→ **Wifi** → è composto da **onde radio** → il segnale fisico è intercettabile

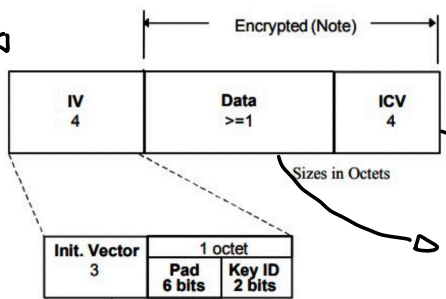
↘ fin da subito si è pensato di cifrare i bit a basso livello

- Fisico → **Intercettabile** → onde radio
- DataLink → Cifratura → indipendemente dalle applicazioni e dalle scelte implementative dei livelli superiori, la comunicazione viene cifrata.
  ↘ all'inizio viene usato RC4
- Rete
- Trasporto
- Applicativo

↘ Siccome il livello fisico è sottoposto a rumori ho bisogno di una cifratura robusta → possibile correzione degli errori

→ **WEP** → tecnologia di sicurezza per reti Wifi ormai obsoleta, basata su RC4

pacchetto ↘ Chiave da 40 a 104 bit



checksum

↘ blocco contenente i dati

↘ contiene anche i payload dei livelli superiori

3 byte

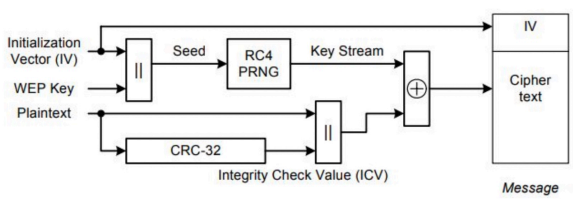
Non cifrato!

→ **Cifratura**

al centro c'è RC4

↘ prende in input, in WEP, 64 bit ottenuti dalla concatenazione dei 3 byte di IV con i 5 byte della chiave (40 bit).

Questi 8 byte in input, fungono da chiave di inizializzazione di RC4. (produce il Key Stream)



→ 8 byte in input RC4
- 3 byte di IV sono variabili, e per tanto cambiano ad ogni frame
- 5 byte di WEP Key sono fissi. Sono inoltre condivisi tra mittente e destinatario

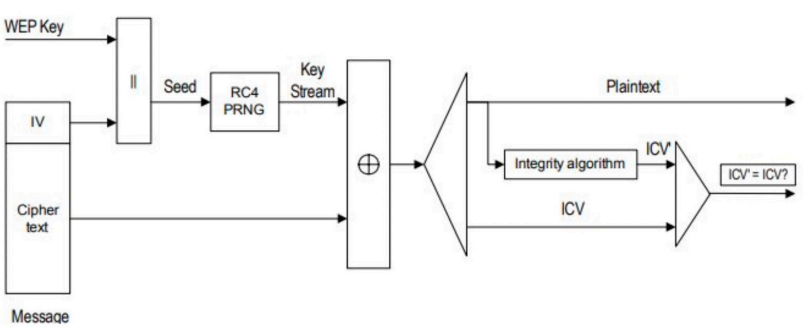
→ Operazione di cifratura
↘ avviene mettendo in XOR tra la chiave corrente e il risultato di una concatenazione tra il plaintext (payload) e il checksum

→ Nella versione proposta da WEP → chiave da 104 bit

→ Decifratura
↘ può ottenere il keystream con WEP e IV, applicare l'operazione di XOR che è invertibile
↘ Il destinatario:
- ha la WEB Key segreta utilizzata per inizializzare RC4
- riceve IV + ciphertext

→ Attacco
↘ Oppure avere alcuni o tutti i Keystream
↘ Individuando le componenti l'attaccante avrebbe il controllo completo della cifratura WEP
→ Scelta più semplice, individuare i 3 bit di IV semplificando la ricerca a 5 byte

- RC4, pubblico
- IV + Ciphertext osservabile
- WEP o Keystream



Debolezze WEP

- prevede RC4 più tutta la parte di costruzione del frame.
- IV (24 bit) + Web Key (40 bit) = K → input a RC4
- Mettere in chiaro IV, scelta di WEP
- Attacco

Partiamo con l'idea che avere una parte della chiave segreta visibile può essere sfruttata come vulnerabilità del sistema.
Inoltre se RC4 ha delle debolezze nella parte matematica / crittografica porta a un possibile attacco.

Primo attacco

- ripetizione visibile di IV tra diversi frame e sul fatto che sia noto l'utilizzo di XOR per effettuare la cifratura su keystream.
- attacco che non va a fondo
  - no WEP Key
  - no controllo sulla cifratura
- leggo alcuni frame
- Il meccanismo di cifratura è di prendere tutti i byte in chiaro, ciascuno viene messo in XOR con la chiave generata da RC4 produce il messaggio cifrato.
  - Ogni byte viene cifrato e trasmesso, permettendo all'attaccante di intercettare diverse parti.

$$c_1 = p_1 \oplus K$$
$$c_2 = p_2 \oplus K$$
$$c_1 \oplus c_2 = (p_1 \oplus K) \oplus (p_2 \oplus K) = p_1 \oplus p_2$$

- Proprietá algebrica che, se prendo due elementi cifrati e li metto in XOR riesco ad ottenere lo XOR di a plaintext.
  - Non conosco K ma posso scoprire quali bit sono comuni o diversi
  - e trovare K sapendo che viene usata sempre la stessa
  - IV cambia ed é visibile

- L'impatto dell'attacco é dipendente da quanto facilmente posso rilevare una collisione e quindi una ripetizione di IV. La quantitá di tempo é

$$\frac{1500 \text{ bytes}}{\text{packet}} \times \frac{8 \text{ bit}}{1 \text{ byte}} \times \frac{1 \text{ sec}}{11 \text{ Mbit}} \times \frac{1 \text{ Mbit}}{10^6 \text{ bit}} \times 2^{24} \text{ packet} = 18,500 \text{ s} = 5 \text{ ore}$$

- 1500 bytes / packet: frame
- 8 bit / 1 byte: dati
- 1 sec / 11 Mbit: se la vete é piú veloce potrebbe avenire prima
- 1 Mbit / 10^6 bit: canale
- 2^{24} packet: IV é da 24 bit, dopo 2^{24} pacchetti abbiamo sicuramente una rip.

-> Secondo Attacco (FSM)

-> andiamo a considerare l'eventualitá dell'utilizzo di opp informatione specifiche

Impatto molto grave -> vicostruzione di K PSK + IV

deriva da debolezza di RC4

Presenza di una relazione tra K e lo stream

-> se conosco: primi caratteri del plain text(p)

-> RC4 -> funzionamento

-> 2 fasi

1. -> La K che io voglio scoprire, quella iniziale, pilota la fase di ordinamento dell'array S. (fatto tutte le volte)
2. -> Per ogni byte da trasmettere, genero un byte da K.

Quello che hanno scoperto é che se io riesco a scoprire la prima parte della Keystream, posso risalire a K

-> Stima del costo per l'attaccante

-> Per ricavare i primi K del keystream devo conoscere il primo byte in chiaro

→ Utilizzando → I primi byte → So che il primo byte è standard 0xAA

Alcune info ottenute catturando IV

Lo sviluppo di questo attacco FSM → Riesco a ricostruire K

Con 40.000 frames → 50%

Con 85.000 frames → 95%

→ AirCrack → algoritmo