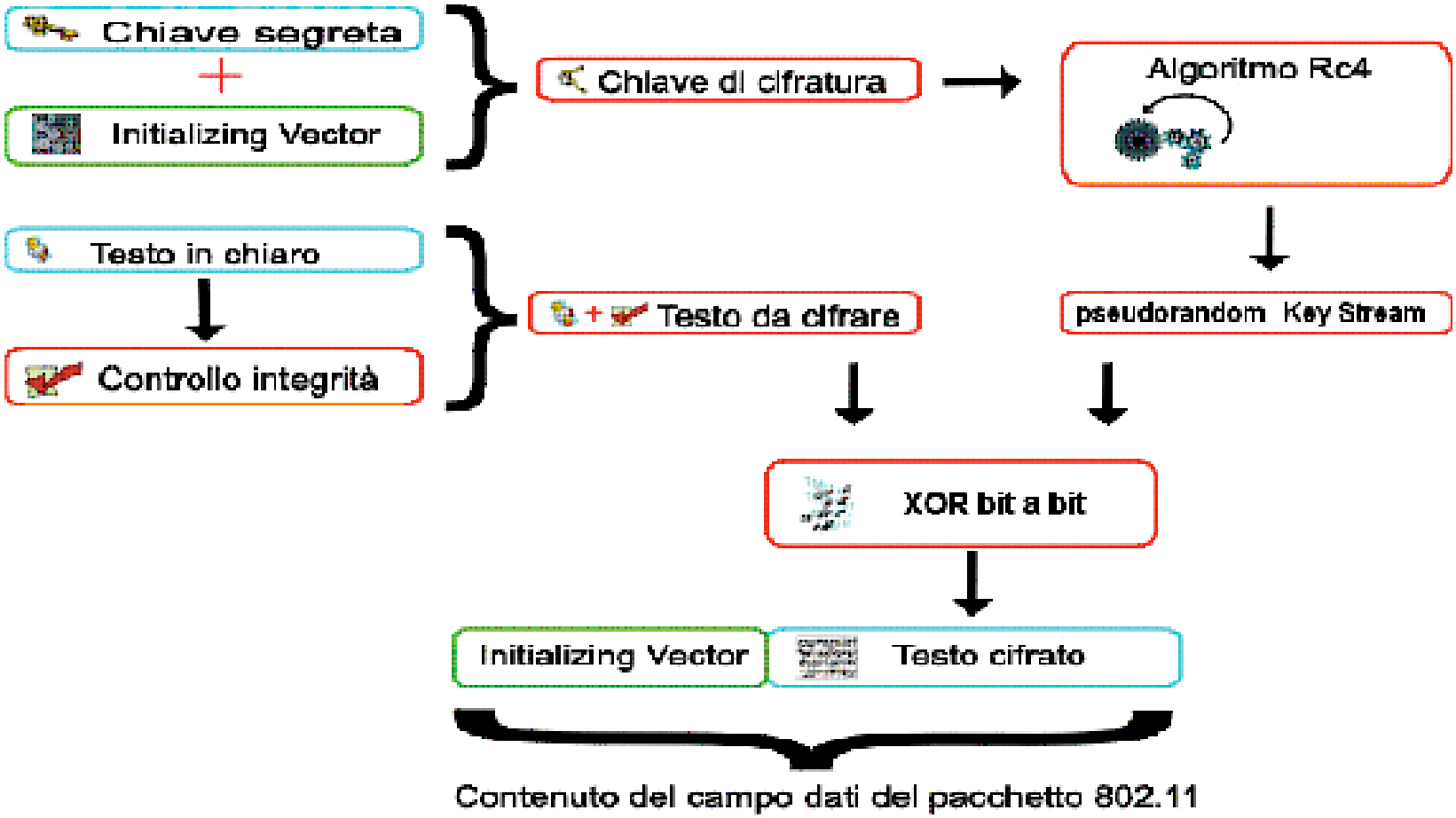
Confidenza (riservatezza)

- I dati devono protetti dall'intercettazione di persone non autorizzate

Integrità

- I dati non devono essere modificati

Funzionamento di WEP



Creazione della chiave WEP

La password da utilizzare è immessa al momento della configurazione degli apparati.

Si possono utilizzare sequenze alfanumeriche di 5 o 13 caratteri a seconda della chiave utilizzabile:

5 x 8 = 40 bit

13 x 8 = 104 bit

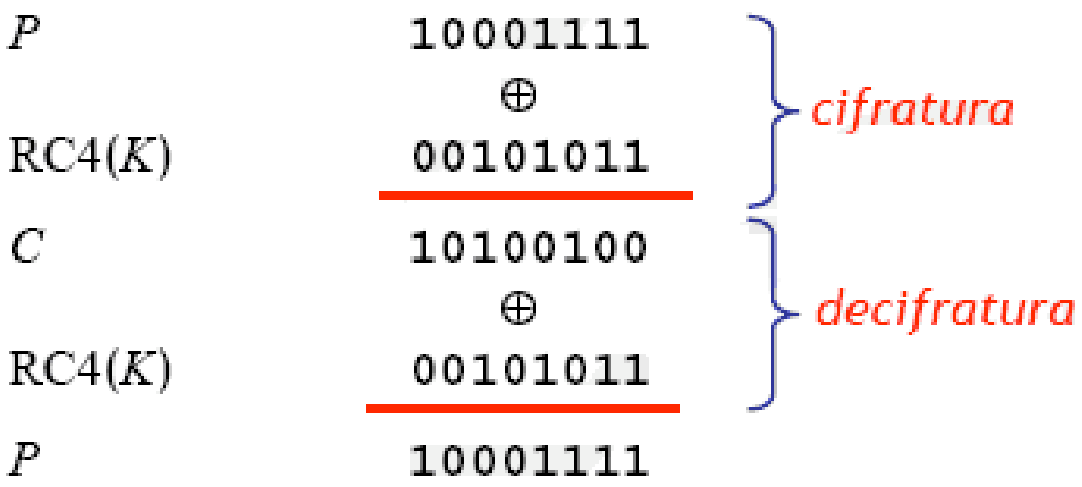
Es: AA BB CC DD EE FF AA BB CC DD EE FF 00

Algoritmo RC4

RC4 è un cifrario a flusso progettato da Ron Rivest (la "R" di RSA) nel 1987.

Era un segreto commerciale della RSA Security, ma nel 1994 è stato inviato in maniera anonima a una mailing list su Internet, e da allora è stato ampiamente analizzato.

A partire da una chiave (lunga da 1 a 256 ottetti), genera una sequenza pseudocasuale (keystream) utilizzata per cifrare e decifrare (mediante XOR) un flusso dati.



Algoritmo a due fasi

L'algoritmo RC4 è composto da due fasi:

- Key Scheduling
  - Inizializza i fattori generativi della chiave. Genera una permutazione che appare casuale della shared key; la permutazione è ciclica con un periodo molto lungo (più di 10^{100}).
- Pseudo Random Generation
  - Genera la chiave (keystream)

Key Scheduling Algorithm (KSA)

- in *S* si inseriscono i valori da 0 a 255: *S*[*n*] = *n*;
- in un altro vettore temporaneo *K* (di 256 ottetti) si inserisce la chiave (ripetendola se più corta);
- i contatori *i* e *j* si inizializzano a zero;
- si percorre *S* scambiando (*scrambling*) l'elemento corrente (*i*-esimo) con un altro determinato usando la chiave:
  - Inizializzazione
    - i=0, j=0;
    - For i=0 to N-1
      - S[i]=i;
  - Scrambling (Mischiare)
    - For i=0 to N-1
      - j=(j+S[i]+K[i]) mod L; /* L = sizeof(K) */
      - SWAP(S[i],S[j]);

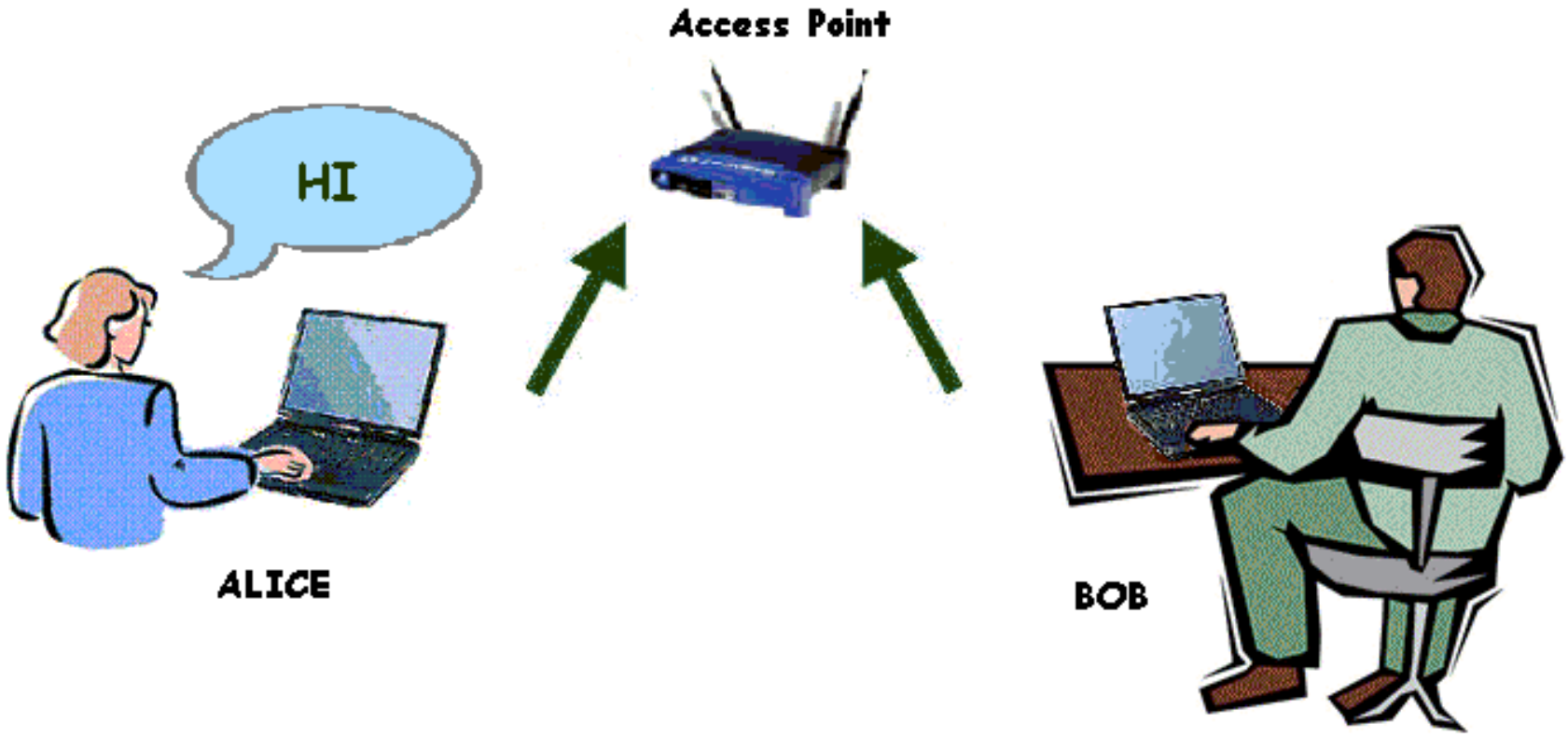
Pseudo Random Generation Algorithm (PRGA)

- Si reinizializzano *i* e *j* a zero e si scarta *K* (la chiave non viene più utilizzata).i=0, j=0;

- Si percorre il vettore *S*, scambiando l'elemento corrente (*i*-esimo) con un altro determinato dallo stato corrente di *S* e *j*. Per generare un ottetto *Z* del keystream, dallo stato corrente (*S*, *i*, *j*):
  - LOOP (pari al numero ottetti della Keystream):
    - i=(i+1)mod L; j=(j+S[i]) mod L;
    - SWAP(S[i], S[j]);
    - Output z = (S[(S[i]+S[j]) mod L]) mod L;

- Considerando *S*, *i*, *j*, RC4 può trovarsi in ben 256! _ 256^2 (circa 2^{1700}, o 5,62 _ 10^{511}) stati.

RC4: ESEMPIO(1)



RC4: ESEMPIO(2)

Codificare la parola "HI"

Password = "6152" (shared Key,
conosciuta sia da Alice che da Bob)

Lunghezza L = 4

- K[0]=6
- K[1]=1
- K[2]=5
- K[3]=2

1. Inizializzazione S[L];

S[0]=0, S[1]=1, S[2]=2, S[3]=3

```
Inizializzazione
i=0, j=0;
For i=0 to N-1
    S[i]=i;
```

RC4: ESEMPIO(3)

```
For i=0 to N-1
j=(j+S[i]+K[i]) mod L
SWAP(S[i],S[j])
```

2. SWAP S;

First Loop

Valori Iniziali:

S[0]=0, S[1]=1, S[2]=2, S[3]=3
K[0]=6, K[1]=1, K[2]=5, K[3]=2
i=0, j=0

Equazioni:

j=(j+S[0]+K[0]) mod 4 = (0+0+6) mod 4 = 2
SWAP(S[0],S[2]) = S[0]=2, S[2]=0

Valori Finali:

S[0]=2, S[1]=1, S[2]=0, S[3]=3
K[0]=6, K[1]=1, K[2]=5, K[3]=2
i=0, j=2

RC4: ESEMPIO(4)

2. SWAP S;

Second Loop

```
For i=0 to N-1
j=(j+S[i]+K[i]) mod L
SWAP(S[i],S[j])
```

Valori Iniziali:

S[0]=2, S[1]=1, S[2]=0, S[3]=3

K[0]=6, K[1]=1, K[2]=5, K[3]=2

i=1, j=2

Equazioni:

j=(j+S[1]+K[1]) mod 4 = (2+1+1) mod 4 = 0

SWAP(S[1],S[0]) = S[1]=2, S[0]=1

Valori Finali:

S[0]=1, S[1]=2, S[2]=0, S[3]=3

K[0]=6, K[1]=1, K[2]=5, K[3]=2

i=1, j=0

RC4: ESEMPIO(5)

2. SWAP S;

Third Loop

```
For i=0 to N-1
j=(j+S[i]+K[i]) mod L
SWAP(S[i],S[j])
```

Valori Iniziali

S[0]=1, S[1]=2, S[2]=0, S[3]=3

K[0]=6, K[1]=1, K[2]=5, K[3]=2

i=2, j=0

Equazioni:

j=(j+S[2]+K[2]) mod 4 = (0+0+5) mod 4 = 1

SWAP(S[2],S[1]) = S[2]=2, S[1]=0

Valori Finali:

S[0]=1, S[1]=0, S[2]=2, S[3]=3

K[0]=6, K[1]=1, K[2]=5, K[3]=2

i=2, j=1

RC4: ESEMPIO(6)

2. SWAP S;

Fourth Loop

```
For i=0 to N-1
j=(j+S[i]+K[i]) mod L
SWAP(S[i],S[j])
```

Valori Iniziali

S[0]=1, S[1]=0, S[2]=2, S[3]=3

K[0]=6, K[1]=1, K[2]=5, K[3]=2

i=3, j=1

Equazioni:

j=(j+S[3]+K[3]) mod 4 = (1+3+2) mod 4 = 2

SWAP(S[3],S[2]) = S[3]=2, S[2]=3

Valori Finali:

S[0]=1, S[1]=0, S[2]=3, S[3]=2

K[0]=6, K[1]=1, K[2]=5, K[3]=2

i=3, j=2

RC4: ESEMPIO(7)

3. PRGA;

Initialization

Valori Iniziali

i=0, j=0

LOOP:
i=(i+1)mod L; j=(j+S[i]) mod L
SWAP(S[i], S[j])
Output z = (S[(S[i]+S[j]) mod L]) mod L

First Loop

Valori Iniziali

S[0]=1, S[1]=0, S[2]=3, S[3]=2

i=0, j=0

Algoritmo:

i=0+1=1

j=0+S[1]=0+0=0

SWAP(S[1],S[0]) = S[1]=1, S[0]=0

Valori Finali:

S[0]=0, S[1]=1, S[2]=3, S[3]=3

i=1, j=0

z1 = S[S[0]+S[1]]=S[0+1]=1 = 00000001

RC4: ESEMPIO(8)

3. PRGA;

```
LOOP:
i=(i+1)mod L; j=(j+S[i]) mod L
SWAP(S[i], S[j])
Output z = (S[(S[i]+S[j] ) mod L]) mod L
```

Second Loop

Valori Iniziali

S[0]=0, S[1]=1, S[2]=3, S[3]=2

i=1, j=0

Algoritmo:

i=1+1=2

j=0+S[2]=0+3=3

SWAP(S[2],S[3]) = S[2]=2, S[3]=3

Valori Finali:

S[0]=0, S[1]=1, S[2]=2, S[3]=3

i=2, j=3

z2 = S[S[2]+S[3]] = S[(2+3) mod 4] = S[1] = 1 = 00000001

RC4: ESEMPIO(8)

3. PRGA;

Second Loop

Valori Iniziali

S[0]=0, S[1]=1, S[2]=3, S[3]=2

i=1, j=0

Algoritmo:

i=1+1=2

j=0+S[2]=0+3=3

SWAP(S[2],S[3]) = S[2]=2, S[3]=3

Valori Finali:

S[0]=0, S[1]=1, S[2]=2, S[3]=3

i=2, j=3

z2 = S[S[2]+S[3]] = S[(2+3) mod 4] = S[1] = 1 = 00000001

RC4: ESEMPIO(9)

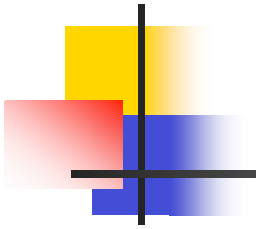
4. Cifratura del testo

H (ASCII) = 072 (ANSI) = 01001000 (Binario)
I (ASCII) = 073 (ANSI) = 01001001 (Binario)

Operazione di XOR:

H _ z1 = 01001000 _ 00000001 = 01001001 = I
I _ z2 = 01001001 _ 00000001 = 01001000 = H

5. Testo Cifrato: IH



RC4: ESEMPIO(fine)

Nella realtà:

- La chiave non è statica
- Si utilizza l'*initialization vector* (IV)

Si giustappone la Keystream all' IV:

- Si genera una chiave diversa per ogni pacchetto

WPA: il presente della sicurezza

Le debolezze del WEP sono note. Lo standard di sicurezza della IEEE riguardo alle WLAN è ancora un draft.

Un'associazione di costruttori (WI-FI Alliance) si è mossa per offrire una soluzione immediata per il problema della sicurezza:

WPA= *Wi-Fi Protected Access*

- E' utilizzato sugli standard IEEE 802.11i
- Incrementa in maniera significativa il livello di protezione
- Garantisce una compatibilità all'indietro e all'avanti

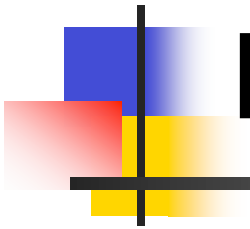
WPA e TKIP

WPA: maggiori garanzie di sicurezza

- impiego del Temporary Key Integrity Protocol (TKIP):

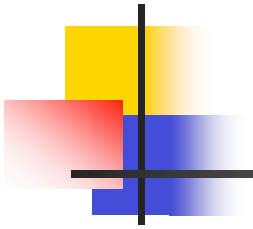
  gestisce dinamicamente le chiavi (2 gerarchie di chiavi: Pairwise Hierarchy e Group Hierarchy) e fa un controllo di integrità del pacchetto.

  E' responsabile della generazione di una nuova chiave WEP per ciascuna sessione instaurata.
- introduce anche una serie di regole per la generazione degli IV, per il rilevamento di attacchi di forza bruta e per la costruzione della chiave che viene usata dall'algoritmo RC4.



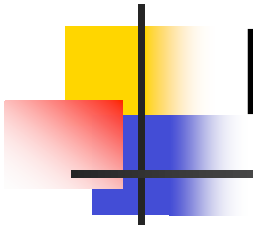
Meccanismi di Autenticazione

Fabio Faruoli



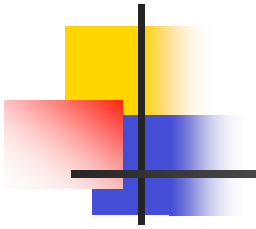
Processo di Autenticazione

- L’autenticazione e’ uno degli elementi piu’ critici nella sicurezza
- Una buona infrastruttura di autenticazione protegge dalla maggior parte degli attacchi



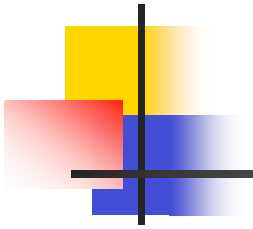
Processo di autenticazione

- È formato da tre fasi:
  1. Autenticazione
  2. Autorizzazione
  3. Accounting



Autenticazione

- Consente a un entita' (una persona o un sistema) di dichiarare la sua identita' a un'altra entita'
- Di solito l'entita' che vuole identificarsi deve dimostrare la conoscenza di un segreto all'altra
- Autenticarsi significa disporre di **credenziali**



Autenticazione

Le credenziali possono essere di alcuni tipi:

Quello che sai

- Password, pin.

Quello che hai

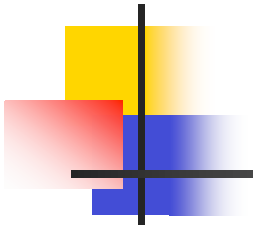
- Token, badge, smartcard

Quello che sei

- Impronte digitali, riconoscimento vocale, analisi della retina

Combinazioni:

- Quello che hai + quello che sai . Token con pass dinamiche
- Quello che sei + quello che sai . Impronte digitali + pin .

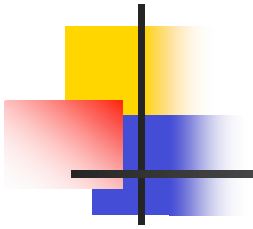


Autorizzazione

- Dopo avere verificato l'identita' del soggetto il sistema informatico deve determinare i suoi diritti e privilegi

- Es:

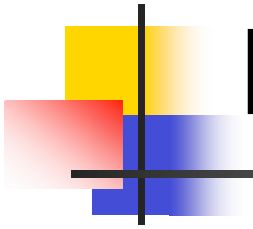
  consentire ad un utente dell'area marketing di potere accedere alle sole risorse (reti, fileserver, web interno, etc) del marketing e non a quelle dell'amministrazione



Accounting

- La registrazione di eventi relativi alle autenticazioni e autorizzazioni
- Es:

```
user pippo logged in on 7 Mar 2002 on port 12
Failed password for user mario on 8 Mar 2002
```



Protocolli di autenticazione

- Lo scambio delle credenziali deve essere immune allo sniffing

- Per questo e' fondamentale avere dei solidi meccanismi per gestire l'autenticazione.

- I meccanismi sono definiti come protocolli di autenticazione ma non tutti sono sicuri

- La combinazione di questi protocolli assieme ad altre tecnologie di autenticazione sono il fondamento della sicurezza del Wi-Fi

Protocolli di autenticazione

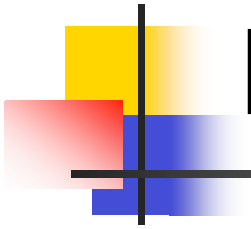
- I protocolli di autenticazione definiscono delle metodologie per lo scambio di credenziali fra due peer
- I protocolli più diffusi sono:
  - -PAP
  - -CHAP
  - -MS-CHAP v1/v2
  - -EAP

Protocollo PAP

- PAP: Password Authentication Protocol

-Il modulo base di autorizzazione **username** e **password** viene trasferito sulla rete e confrontato con una tabella delle coppie username-password che risiede nel server

- **La password attraversa la rete in chiaro**



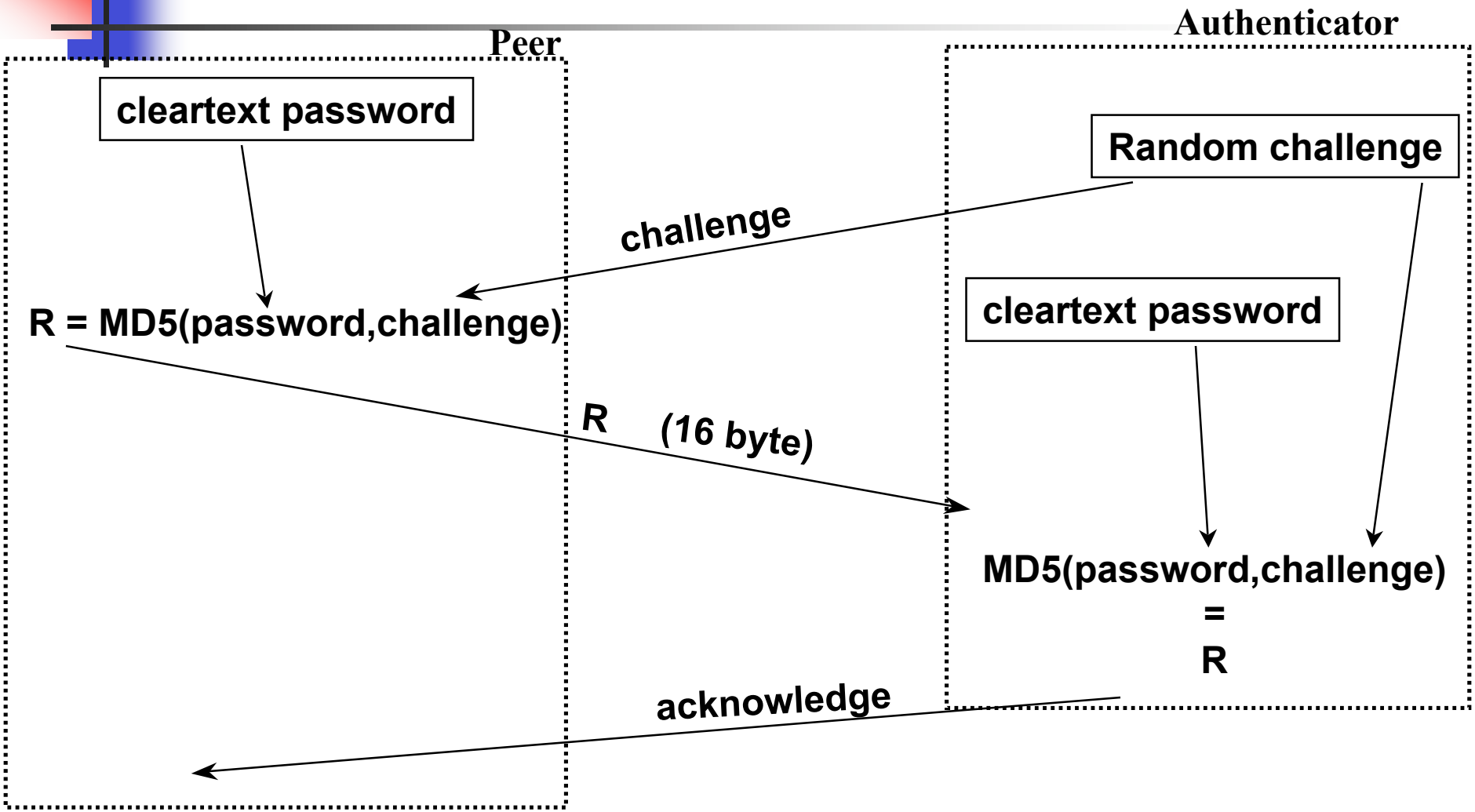
Protocollo CHAP

- CHAP: Challenge Authentication Password Protocol

- L'autenticatore invia, dopo aver stabilito la connessione, un challenge al client che chiede di essere autenticato.

- Il client prova di essere in possesso dello shared secret rispondendo al suo challenge .

Protocollo CHAP



Protocollo MS-CHAP v1

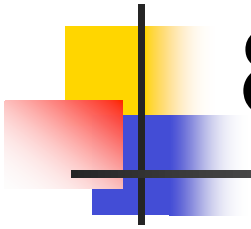
- **MS-CHAP v1**:Microsoft Challenge Handshake Authentication Protocol v1
- E' una versione proprietaria del protocollo CHAP sviluppata da Microsoft
- Supporta esclusivamente l'autenticazione del client verso il server
- Utilizza lo schema di autenticazione Lan Manager (LM)
- Invia i dati cifrati utilizzando il protocollo MPPE (Microsoft Point-to-Point Encryption).

Protocollo MS-CHAP v2

- **MS-CHAP v2:** Microsoft Challenge Handshake Authentication Protocol v2
- Supporta la mutua autenticazione
- Utilizza esclusivamente lo schema di autenticazione NTLM
- Invia i dati cifrati utilizzando il protocollo MPPE (Microsoft Point-to-Point Encryption)

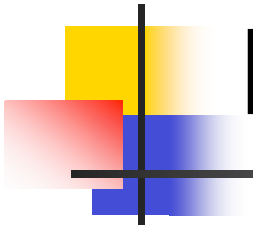
802.1X

- Requisiti che si vogliono raggiungere per il Wi-Fi con l'802.1x:
  - -Mutua autenticazione fra utente e network
  - -Cifratura delle credenziali inviate
  - -Generazione dinamica delle chiavi crittografiche (WEP, TKIP, etc)



802.1X

- È composto da tre elementi:
  - -Supplicant
  - -Authenticator
  - -Authentication server

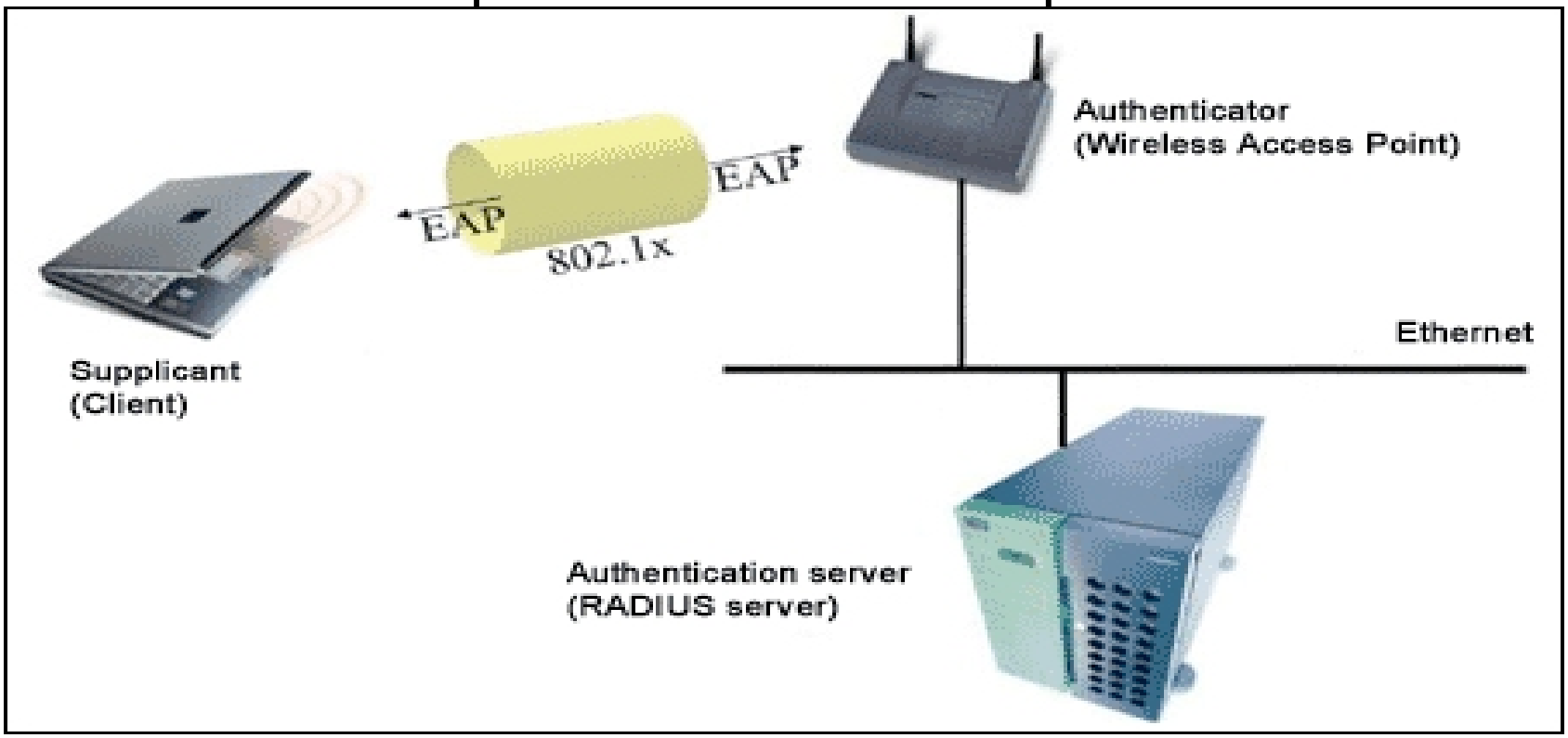


Protocollo EAP

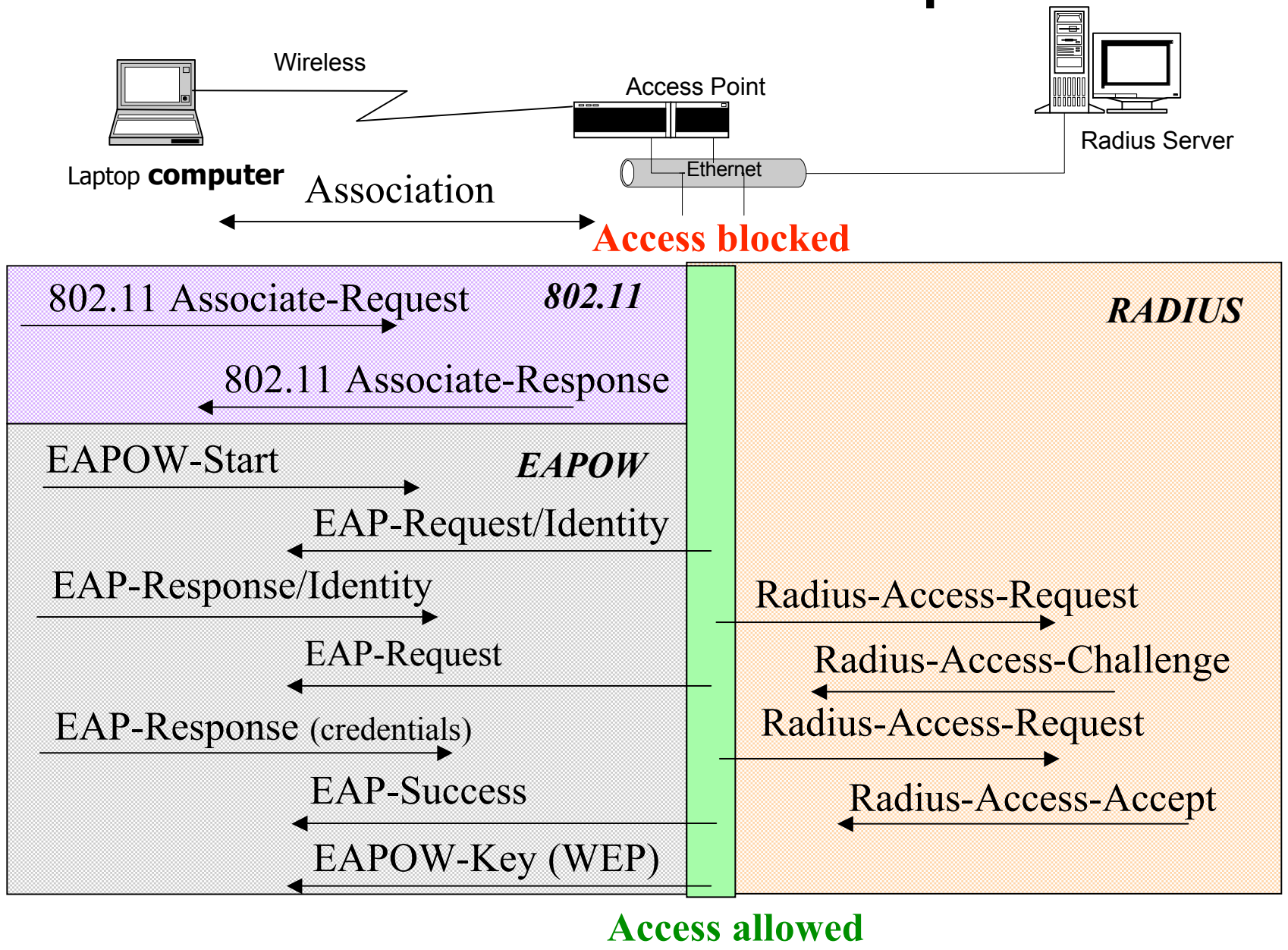
- EAP: Extensible Authentication Protocol
- È utilizzato per selezionare uno specifico meccanismo di autenticazione, negoziato tra Supplicant e Authenticator

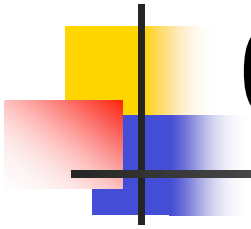
802.1X - EAP

802.1x utilizza per l'autenticazione il protocollo EAP



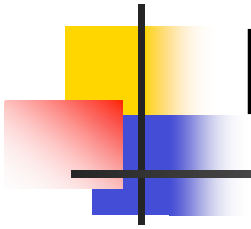
802.1X: Esempio





06 - I metodi EAP

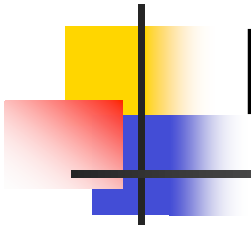
- EAP-MD5
- EAP-SIM / EAP-AKA
- EAP-LEAP
- EAP-TLS Transport Layer Security
- EAP-TTLS Tunnelled TLS
- EAP-PEAP Protected EAP
- Fast EAP



EAP-TLS

TLS: Transport Layer Security

- Proposta di standardizzazione dell' SSL (Secure Socket Layer) da parte di Netscape
- la prima versione di TLS può essere considerata come SSL v3.1



EAP-TLS

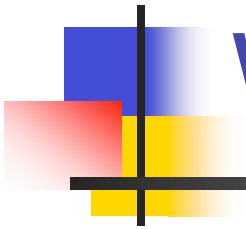
Privatezza del collegamento La crittografia è usata dopo un handshake iniziale per definire una chiave segreta. Per crittografare i dati è usata la crittografia simmetrica

Autenticazione L'identità nelle connessioni può essere autenticata usando la crittografia asimmetrica, o a chiave pubblica. Così i clients sono sicuri di comunicare con il corretto server, prevenendo ogni interposizione

Affidabilità verifica che i dati spediti tra client e server non siano stati alterati durante la trasmissione.

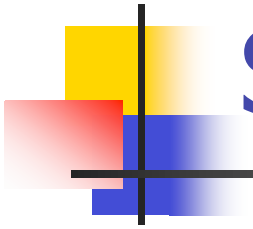
Sviluppi Futuri: 802.11i

- 802.11i risolverà problemi di sicurezza dell'802.11
- Utilizza 802.1x per l'autenticazione
- I suoi elementi sono:
  - -WPA
  - -TKIP
  - -Message Integrity Check
- 802.11i non e' ancora definito completamente ma alcune sue feature sono gia' utilizzate



WEP (In) Security

Claudio Bortone



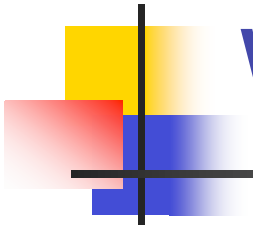
Sommario

- Introduzione
- Debolezza degli stream cipher
- Attacco FMS (Fluhrer, Mantin, Shamir)
- Riutilizzo del key stream
  - 2 messaggi con lo stesso K (key stream)
  - Trovare un P (plaintext)
- Dizionario di decrittazione
- Problemi di CRC
  - Message modification
  - Message injection
- Authentication spoofing
- Bibliografia

Implementazioni WEP

- Lo standard IEEE802.11 prevede due implementazioni di WEP:
  - *classica*, con chiave a 40 bit
  - *estesa*, con chiave a 128 bit
- Versioni WEP anche a 256 bit ma fuori standard

Con 128 bit è praticamente impossibile ricavare la shared key attraverso attacchi brute-force



WEP Security goals

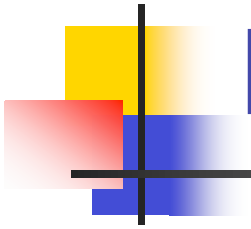
- Confidenzialità
  - solamente gli interlocutori devono conoscere il contenuto dei messaggi scambiati
- Autenticazione
  - la rete deve essere utilizzata solamente da chi è autorizzato a farlo
- Integrità dei dati
  - i dati scambiati non debbono essere contraffatti

- In tutti e tre i casi la sicurezza risiede nella difficoltà di ricerca della chiave attraverso attacchi tipo brute-force

- … vedremo che indipendentemente dal tipo di implementazione WEP utilizzata, gli obiettivi di sicurezza prefissati non vengono garantiti!

Wireless Equivalent Privacy

- Ricordiamo che WEP utilizza l'algoritmo di crittazione di tipo stream cipher RC4
- Questo tipo di algoritmi operano tramite l'espansione di una chiave "corta" (in WEP formata dalla concatenazione dell'IV e della WEP key o shared key) in un flusso infinito di bit pseudo-random (key stream).
- Lo XOR tra il key stream e ed il messaggio in chiaro (plaintext) effettuato dal trasmettitore produce il testo cifrato (ciphertext)



Debolezza degli Stream Cipher

- Esiste una ben nota debolezza degli algoritmi stream cipher:
- crittando 2 messaggi diversi con lo stesso key stream, dal conseguente ciphertext si possono ricavare facilmente informazioni riguardanti i due messaggi
- … vediamo come!

Un po' di logica …

- Supponiamo che:
  - *P1* e *P2* siano due messaggi in chiaro (plaintext)
  - *K* sia la chiave (keystream) con cui vengono crittografati i due messaggi

- ricordando che :

$$A \oplus A = identità$$

Un po' di logica …

$$C1 = P1 \oplus K$$

$$C2 = P2 \oplus K$$

$$C1 \oplus C2 = (P1 \oplus K) \oplus (P2 \oplus K) =$$

$$= P1 \oplus P2$$

Ottenendo così lo XOR del plaintext

Un po' di logica …

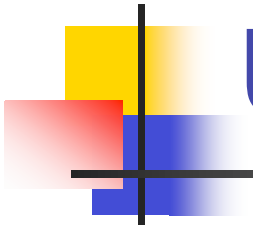
Se conoscessimo uno dei due plaintext …

$$P1 \oplus (C1 \oplus C2) =$$

$$= P1 \oplus (P1 \oplus P2) =$$

$$= P2$$

… otterremmo l'altro con un semplice XOR



Un po' di logica …

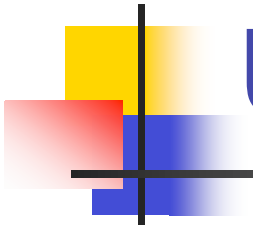
Inoltre se conoscessimo un plaintext ed il corrispondente ciphertext …

$$P1 \oplus C1 =$$

$$= P1 \oplus (P1 \oplus K) =$$

$$= K$$

… otterremmo la key stream!



Un po' di logica ...

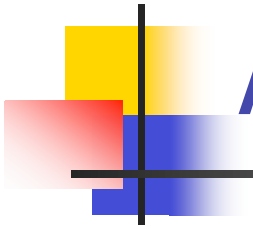
La conoscenza della chiave K permette di effettuare attacchi statistici per il recupero del testo in chiaro!

Attacco FMS

- Fluhrer, Mantin e Shamir hanno dimostrato la debolezza dell'algoritmo KSA (Key Scheduling Algorithm) dell'RC4 utilizzato nell'implementazione WEP

- Il loro attacco si basa sull'utilizzo del solo primo byte della sequenza pseudo-random prodotto dall'output generator di RC4

Attacco FMS



- L'equazione associata a questo byte è

$$S[S[1] + S[1]]$$

- Dopo la fase di setup, questo byte dipende unicamente da 3 valori dell'array di stato

$$S[1], S[S[1]], S[S[1] + S[1]]$$

Attacco FMS

- L'attacco consiste nel ricavare informazioni sulla chiave osservando questo valore.

- Utilizzando la terminologia di *Fluhrer e altri* chiameremo *resolved* il pacchetto che assume il particolare stato in cui :

  "il suo IV è tra quelli che portano l'algoritmo KSA a rilasciare informazioni sulla chiave"

Attacco FMS

- L'attacco è statistico per sua natura, ogni pacchetto *resolved* ci dà una probabilità del 5% associata all' ipotesi di chiave corretta ed una probabilità di 95% di ipotesi di chiave errata

L'osservazione di un numero sufficiente di pacchetti "*resolved*" porta sicuramente alla rivelazione della shared key utilizzata da WEP

Riutilizzo del key stream

- Perchè sia possibile sfruttare le vulnerabilità prima esposte devono essere soddisfatte due condizioni:
  - esistano messaggi crittati con lo stesso key stream
  - conoscenza parziale di parte di questi messaggi

2 Messaggi con lo stesso K

- Sappiamo che il key stream K utilizzato in WEP, è generato dall'inizialization vector v e dalla WEP key k come:

$$K = RC4(v, k)$$

- quindi K dipende unicamente da v e da k

- Possiamo fare alcune considerazioni ...

2 Messaggi con lo stesso K

- Generalmente la WEP key k è una chiave fissa (gli amministratori rete raramente la cambiano)
- In pratica il keystream K dipende solo da v
- ricordiamo che v viene trasmesso in chiaro sulla rete e quindi l'attaccante può facilmente verificare se un IV venga riutilizzato

2 Messaggi con lo stesso K

- Inoltre considerando che:
  - la dimensione di v è di 24 bit
  - il pacchetto di livello 2 utilizzato in IEEE802.11 ha una dimensione massima di 1500 byte
- Potremmo effettuare una stima sull'intervallo di tempo massimo prima del riutilizzo di un IV

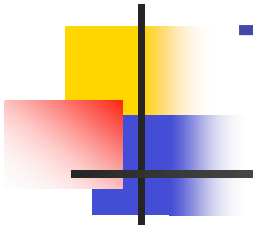
2 Messaggi con lo stesso K

$$\frac{1500byte}{11Mbps}\cdot 24bit = \frac{1500bit\cdot 8}{(11bit\cdot 10^{6})\cdot \sec^{-1}}\cdot 2^{24}bit \approx 18000\sec$$

- Ovvero nella versione classica di WEP dovremmo attendere al massimo 5 ore prima di vedere un IV duplicato

2 Messaggi con lo stesso K

- E se non bastasse:
  - alcune schede wifi resettano il loro IV a 0 ogni volta che vengono attivate e incrementano di 1 ogni pacchetto inviato
  - lo standard non specifica come debbano essere calcolati gli IV quindi estremizzando un apparato potrebbe utilizzare sempre lo stesso IV senza essere fuori standard
  - ogni volta che si verifica una collisione il pacchetto viene rispedito con lo stesso IV



Trovare P

- Risalire ad un plaintext:
  - dal traffico IP rilevato in una rete wireless (IP utilizza strutture ben definite all'interno di un pacchetto)
  - sequenze di login, generalmente le stesse per tutti gli utenti
  - messaggi di benvenuto
  - invio di e-mail ad un utente per poi attendere che passi attraverso il link wireless.(e-mail di spam sicuramente non desterebbero allarmi)
  - …

Dizionario di decrittazione

- Abbiamo visto come ottenere la key stream K a partire da un plaintext $P1$ e dal suo corrispondente ciphertext $C1$

$$P1 \oplus C1 =$$

$$= P1 \oplus (P1 \oplus K) =$$

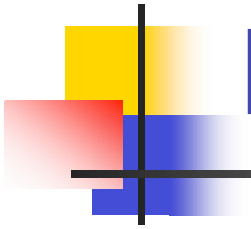
$$= K$$

Dizionario di decrittazione

- Con questa key stream K si può decrittare qualsiasi pacchetto che sia stato crittato con lo stesso IV
- E' quindi possibile creare una tabella che metta in relazione un pacchetto con il corrispondente IV
- nel WEP classico richiederà uno spazio di $2^{24} \cdot 1500 = 25GB$

Dizionario di decrittazione

- Il vantaggio nell'utilizzo di questo sistema è nella velocità di decrittazione dei pacchetti
- lo svantaggio invece è nel tempo di necessario alla creazione della tabella completa.
- questo approccio è risulta sconveniente nel caso di utilizzo di chiavi WEP di lunghezza maggiore di 40 bit



Problemi di CRC

- Il protocollo WEP utilizza un campo di controllo di integrità(o checksum) che dovrebbe assicurare che i pacchetti non vengano modificati durante il transito
- il meccanismo checksum CRC (Ciclic Redundant Code) applicato allo stream cipher utilizzato in WEP, garantisce solo l'integrità di pacchetti che siano stati modificati da errori nel canale e non contro attacchi maliziosi

- vediamo come sfruttare questa debolezza ...

Message modification

- Mostriamo come un messaggio possa essere modificato in transito senza che questo venga rilevato, utilizziamo la proprietà di linearità dei checksum CRC:

$$c(X \oplus Y) = c(X) \oplus c(Y)$$

Message modification

- sia C un ciphertext intercettato prima che abbia raggiunto la destinazione
- sia P il plaintext associato a C
- allora:

$$C = RC4(v,k) \oplus < P, c(P) >$$

- Con $<P,c(P)>$ pari alla concatenazione del plaintext con il suo checksum

Message modification

- E' possibile trovare un nuovo ciphertext C' associato al plaintext P':

$$P' = P \oplus \ddot{A}$$

- con _ scelto arbitrariamente dal malintenzionato

- Quindi sfruttando la linearita del CRC…

$$c(P) \oplus c(\ddot{A}) = c(P \oplus \ddot{A})$$

Message modification

- … è possibile ottenere C'

$$C' = C \oplus \langle \ddot{A}, c(\ddot{A}) \rangle =$$
$$= RC4(v,k) \oplus [\langle P, c(P) \rangle \oplus \langle \ddot{A}, c(\ddot{A}) \rangle]$$
$$= RC4(v,k) \oplus \langle P \oplus \ddot{A}, c(P) \oplus c(\ddot{A}c) \rangle$$
$$= RC4(v,k) \oplus \langle P', c(P \oplus \ddot{A}) \rangle$$
$$= RC4(v,k) \oplus \langle P', c(P') \rangle$$

Message injection

- Ora mostreremo come il WEP non fornisca un valido controllo di acesso, per fare questo utilizzamo un'altra proprietà del checksum :

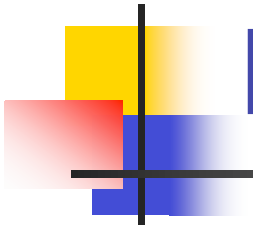
il checksum WEP è funzione del messaggio

Message injection

- Un malintenzionato conoscendo un plaintext P e il corrispondente ciphertext C potrebbe inviare un nuovo messaggio P' da lui generato (injection)

$$P \oplus C = P \oplus (P \oplus RC4(v,k)) = RC4(v,k)$$

$$C' =< P', c(P') > \oplus RC4(v,k)$$



Message injection

- il nuovo pacchetto utilizzerà lo stesso IV dell'originale
- questo non è un problema poiché, come è stato già detto, lo standard non specifica come debbano essere variati gli IV

Authentication spoofing

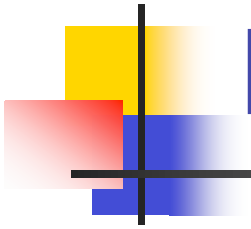
- Il meccanismo di associazione di una stazione mobile ad un access point è qui descritto:
  - la stazione mobile, all'accensione, invia all'access point una richiesta di autenticazione
  - l'access point risponderà con una stringa random di 128 bit inviata in chiaro (challenge)
  - la stazione mobile cifrerà la stringa con la chiave WEP e la reinvierà all'access point
  - l'access point verificherà se la stringa ricevuta è stata cifrata con la chiave WEP corretta

Authentication spoofing

- Un malintenzionato quindi, sniffando un una sequenza di autenticazione leggittima

  potrebbe facilmente ricavare una coppia plaintext/ciphertext valida da cui ricavare il corrispondente key stream

- il che è sufficiente per inviare la risposta corretta al challenge (inviato in chiaro) lanciato dall'access point



Bibliografia

- *Nikita Borisov, Ian Goldberg, David Wagner* - Intercepting Mobile Communications: The Insecurity of 802.11
- *Scott Fluhrer, Itsik Mantin, Adi Shamir* - Weaknesses in the Key Scheduling Algorithm of RC4
- *Adam Subblefield, John Ioannidis, Aviel D. Rubin* - Using the Fluhrer, Mantin and Shamir Attack to Break WEP