Fourth Lab Report

Dandrea Silvio Facchini Christian Ferrari Rudi
{*silvio.dandrea, c.facchini, rudi.ferrari*}*@studenti.unitn.it*

4 giugno 2007

Compendio

In questo documento andremo ad analizzare una parte importante dello Standard IEEE 802.11, quella che specifica il protocollo utilizzato per rendere sicure le trasmissioni radio delle reti Wi-Fi, ossia Wired Equivalent Privacy (in seguito WEP). In specifico andremo ad analizzare alcune Network Interface Card (NIC in seguito) e come cambiano i loro Initialization Vector (IV in seguito) sulla base dell'algoritmo RC4. Infine proveremo a determinare la chiave WEP di una rete mediante l'utilizzo del software Aircrack, riportando i tempi e il numero di IV usati per la determinazione della chiave[1].

1 Test bench

Questa sezione è strutturata in tre parti. La prima verrà dedicata a descrivere brevemente l'importanza del WEP e la sua storia, includendo il suo funzionamento; la seconda per descrivere brevemente l'hardware utilizzato nei test, infine una terza per la descrizione del software.

1.1 WEP

Il WEP è quella parte dello Standard IEEE 802.11 che specifica il protocollo utilizzato per rendere sicure le trasmissioni radio delle reti Wi-Fi. È stato implementato e progettato per poter rendere le trasmissioni radio sicure, ma nel corso degli anni e con l'attenta analisi da parte degli analisti sono state riscontrate numerose falle nell'implementazione dell'algoritmo crittografico. Questo ha fatto sì che venisse progettato ed implementato un nuovo standard denominato Wi-Fi Protected Access (WPA) rilasciato nel 2003 e facente parte dell'IEEE 802.11i (conosciuto come WPA2) definito nel giugno del 2004. Il WEP è quindi considerato il minimo strumento di sicurezza per garantire la privacy degli utenti di una rete. Il WEP si basa sull'algoritmo di cifratura RC4 il quale può utilizzare due chiavi, a 40 bit o a 104 bit. A queste vengono in seguito aggiunti 24 bit del vettore di inizializzazione (IV). Quindi in totale si hanno 64/128 bit i quali vengono usati come seed per il generatore di numeri casuali (PRNG). La chiave è segreta mentre l'IV viene spedito in chiaro (necessario per poter effettuare la decodifica). Proprio per il fatto che l'IV viene spedito in chiaro, occorre che cambi in continuazione altrimenti la chiave WEP può essere molto più facilmente crackata.

1.2 Hardware

Questa sessione serve a dare una panoramica generale dei dispositivi hardware utilizzati nel corso degli esperimenti. Nei vari test sono stati utilizzati tre PC. Il primo (numero 1 in tabella 1) è servito a generare del traffico di pacchetti, il secondo (numero 2) è stato utilizzato per fare sniffing, mentre il terzo (numero 3) è stato utilizzato durante i test di cattura della chiave WEP. Oltre ai calcolatori sono state utilizzate 8 NIC con le quali abbiamo effettuato l'analisi degli IV.

Access Point

L'access point utilizzato nei nostri test è un Linksys modello WRT54G. Questo è un AP di fascia medio-bassa e implementa lo Standard IEEE 802.11b.

[1] Abbiamo scelto di utilizzare le abbreviazioni e gli acronimi presenti e utilizzati nello Standard IEEE 802.11. Per termini particolarmente importanti, verranno riportate le loro intere trascrizioni e il loro acronimo, o abbreviazione, tra parentesi.

Numero di rif.	Processore	RAM	S.O.	NIC
1	Intel Pentium IV 2.8GHz	512 MB	Debian 4.0	Nortek Cisco Aironet 350
2	Intel Pentium M 1.7Ghz	512 MB	BackTrack	Integrata Broadcom 10/100/1000 Gigabit
3	Intel Pentium IV 2.66GHz	512 MB	SuSe 10.2 BackTrack	NetGear MA521 Senao SL2511-CD Plus

Tabella 1: Caratteristiche dei PC usati.

Wireless NIC

L'algoritmo su cui si basa il WEP è l'RC4, inoltre sappiamo che l'IV continua a cambiare ad ogni pacchetto. Il modo con cui i 24 bit dell'IV cambiano è diverso da schedina a schedina. Nei test che andremo a sostenere vogliamo osservare come cambia l'IV in vari tipi di schedine. A questo scopo sono state utilizzate delle NIC le cui caratteristiche sono descritte nella tabella 2.

1.3 Software

In questa sessione sperimentale, sono stati utilizzati tre software: BackTrack, Wireshark e Aircrack-ng.

BackTrack

Backtrack è una distribuzione Live basata su Slackware, ed è il risultato dell'unione di due precedenti distribuzioni: Whax e Auditor. È stata utilizzata per questo laboratorio in quanto contiene tutti gli strumenti utili ad effettuare monitoraggio di reti wireless e sniffing, senza bisogno di installare nessun pacchetto aggiuntivo.

Wireshark

Wireshark (in precedenza chiamato Ethereal) è un software libero, utile ad effettuare l'analisi dei protocolli. Wireshark è solitamente utilizzato per determinare malfunzionamenti all'interno di una rete, attraverso lo sniffing di pacchetti. Ne sono state utilizzate diverse versioni, a seconda di quali sistemi operativi e dispositivi venivano utilizzati.

Aircrack-ng

Aircrack-ng è un insieme di strumenti per la verifica di reti wireless. È un software che funziona da linea di comando e gli strumenti che lo compongono sono diversi, ma quelli utilizzati nei test sono: airodump, aireplay e aircrack. Questo software è stato utilizzato nei test per la cattura della chiave WEP.

- **airodump** è lo strumento che permette di catturare i pacchetti direttamente dall'interfaccia di rete e salvarli in un file *.cap o *.ivs. Il comando utilizzato nei test è il seguente:

  ```
  airodump-ng -w FILE --channel CANALE --bssid MAC IF
  ```

  dove `FILE` è il nome del file su cui andare a scrivere, `CANALE` il numero del canale da cui sniffare, `MAC` l'indirizzo dell'Access Point e `IF` l'interfaccia utilizzata per la cattura;

- **aireplay** è il tool che permette di generare pacchetti secondo lo standard 802.11 e quindi di inserirli nella trasmissione. Con tale strumento abbiamo potuto ridurre il tempo necessario a generare un sufficiente numero di pacchetti affinché venisse trovata la chiave WEP. Il comando utilizzato in tal caso è:

  ```
  aireplay-ng -b MAC_1 -a MAC_2 -x PCK -0 COUNT IF
  ```

  `MAC_1` è il BSSID, `MAC_2` è il parametro che permette di settare il MAC address dell'AP, `PCK` fa riferimento al numero di pacchetti al secondo che si vogliono creare, `-0 COUNT` è un parametro che serve ad impostare la modalità d'attacco e nel nostro caso abbiamo impostato un attacco di deautentificazione infinita (`COUNT`=0) ed infine `IF` fa riferimento all'interfaccia utilizzata;

- **aircrack** è il cuore di tutto il software. Permette di ricavarsi le chiavi WEP utilizzando due metodi fondamentali denominati *approccio PTW* e *metodo di FMS/Korek*.

  Il metodo di FMS/KoreK, come suggerisce il nome, è un miglioramento dell'attacco basato sullo storico paper di Fluhrer, Mantin e

Numero di rif	Marca	Modello	Tipo	Std Supportato	Mac Address
1	D-link	DWL-610	CardBus	IEEE 802.11b/g	00:0d:88:51:ac:92
2	Nortek	W-11	CardBus	IEEE 802.11b	00:02:72:00:1c:0a
3	Senao	NL-2511CD PLUS	CardBus	IEEE 802.11b	00:02:6f:34:3a:4d
4	Orinoco	Gold ETS	CardBus	IEEE 802.11b	00:02:2d:b7:2e:3d
5	Orinoco	Silver	CardBus	IEEE 802.11b	00:02:2d:3d:11:e6
6	Orinoco	Combocard	CardBus	IEEE 802.11b/g	00:20:a6:4f:9e:a1
7	NetGear	MA401	CardBus	IEEE 802.11b	00:09:5b:27:ec:91
8	Cisco	Aironet 350	CardBus	IEEE 802.11b	00:0b:be:b2:6b:7e
	AP Linksys	WRT54G		IEEE 802.11b	00:12:17:49:dd:94

Tabella 2: Caratteristiche delle NIC utilizzate e dell'Access Point.

Shamir[2] (ad opera di KoreK[3] appunto): incorpora diversi tipi di attacchi statistici che permettono di scoprire la chiave, questi ultimi vengono utilizzati assieme ad attacchi di forza bruta. L'approccio PTW è un'estensione dell'attacco di Klein[4], il quale dimostrò nel 2005 come le correlazioni tra il keystream RC4 e la chiave siano più di quelle trovate da Fluhrer, Mantin e Shamir. Il punto di forza di questo attacco è il numero ridotto di pacchetti di cui si necessita per crackare la chiave (infatti si ha una probabilità del 95% di crackare una chiave a 104 bit con 85.000 pacchetti e del 50% con 40.000 pacchetti); purtroppo al momento richiede che i pacchetti da cui si estrae l'IV siano di tipo ARP (Request o Reply). Il comando utilizzato nei test è:

```
aircrack-ng -a1 -b MAC -n L_KEY FILE
```

l'opzione `-a1` forza l'attacco alla ricerca della chiave WEP, `MAC` è l'indirizzo dell'AP, `L_KEY` indica la lunghezza della chiave che si intende catturare, infine occorre specificare il file *.ivs in cui sono stati memorizzati i pacchetti.

2 Tests

2.1 Initialization Vector

L'Initialization Vector è un vettore di 3 byte che ha la funzione di "estendere il tempo di vita utile della chiave segreta". Dalla figura 1 possiamo osservare come viene generata la Key Sequence (o "chiave di cifratura dei dati"). Il vettore di inizializzazione viene concatenato alla chiave WEP, che ha una lunghezza di 40 o 104 bit, formando così una stringa di 64 o 128 bit. Questa sequenza di bit verrà data in ingresso all'algoritmo RC4 il quale ha il compito di generare la chiave di cifratura dei dati. Mentre viene generata la chiave di cifratura dei dati, un algoritmo di integrità opera sui dati che si vogliono crittografare e produce un Integrity Check Value (ICV) che verrà concatenato assieme ai dati stessi: questi saranno poi uniti dall'operazione di OR esclusivo insieme alla chiave di cifratura, formando così il testo cifrato. A questo punto il vettore di inizializzazione verrà aggiunto al testo cifrato e trasmesso. La trasmissione dell'IV in chiaro è necessaria per poter decifrare il messaggio in ricezione. Di conseguenza se l'IV cambia in modo non efficiente (non random) si presenta una maggior vulnerabilità del WEP. A tale proposito lo Standard IEEE 802.11 del 1999 fa presente quanto sia critica la scelta dell'IV. Quotiamo:

> When choosing how often to change IV values, implementors should consider that the contents of some fields in higher-layer protocol headers, as well as certain other higher-layer information, is constant or highly predictable. When such information is transmitted while encrypting with a particular key and IV, an eavesdropper can readily determine portions of the key sequence generated by that (key, IV) pair. If the same (key, IV) pair is used for successive MPDUs, this effect may substantially reduce the degree of privacy conferred by the WEP algorithm, allowing an eavesdropper to recover a subset of the user data without

[2] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In Eight Annual Workshop on Selected Areas in Cryptography, August 2001

[3] `http://www.netstumbler.org/showthread.php?t=11869`

[4] A. Klein. Attacks on RC4 stream cipher. Submitted to Designs, Codes and Cryptography, 2007

any knowledge of the secret key. Changing the IV after each MPDU is a simple method of preserving the effectiveness of WEP in this situation.

A questo punto vogliamo osservare come cambia l'IV per le 8 diverse NIC descritte nel capitolo precedente. Per poter analizzare la sequenza degli IV utilizzata per generare la chiave di cifratura attraverso l'algoritmo RC4, occorre sniffare del traffico di una rete.

A questo scopo è stata instaurata una rete privata composta da un AP ed un PC (numero 3 nella tabella 1, O.S.(2) e NIC(1)), dopodiché è stato generato del traffico pingando l'AP dal PC. Mediante un secondo calcolatore (numero 2 in tabella 1) ed il software Wireshark sono stati quindi catturati alcuni pacchetti. Ogni schedina è stata valutata in 5 test effettuati in sessioni separate, ovvero attivando e disattivando l'interfaccia di rete ad ogni sessione. Tali operazioni sono state compiute fisicamente per essere sicuri il più possibile che non si presentassero correlazioni tra le sessioni di prove. Per avere una visione più chiara dei pacchetti che riporteremo, facciamo riferimento ai MAC Address riportati in tabella 2 delle NIC utilizzate. Inoltre evidenzieremo le cifre dell'IV che cambiano.

D-Link DWL-610

Prova 1

```
Frame 1755 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: D-Link_51:ac:92 (00:0d:88:51:ac:92)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 15
    WEP parameters
        Initialization Vector: 0x9c95e4
        Key Index: 0

Frame 1776 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: D-Link_51:ac:92 (00:0d:88:51:ac:92)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 16
    WEP parameters
        Initialization Vector: 0x9c95e5
        Key Index: 0

Frame 1797 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: D-Link_51:ac:92 (00:0d:88:51:ac:92)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 17
    WEP parameters
        Initialization Vector: 0x9c95e6
        Key Index: 0
```

Prova 2

```
Frame 40 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: D-Link_51:ac:92 (00:0d:88:51:ac:92)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 7
    WEP parameters
        Initialization Vector: 0x675105
        Key Index: 0

Frame 57 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: D-Link_51:ac:92 (00:0d:88:51:ac:92)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 8
    WEP parameters
        Initialization Vector: 0x675106
        Key Index: 0

Frame 93 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: D-Link_51:ac:92 (00:0d:88:51:ac:92)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 9
    WEP parameters
        Initialization Vector: 0x675107
        Key Index: 0
```

Le successive prove hanno evidenziato che il cambiamento del vettore di inizializzazione avviene sempre nello stesso modo, infatti ogni qualvolta si presenta un pacchetto dati, l'IV cambia le sue cifre esadecimali meno significative. Più precisamente incrementa di uno la prima cifra rispetto all'IV precedentemente utilizzato. Inoltre abbiamo notato che l'IV di partenza è diverso da sessione a sessione. Riportiamo ora una tabella riassuntiva degli IV (con il corrispettivo valore binario) riferiti alla prova 1, in modo tale da far notare il reale incremento unitario della prima cifra esadecimale meno significativa e alcuni salti nella sequenza IV ipotizzabili ad una perdita di pacchetti nella fase di sniffing.

SN	IV (HEX)	IV (BIN)
5	9c 95 da	10011100 10010101 11011010
7	9c 95 dc	10011100 10010101 11011100
9	9c 95 de	10011100 10010101 11011110
11	9c 95 e0	10011100 10010101 11100000
13	9c 95 e2	10011100 10010101 11100010
15	9c 95 e4	10011100 10010101 11100100
16	9c 95 e5	10011100 10010101 11100101
17	9c 95 e6	10011100 10010101 11100110

Tabella 3: Incremento unitario dell'IV per D-Link DW-610.

Nortek W-11

```
Frame 301 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cc&CTech_00:1c:0a (00:02:72:00:1c:0a)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 27
    WEP parameters
        Initialization Vector: 0x0c0000
        Key Index: 0

Frame 313 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cc&CTech_00:1c:0a (00:02:72:00:1c:0a)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 28
    WEP parameters
```

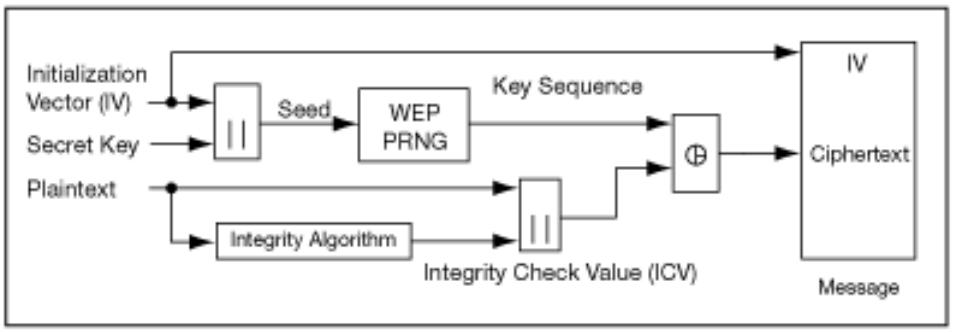


Figura 1: Schema di cifratura WEP.

```
        Initialization Vector: 0x0d0000
        Key Index: 0

Frame 328 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cc&CTech_00:1c:0a (00:02:72:00:1c:0a)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 29
    WEP parameters
        Initialization Vector: 0x0e0000
        Key Index: 0
```

Anche per la NIC in questione l'IV cambia incrementando in modo unitario una cifra, ma in questo caso la cifra che cambia è la seconda più significativa. Nelle successive prove abbiamo notato, al contrario di ciò che capitava con la schedina D-link, che gli IV vengono resettati ovvero il primo IV che si riscontra ha sempre lo stesso valore.

Senao NL-2511CD PLUS

```
Frame 594 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: SenaoInt_34:3a:4d (00:02:6f:34:3a:4d)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 28
    WEP parameters
        Initialization Vector: 0x0d0000
        Key Index: 0

Frame 601 (68 bytes on wire, 68 bytes captured)
IEEE 802.11
    Source address: SenaoInt_34:3a:4d (00:02:6f:34:3a:4d)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 29
    WEP parameters
        Initialization Vector: 0x0e0000
        Key Index: 0

Frame 613 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: SenaoInt_34:3a:4d (00:02:6f:34:3a:4d)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 30
    WEP parameters
        Initialization Vector: 0x0f0000
        Key Index: 0
```

Omettiamo il listato delle prove successive in quanto non presentano aspetti rilevanti. Come nel caso precedente l'IV viene incrementato di un unità nel primo ottetto della sequenza.

Anche con la Senao, da una prova alla successiva l'IV viene resettato sempre allo stesso valore.

Orinoco Gold ETS

```
Frame 499 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Agere_b7:2e:3d (00:02:2d:b7:2e:3d)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 25
    WEP parameters
        Initialization Vector: 0xff1467
        Key Index: 0

Frame 505 (68 bytes on wire, 68 bytes captured)
IEEE 802.11
    Source address: Agere_b7:2e:3d (00:02:2d:b7:2e:3d)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 26
    WEP parameters
        Initialization Vector: 0x001567
        Key Index: 0

Frame 529 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Agere_b7:2e:3d (00:02:2d:b7:2e:3d)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 27
    WEP parameters
        Initialization Vector: 0x011567
        Key Index: 0

Frame 540 (96 bytes on wire, 96 bytes captured)
IEEE 802.11
    Source address: Agere_b7:2e:3d (00:02:2d:b7:2e:3d)
    Destination address: IPv6-Neighbor-Discovery_00:00:00:02
                         (33:33:00:00:00:02)
    Sequence number: 28
    WEP parameters
        Initialization Vector: 0x021567
        Key Index: 0
```

Il frame 540 è stato etichettato come un frame IPv6 che non cambia la sequenza dell'IV in quanto appartiene allo stesso flusso di dati. Anche per la NIC in esame si nota che l'IV incrementa di 1. Per quanto riguarda la possibilità di trovare lo stesso IV in prove diverse, come accaduto con la schedina precedente, con l'utilizzo dell'Orinoco Gold ETS questo fatto non si presenta, infatti ad ogni sessione di prova l'IV iniziale è completamente casuale e diverso da quello che si presenta nelle altre sessioni.

Orinoco Silver

```
Frame 538 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Agere_3d:11:e6 (00:02:2d:3d:11:e6)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 22
    WEP parameters
        Initialization Vector: 0x080000
        Key Index: 0

Frame 564 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Agere_3d:11:e6 (00:02:2d:3d:11:e6)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 23
    WEP parameters
        Initialization Vector: 0x090000
        Key Index: 0

Frame 580 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Agere_3d:11:e6 (00:02:2d:3d:11:e6)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 24
    WEP parameters
        Initialization Vector: 0x0a0000
        Key Index: 0
```

Si riscontrano alcune perdite di pacchetti, ma nulla di anomalo e diverso dall'analisi delle precedenti interfacce di rete. Un aspetto interessante è il *reset* dell'IV, ovvero in ogni sessione di prova il primo frame di dati ha lo stesso IV. Possiamo quindi pensare che sia molto più facile determinare la chiave WEP dal traffico generato da una NIC Orinoco Silver piuttosto che da una Orinoco Gold.

Orinoco Combocard

```
Frame 815 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Proxim_4f:9e:a1 (00:20:a6:4f:9e:a1)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 37
    WEP parameters
        Initialization Vector: 0xa8bae7
        Key Index: 0

Frame 829 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Proxim_4f:9e:a1 (00:20:a6:4f:9e:a1)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 38
    WEP parameters
        Initialization Vector: 0xa8bae8
        Key Index: 0

Frame 853 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Proxim_4f:9e:a1 (00:20:a6:4f:9e:a1)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 39
    WEP parameters
        Initialization Vector: 0xa8bae9
        Key Index: 0
```

Rispetto ai frame catturati utilizzando le precedenti schedine, con la NIC denominata Orinoco Combocard è possibile notare che l'IV cambia il suo valore modificando le cifre meno significative.

SN	IV (HEX)	IV (BIN)
37	a8 ba e7	10101000 10111010 11100111
38	a8 ba e8	10101000 10111010 11101000
39	a8 ba e9	10101000 10111010 11101001
41	a8 ba eb	10101000 10111010 11101011
42	a8 ba ec	10101000 10111010 11101100
43	a8 ba ed	10101000 10111010 11101101
44	a8 ba ee	10101000 10111010 11101110

Tabella 4: Incremento unitario dell'IV per Orinoco Combocard.

La tabella 4 ci permette di notare la perdita di pacchetti e il relativo salto nella sequenza dell'IV. È possibile inoltre rilevare il passo con cui il vettore di inizializzazione cambia.

NetGear MA401

Della NIC in esame riporteremo solamente due frame in quanto non sono state rilevate caratteristiche significative diverse dalle analisi precedentemente effettuate.

```
Frame 194 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Netgear_27:ec:91 (00:09:5b:27:ec:91)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 51
    WEP parameters
        Initialization Vector: 0x0a0000
        Key Index: 0

Frame 211 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Netgear_27:ec:91 (00:09:5b:27:ec:91)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 52
    WEP parameters
        Initialization Vector: 0x0b0000
        Key Index: 0
```

Cisco Aironet 350

Prova 1

```
Frame 217 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 26
    WEP parameters
        Initialization Vector: 0x9e039f
        Key Index: 0

Frame 271 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 27
    WEP parameters
        Initialization Vector: 0x2b012a
        Key Index: 0

Frame 316 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 28
    WEP parameters
        Initialization Vector: 0x5a015b
        Key Index: 0
```

```
Frame 357 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 29
    WEP parameters
        Initialization Vector: 0x6f006e
        Key Index: 0

Frame 388 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 31
    WEP parameters
        Initialization Vector: 0x2d032f
        Key Index: 0

Frame 441 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 32
    WEP parameters
        Initialization Vector: 0x7e027c
        Key Index: 0

Frame 464 (86 bytes on wire, 86 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 33
    WEP parameters
        Initialization Vector: 0x7f027d
        Key Index: 0

Frame 466 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 34
    WEP parameters
        Initialization Vector: 0xbf01bd
        Key Index: 0
```

Prova 3

```
Frame 304 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 28
    WEP parameters
        Initialization Vector: 0x9e029f
        Key Index: 0

Frame 349 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 29
    WEP parameters
        Initialization Vector: 0x350034
        Key Index: 0

Frame 363 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 30
    WEP parameters
        Initialization Vector: 0x980199
        Key Index: 0

Frame 404 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 31
    WEP parameters
        Initialization Vector: 0x120210
        Key Index: 0

Frame 451 (86 bytes on wire, 86 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 32
    WEP parameters
        Initialization Vector: 0x9e039c
        Key Index: 0

Frame 461 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 34
    WEP parameters
        Initialization Vector: 0x78007a
        Key Index: 0

Frame 478 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 35
    WEP parameters
        Initialization Vector: 0xf500f7
        Key Index: 0

Frame 570 (124 bytes on wire, 124 bytes captured)
IEEE 802.11
    Source address: Cisco_b2:6b:7e (00:0b:be:b2:6b:7e)
    Destination address: Cisco-Li_49:dd:94 (00:12:17:49:dd:94)
    Sequence number: 36
    WEP parameters
        Initialization Vector: 0x470345
        Key Index: 0
```

Prova 1		Prova 3	
SN	IV (HEX)	SN	IV (HEX)
26	9e 03 9f	28	9e 02 9f
27	2b 01 2a	29	35 00 34
28	5a 01 5b	30	98 01 99
29	6f 00 6e	31	12 02 10
31	2d 03 2f	32	9e 03 9c
32	7e 02 7c	34	78 00 7a
33	7f 02 7d	35	f5 00 f7
34	bf 01 bd	36	47 03 45

Tabella 5: Variazioni dell'IV nella NIC Cisco.

Dai due listati è possibile notare come la sequenza di IV generata dalla schedina non ha un andamento sequenziale, infatti tra due frame vicini l'IV cambia quasi sempre di 5 cifre su 6. Sembra quindi che la NIC generi l'IV in maniera non lineare. In tabella 5 sono riportate le sequenze degli IV riscontrati nelle prove in listato, in maniera tale da poter notare il cambiamento degli IV.

2.2 Analisi dello spazio degli stati

Lo spazio degli stati (o delle fasi) è uno spazio vettoriale finito m-dimensionale che può rappresentare tutti gli stati possibili di un sistema dinamico (ogni stato è definito da un vettore $\underline{x} \in \mathbb{R}^m$).

Questo tipo di visualizzazione è particolarmente utile per capire il comportamento dinamico (ed eventualmente caotico) del sistema che si studia: una successione di punti in questo spazio indica l'evoluzione del sistema nel tempo.

Qualora nello spazio degli stati, i punti formassero un pattern ben definito, si parla di *attrattore*, cioè un sottospazio vettoriale verso il quale il sistema evolve dopo un certo lasso temporale sufficientemente lungo.

È importante notare che la comparsa di un attrattore significa che, almeno a livello teorico, è possibile prevedere i futuri stati del sistema.

La tecnica più importante e più conosciuta per ricostruire lo spazio degli stati è detta delle "coordinate ritardate", secondo la quale si possono ottenere vettori in m dimensioni a partire da campioni ritardati temporalmente di una sequenza monodimensionale. Noi applicheremo questo metodo alla sequenza di IV che abbiamo collezionato (e che dunque è una sequenza discreta). In pratica, data una sequenza discreta di scalari $s[\cdot]$, posto n l'istante di osservazione e τ il ritardo con cui si vogliono estrarre le coordinate, il punto $\underline{x}(s,n)$ nello spazio degli stati m-dimensionale è dato da:

$$\underline{x}(s,n) = (s[n], s[n+\tau], s[n+2\tau], \ldots, s[n+m\tau])$$

La scelta del ritardo τ non dovrebbe influire sul risultato finale e quindi per semplicità abbiamo posto $\tau = 1$. Nel nostro caso, essendo $m = 3$ (vogliamo rappresentare la sequenza in uno spazio tridimensionale) avremo che:

$$\underline{x}(s,n) = (s[n], s[n+1], s[n+2])$$

Per quanto riguarda le NIC testate, abbiamo pensato di studiare la randomicità di quella che ci è sembrata essere caratterizzata dalla maggiore aleatorietà e di quella i cui IV sembravano abbastanza predicibili: la Cisco Aironet 350 e la Nortek W-11. Per tutte le analisi ci siamo serviti di circa 100.000 IV.

Lo spazio degli stati della Cisco è riportato in figura 2(a): si può osservare una buona casualità, ciononostante si può rilevare (sebbene non sia particolarmente visibile) la presenza di un attrattore (la diagonale del cubo che parte da $(0,0,0)$). Forse una leggera correlazione si poteva prevedere dall'analisi svolta nella sezione 2.1: gli IV della NIC in questione cambiavano in maniera aleatoria, ma i primi 4 bit del secondo byte erano sempre a zero.

Per quanto riguarda la Nortek, non ci sorprende vedere che il comportamento del suo PRNG sia di fatto assolutamente non casuale. Il risultato dell'analisi dello spazio delle fasi è riportato in figura 2(b): l'attrattore è presente (sempre la diagonale del cubo, che questa volta è molto ben definita).

Proponiamo inoltre anche un analisi del PRNG di Linux, accedendo alla variabile `$RANDOM` fornita dalla shell bash di Linux, per avere un confronto con quella che dovrebbe essere una buona aleatorietà. Lo spazio è coperto uniformemente dai punti e, ad una prima analisi, attrattori non sembrano essere presenti: la somiglianza con lo spazio degli stati della NIC Cisco è chiara. Questa è la caratteristica che un buon PRNG di una NIC dovrebbe avere.

2.3 Cracking della chiave WEP

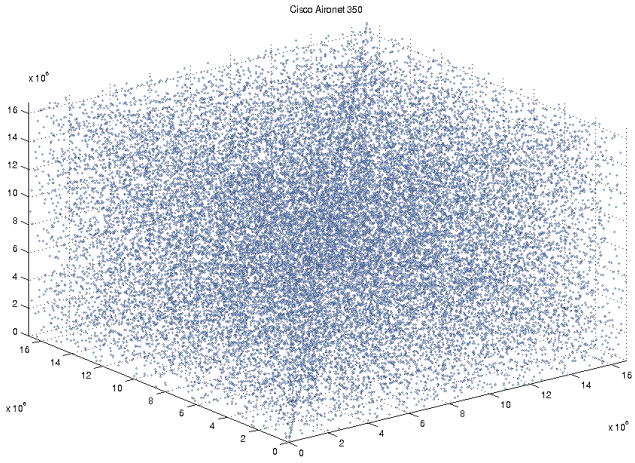
Visti i risultati della sezione 2.2 abbiamo voluto testare l'affidabilità effettiva delle NIC Cisco e Nortek e verificare se realmente la scarsa randomicità della Nortek potesse peggiorare ulteriormente la già scarsa sicurezza offerta dal WEP. Abbiamo quindi instaurato una rete "11b" infrastrutturata, in un primo tempo protetta da una chiave a 40 bit, poi da una a 104 bit.

Per generare molto traffico (senza dover usare Iperf) abbiamo pensato di ricorrere a scambi di pacchetti ICMP di tipo Echo Request / Echo Reply, cioè dei semplici "ping".

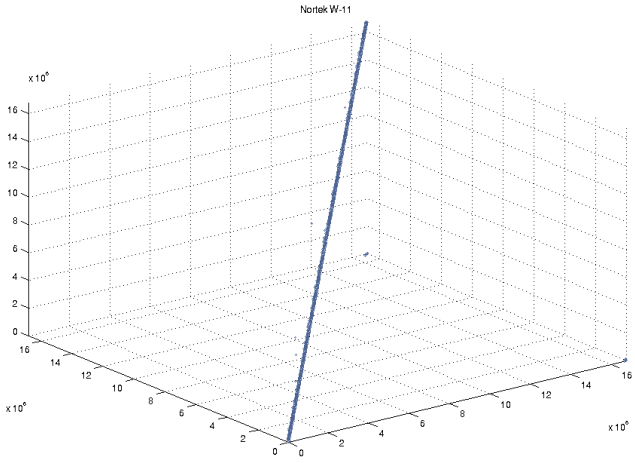
Ci siamo assicurati di trasmettere un'alta mole di traffico tramite il flag `-i 0.00001` del comando `ping`, che specifica l'intervallo di invio tra due Echo Request consecutive pari a 10 µs (chiaramente impossibile da proporre su un link wireless 802.11b).

In linea teorica, significa che vengono mandati 100.000 ping al secondo e, di conseguenza, 200.000 IV (perché si contano anche quelli dell'AP). Se però andiamo a stimare quanti pacchetti vengono trasmessi in secondo, vediamo che tale numero si dovrebbe attestare intorno a 1.000. La veloce stima è stata fatta sulla base delle seguenti considerazioni:

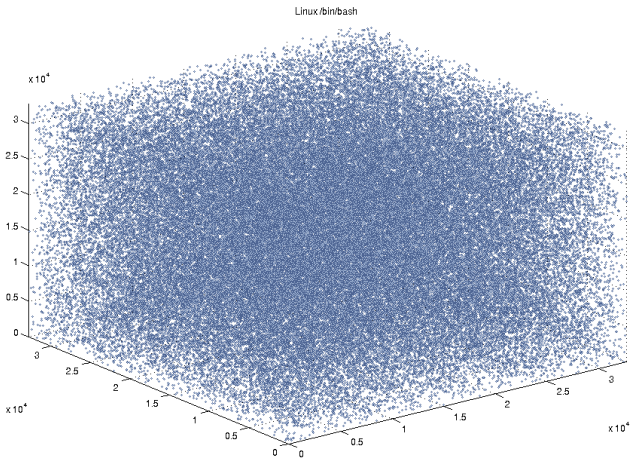
- La velocità di trasmissione è pari a 11 Mbps
- Viene usato il Short PLCP (96 µs di durata)
- SIFS dura 10 µs e DIFS 50 µs
- La procedura di Backoff viene stimata da un punto di vista statistico: $E[contention\_window] = 7.5$, valore che moltiplichiamo per la durata dello slot temporale (20 µs) per ottenere quindi un totale di 150 µs



(a) Spazio degli stati della NIC Cisco Aironet 350 ($m = 3, \tau = 1$)



(b) Spazio degli stati della NIC Nortek W-11 ($m = 3, \tau = 1$)



(c) Spazio degli stati del PRNG di Linux ($m = 3, \tau = 1$)

Figura 2: Analisi degli spazi degli stati

TS	11 Mbps
DIFS	$2 * 50 = 100$
BO	$2 * 150 = 300$
DATA	$2 * 186 = 372$
SIFS	$2 * 10 = 20$
ACK	$2 * 106 = 212$
TOT	1004
NS	996 s^{-1}

Tabella 6: Stima del numero di ping trasmessi in un secondo. TS indica la velocità di trasmissione; NS è la stima del numero di ping trasmessi al secondo. Dove non specificata, l'unità di misura è il µs.

- Di default, sotto Linux, Echo Request e Echo Reply generati da `ping` hanno una dimensione di 64 byte (8 di header e 56 di dati); contando anche 20 byte di header IP, 8 di LLC e 24 a livello MAC la dimensione totale del frame ammonta a 124 byte: dunque il tempo impiegato per trasmettere un ping è pari a $96\mu s + 124 * 8/11\text{Mbps} \approx 186$ µs
- l'ACK, lungo 14 byte, ha bisogno di $96\mu s + 14 * 8/11\text{Mbps} \approx 106$ µs per essere trasmesso

Poiché alla trasmissione di una Request segue sempre la Response, dobbiamo calcolare i tempi due volte (i dettagli sono riportati in tabella 6): circa mille sono i pacchetti ICMP che andremo a mandare al secondo (quindi più o meno 2.000 IV).

Abbiamo testato prima l'efficacia del tool di cracking: volevamo vedere quanti IVs sarebbero serviti per recuperare una chiave di 40 bit. Sulla homepage di aircrack si dice:

> Typically you need 250,000 or more unique IVs for 64 bit keys and 1.5 million or more for 128 bit keys. [...] The number of IVs is extremely hard to predict since some access points are very good at eliminating IVs that lead the WEP key.

Tramite airodump abbiamo catturato all'incirca 400.000 pacchetti, per un totale di 362.681 IV. Il cracking della chiave, effettuato per così dire "offline", cioè dopo che airodump è stato fermato, è riuscito in 1 secondo (tempo riportato dallo stesso software). Per altro abbiamo visto che venivano catturati all'incirca 20.000 pacchetti ogni mezzo minuto: tale dato è minore delle nostre stime (le nostre previsioni parlavano di 60.000 pacchetti ogni mezzo minuto), ma è comunque un valore ottimo.

Nei test a seguire abbiamo deciso di far partire aircrack ancora mentre airodump era in funzione, effettuando il recupero della chiave più o meno in real time.

Come viene suggerito sul sito, è bene non far partire il tool di cracking troppo presto, poiché, dopo aver esaurito gli attacchi di tipo statistico, passerà il resto del tempo ad effettuare un attacco di tipo bruteforce. Il loro consiglio è di aspettare di avere almeno 200.000 IV; noi abbiamo ritenuto di provare con un numero minore, in particolare abbiamo fatto partire aircrack dopo 100.000 IV per quanto riguarda i test con chiave a 40 bit e dopo 200.000 IV per i test con chiave a 104 bit. Riportiamo per ognuna tipologia di test solo una sessione d'esempio, essendo tutte le altre praticamente uguali.

Chiave a 40 bit

Nel caso della Nortek, per crackare la chiave sono bastati 136.872 pacchetti (128.030 IV utili): il tempo di cattura è stato di 3 minuti e mezzo.

Anche per la Cisco sono bastati circa 3 minuti e 40 secondi, ma il numero di pacchetti che aircrack ha usato per trovare la chiave è stato maggiore: 181.979 pacchetti, per un totale di 147.847 IV.

Prove successive hanno mostrato che nel caso si usi la NIC Cisco, il procedimento di cracking ha bisogno di un numero significativo di IV in più. Nella fattispecie ci vogliono circa 50.000 pacchetti in più (o equivalentemente 20.000 di IV). Non possiamo parlare in termini statistici in quanto non è stato effettuato un numero di prove sufficientemente elevato.

Chiave a 104 bit

Nel caso si usi la NIC Nortek, la cifratura riesce a tener testa per circa 9 minuti: ci vogliono 335.504 pacchetti (ed in particolare 246.019 IV) per permettere ad aircrack di effettuare il cracking. In figura 3 riportiamo l'esempio di una sessione volta a recuperare (offline) la chiave a 104 bit.

Molto meglio il caso della Cisco: il tool riesce a trovare la chiave solo dopo 14 minuti e mezzo, e

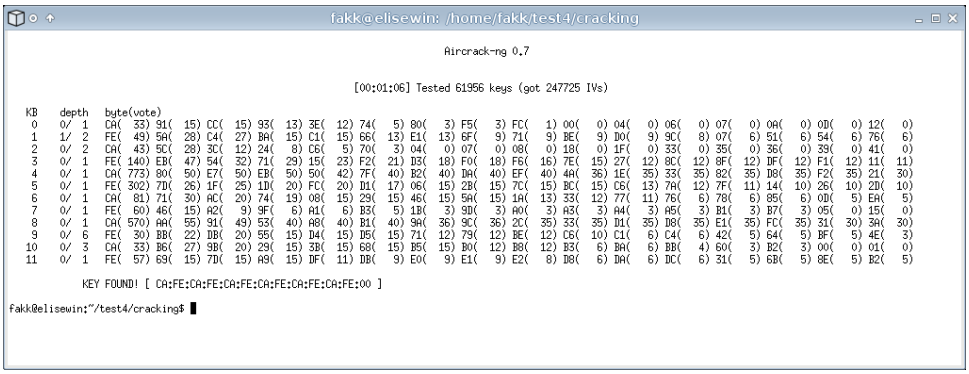


Figura 3: Screenshot di una sessione di cracking di una chiave a 104 bit.

cioè dopo che sono passati ben 757.231 pacchetti (493.628 IV).

Abbiamo notato che per trovare la chiave nel caso si usi la NIC Cisco si hanno tempi più lunghi (comunque variabili) e comunque un numero di IV molto maggiore di quelli che servono nel caso di sessioni in cui si fa uso della Nortek (come per altro avveniva nel caso in cui la chiave fosse di 40 bit).

Anche in questo caso non è possibile parlare da un punto di vista statistico; per questo sarebbe interessante effettuare sugli esperimenti effettuati un'analisi statistica approfondita.