Università degli Studi di Padova

Facoltà di ingegneria

Corso di Laurea in Ingegneria delle Telecomunicazioni

Studio di meccanismi di attacco e difesa nei collegamenti in ponte radio Hiperlan

Laureando

Ervin Konomi

Relatore

Prof. Nicola Laurenti

Referente aziendale

Marco Pisani

Anno Accademico 2012/2013

Alla mia famiglia e a tutti coloro
che mi hanno aiutato
in questa impresa!

Non hai veramente capito qualcosa
fino a quando non sei in grado di
spiegarlo a tua nonna!

Albert Einstein

Indice

Elenco delle figure

Sommario

Il testo esamina in modo approfondito gli standard di sicurezza *WEP* ed *IEEE 802.11i* folcalizzando l'attenzione sulle vulnerabilità che negli ultimi anni si sono scoperte su di essi. Prossegue poi con l'analisi degli attacchi sviluppati considerando l'implementazione di tali standard sulle reti wireless 802.11 con lo scopo di comprendere la reale possibilità di applicazione di tali attacchi sulle reti *Hiperlan 2* ed in particolar modo sui collegamenti punto punto.

Capitolo 1

Introduzione

1.1 Contesto

Uno degli aspetti che richiede sempre più maggiore attenzione, in una società come la nostra che si digitalizza in tempi rapidissimi ed in modo tumultuoso e disordinato riponendo sempre maggiore fiducia in internet e nei device mobili, è quello costituito dalla sicurezza delle informazioni. Se sino a qualche anno fa il problema sembrava limitato alle sole applicazioni informatiche, in questi anni non esiste settore, dai sistemi industriali agli apparati medicali, che non sia stato intaccato da qualche forma di attacco informatico. In questo contesto, lo strumento di cui maggiormente disponiamo per fronteggiare il rischio informatico è la consapevolezza della sua esistenza. Gestire l'impatto di eventi pianificati rispetto a quello di eventi inattesi è decisamente diverso. Sotto quest'ottica la 360 s.r.l in collaborazione sia con il mondo della ricerca industriale che accademica si propone di individuare gli strumenti necessari e le metodologie di prevenzione per garantire ai propri clienti la riservatezza e l'integrità dei loro dati.

1.2 Azienda

Nata come una branca della Sartori S.p.a, azienda leader del mercato italiano nel settore delle telecomunicazioni e dell'elettronica, la 360 s.r.l si presenta ai suoi clienti come una realtà giovane e dinamica. Ereditaria di un know-how di oltre 40 anni opera nei settori Ambiente, Energia, Sicurezza e Impianti di telecomunicazioni. Esperienza, cura per il dettaglio, passione per il lavoro nonchè la continua e periodica formazione tecnica del personale permettono alla 360 s.r.l di essere strategicamente competitiva sul mercato e di vantare

a suo seguito clienti sensibili come l'Esercito e le Forze di Polizia oltre che Pubbliche Amministrazioni ed enti privati.

1.3 Scopo della tesi

L'obiettivo che si propone di ragiungere la tesi consistene nel sensibilizzare la 360 s.r.l sul tema della sicurezza delle informazioni riguardante i collegamenti in ponte radio realizzati con un determinato tipo apparati e tecnologia. Si vuole mettere in risalto la semplicità con cui possono essere carpite informazioni nell'etere e che l'idea di sicurezza assoluta rimane solo un utopia teorica.

1.4 Organizzazione della tesi

Per facilitare la lettura e fornire le nozioni base per la comprensione della tesi si è pensato di organizzare il testo nel seguente modo:

- Capitolo 1: descrive il contesto e gli scopi per cui viene realizzata la tesi.
- Capitolo 2: provvede a introdurre i fondamentali della sicurezza oltre che a sintonizzare il linguaggio del lettore con quello del testo.
- Capitolo 3: fornisce una panoramica sullo standard utilizzato per realizzare la connettività wireless ed i principi fondamentali per i collegamenti in ponte radio.
- Capitolo 4: esamina in modo approfondito gli standard di sicurezza che vengono utilizzati in azienda per prottegere i link punto punto e le reti wireless.
- Capitolo 5: effettua un analisi approfondita sulle vulnerabilita degli standard di sicurezza quando sono applicati sulle reti 802.11 e degli attacchi scoperti negli ultimi anni. Inoltre valuta la possibilità che le reti Hiperlan 2 siano vulnerabili agli stessi attacchi.
- Capitolo 6: vengono effettuate delle valutazioni sugli standard di sicurezza.

Capitolo 2

Fondamentali sulla sicurezza

Termini come *minaccia* e *attaco* sono frequentemente utilizzati nel mondo della sicurezza delle informazioni. In letteratura molto spesso si tende ad utilizzarli, erroneamente, come sinonimi.
Con *minaccia* si intente una potenziale violazione della sicurezza dettata da una circostanza, una capacità, un'azione o un evento che potrebbe violare la sicurezza e provocare danni. Pertanto una minaccia è un pericolo potenziale che potrebbe sfruttare un punto debole. Per *attacco* si intende invece un assalto alla sicurezza di un sistema ovvero un tentativo deliberato di eludere i servizi di sicurezza e di violare la politica di sicurezza di un sistema.

2.1 Attacchi alla sicurezza

Con attacco alla sicurezza si intende qualsiasi azione che compromette la sicurezza delle informazioni di proprietà di un'organizzazione. Un modo utilizzato in letteratura per classificare gli attachi è quello di catalogarli in *attachi attivi* o *attachi passivi*. Un attaco passivo tenta di rilevare o di utilizzare le informazioni del sistema ma non agisce sulle risorse. Un attacco attivo tenta di alterare le risorse o comunque di alterarne il funzionamento.

2.1.1 Attacchi passivi

L'obbiettivo degli attachi passivi è quello di carpire le informazioni che vengono trasmesse e di conseguenza si preocupano di intercettare o monitorare le trasmissioni. A questo gruppo di attacchi appartengono l'*intercettazione del contenuto dei messaggie* l'*analisi del traffico*. Dalla Figura 2.1 è facile comprendere cosa sia l'itercettazione dei messaggi. Una conversazione telefonica,

una e-mail, o un file possono contenere informazioni delicate o confidenziali il cui contenuto non deve essere carpito durante le trasmissioni.

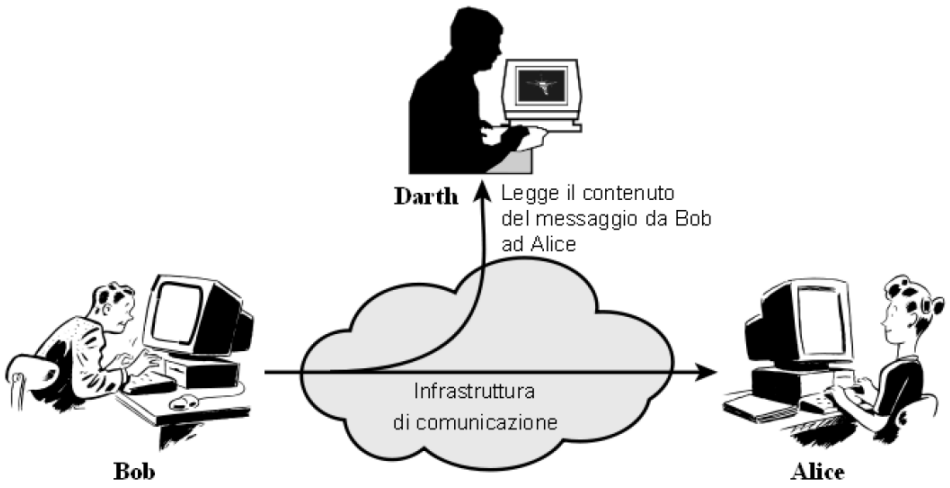

Figura 2.1: *Spionaggio del contenuto dei messaggi*

Il secondo tipo di attacco passivo, l'analisi del traffico, è più subdolo (figura 2.2). Anche non riuscendo a carpire il significato dei messaggi si potrebbe individuare comunque informazioni quali la posizione la frequenza e la lunghezza dei messaggi. Queste informazioni possono essere utili per scoprire la natura delle comunicazioni che si stano svolgendo.

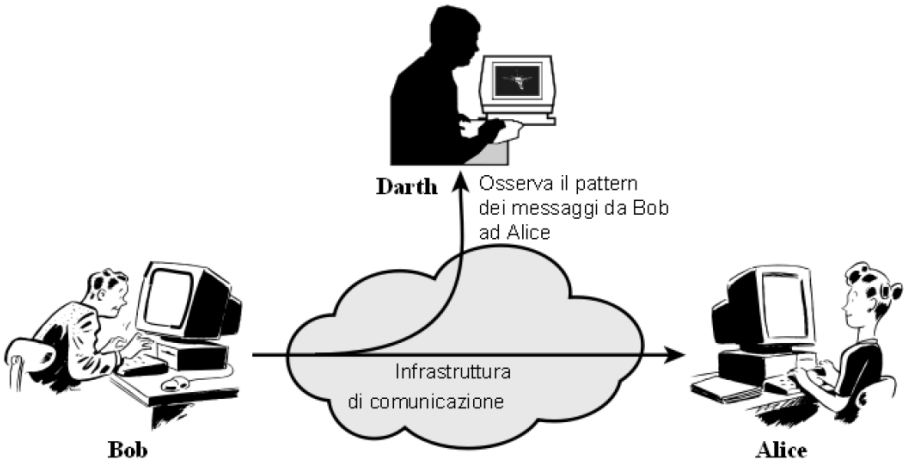

Figura 2.2: *Analisi del traffico*

Gli attacchi passivi sono molto difficili da rilevare poichè non comportano alcuna alterazione dei dati. In genere i messaggi vengono inviati e ricevuti in modo apparentemente normale, senza che il mittente e il destinatario possano rendersi conto che un estraneo ha letto i messaggi o ha osservato il pattern del traffico. Tuttavia è possibile impedire il sucesso di questi attacchi, solitamente utilizzando la crittografia. Questa tecnica ha lo scopo di mascherare i contenuti. Pertanto la difessa contro gli attacchi passivi riguarda più la prevenzione che la rilevazione.

2.1.2 Attacchi attivi

Gli attacchi attivi prevedono una modifica del flusso dei dati o la creazione di un falso flusso e possono essere sudivisi in quattro categorie:

- mascheramento
- ripetizione
- modifica dei mesaggi
- denial-of-service

Si ha un *mascheramento* quando una determinata entità finge di essere un 'altra entità (figura 2.3).

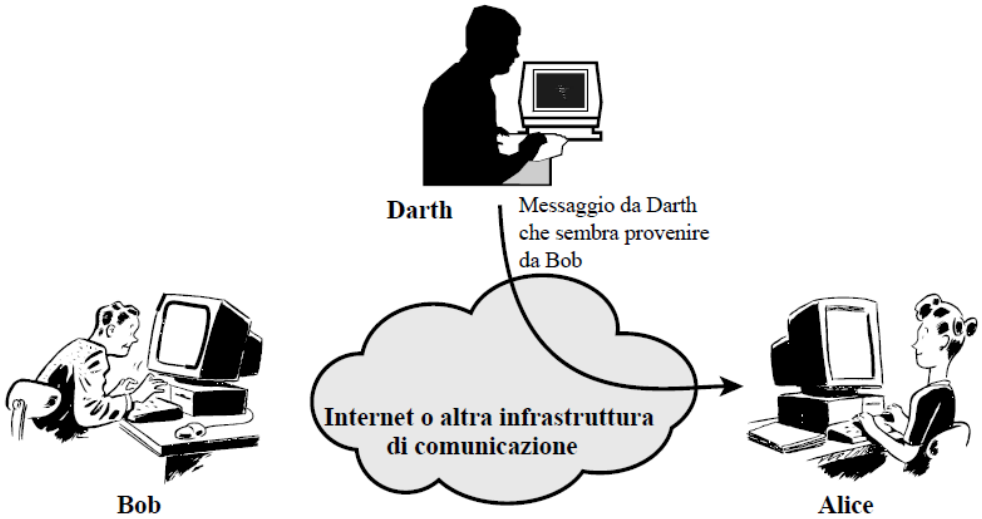

Figura 2.3: *Mascheramento*

La *ripetizione* prevede la cattura passiva di un'unità dati e la sua sucessiva ritrasmissione per produrre un effetto non autorizzato (figura 2.4).

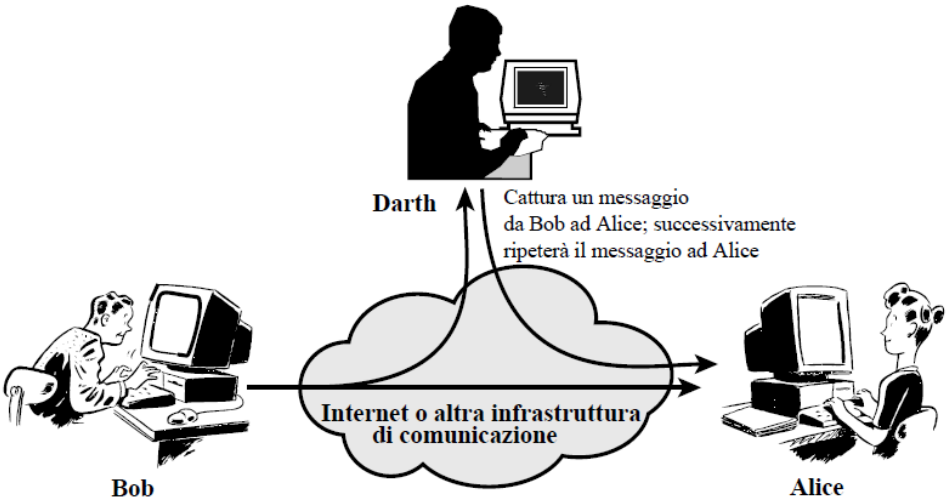

Figura 2.4: *Replay*

Con *modifica dei messaggi* si intende semplicemente l'alterazione di un messaggio legittimo o il ritardo o il riordino di messaggi per produrre effetti non autorizzati (figura 2.5).

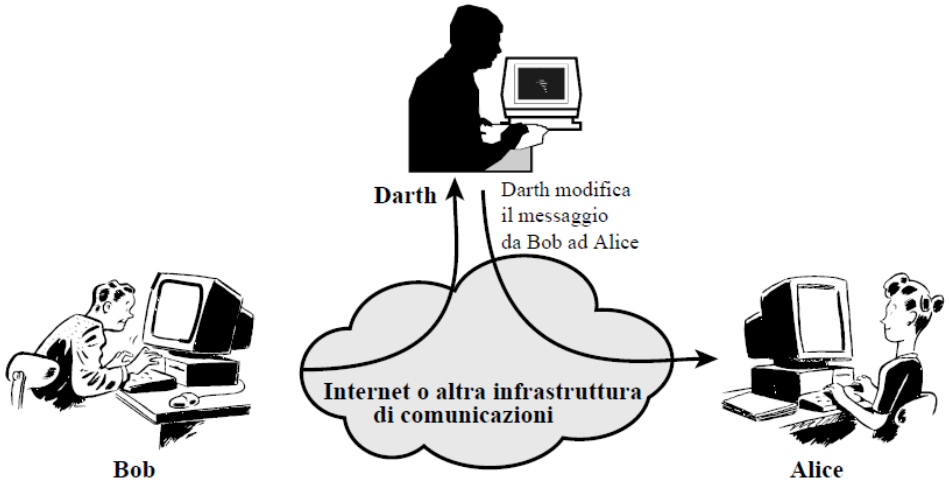

Figura 2.5: *Modifica dei messaggi*

Un attacco *denial-of-service* impedisce il normale utilizzo o la gestione di un sistema di comunicazione (figura 2.6). Questo tipo di attacco può avere un determinato bersaglio; per esempio soprimere tutti i messaggi diretti ad una destinazione oppure sovracaricare una rete di messaggi degradando cosi le sue prestazioni.

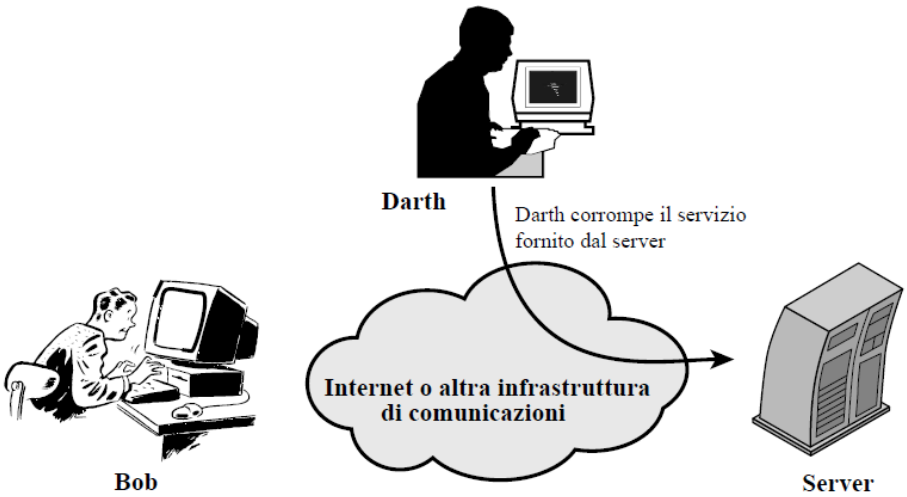

Figura 2.6: *Denial of service*

Gli attacchi attivi hanno caratteristiche opposte a quelli passivi. Gli attacchi passivi sono difficili da rilevare, ma esistono misure per prevenire il loro sucesso. Al contrario è piuttosto difficile prevenire in modo assoluto gli attachi attivi, per l'enorme varietà di potenziali lacune della sicurezza. In questo caso l'obiettivo è dunque quello di rilevarli e recuperare qualsiasi alterazione o ritardo da essi provocato. Poiché il rilevamento ha un effetto deterrente, può contribuire anche alla prevenzione.

2.2 Servizi di sicurezza

Un servizio di sicurezza è un servizio di elaborazione o comunicazione fornito da un sistema per offrire un determinato livello di protezione delle risorse del sistema. Questi servizi possono essere suddivisi nelle seguenti categorie cattegorie:

2.2.1 Segretezza dei dati

Prottegge i dati da qualsiasi accesso non autorizzato.

2.2.2 Integrità dei dati

Cerca di garantire che i dati ricevuti siano esattamente quelli inviatti dall'entità autorizzata, senza essere stati modificati in alcun modo (modifica, inserimento, cancellazione e ripetizione).

2.2.3 Disponibilità dei dati

E' la proprietà di un sistema o di una sua risorsa, di essere accessibile e utilizzabile su richiesta da parte di un entità autorizzata in base alle specifiche prestazionali del sistema (ovvero un sistema è disponible se fornisce i servizi previsti in base alle specifiche quando richiesti dagli utenti).

2.3 Meccanismi di sicurezza

Si intende qualsiasi processo (o dispositivo che incorpora tale processo) progettato per prevenire, rilevare o riprare i danni provocati da un attacco alla sicurezza. Importanti meccanismi di sicurezza sono la *crittografia* e l'*autenticazione*.

2.3.1 Crittogrfia

E'quella tecnica che, attraverso l'uso di algoritmi matematici, trasforma i dati in una forma illegibile. Attraverso la trsformazione inversa si riportano i dati allo stato originario.

2.3.2 Autenticazione

Cerca di garantire che le entità comunicanti siano realmente quelle che sostengono di essere.

2.4 Modello per la sicurezza di rete

La figura 2.7 mostra un modello che cattura, in termini molto generali i concetti di questo elaborato. Un messaggio deve essere trasferito dal mittente al destinatario attraversando una determinata rete. I due protagonisti di

questa transazione devono cooperare per poter eseguire l'operazione. Viene stabilito un canale logico per il trasferimento delle informazioni definendo un percorso attraverso l'internet dall'origine alla destinazione e tramite l'uso dei protocolli di comunicazione (come TCP/IP) da parte dei due protagonisti.

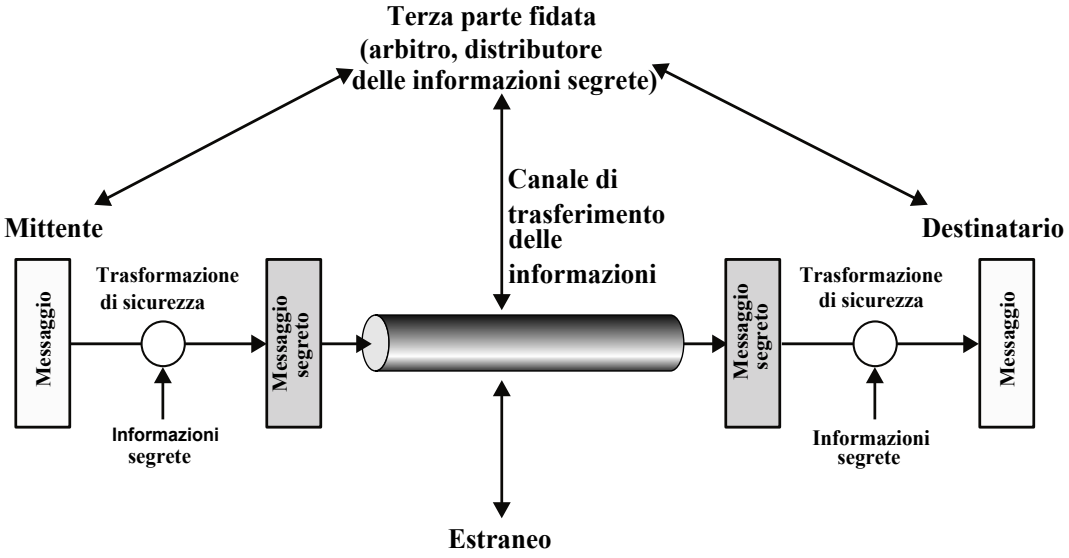

Figura 2.7: *Modello della sicurezza di rete*

Gli aspetti di sicurezza entrano in gioco quando è neccessario o desiderabile proteggere la trasmissione delle informazioni da chiunque possa rappresentare una minaccia alla segretezza, all'autenticità e cosi via. Tutte le tecniche per garantire la sicurezza hanno due componenti:

- Una trasformzazione di sicurezza delle informazioni trasmesse. Fra le varie possibilità vi sono la *crittografia* del messaggio, che codifica il messaggio in modo che risulti illegibile da parte di un estraneo e l'aggiunta di codice basato sul contenuto del messaggio, che può poi essere utilizzato per verificare l'identità del mitente.

- Un'informazione segreta condivisa dai due protagonisti e, si spera, sconosciuta agli estranei. Un esempio è la *chiave di crittografia* utilizzata per codificare il messaggio prima della trasmissione e per codificarlo dopo la ricezione.

Per garantire la sicurezza della trasmissione può essere necessario ricorrere a una terza parte fidata. Per esempio, questa terza parte può essere responsabile della distribuzione riservata dele informazioni segrete ai due protagonisti.

La terza parte potrebbe anche essere necessaria per arbitrare le dispute fra i due protagonisti riguardanti l'autenticità della trasmissione di un messaggio. Questo modello mostra che nella progettazione di un servizio di sicuezza vi sono quattro compiti fondamentali.

1. Progettazione di un algoritmo per l'esecuzione della trasformazione di sicurezza. L'algoritmo deve essere tale da impedire a un estraneo di conoscere il contenuto del messaggio.

2. Generazione dele informazioni segrete da utilizzare con l'algoritmo.

3. Sviluppo dei metodi per la distribuzione e la condivisione dell'informazione segreta.

4. Specifica del protocollo impiegato dai due protagonisti che utilizzano l'algoritmo di sicurezza e dell'informazione segreta per ottenere un determinato servizio di sicurezza.

Nei prossimi capitoli il lettore avrà modo di selezionare e distinguere chiaramente le varie componenti di queto modello, che anche se non estremamente preciso, ci aiuta nei nostri obbiettivi.

Capitolo 3

Standard per la connetività wireless

Le Wireless LAN (WLAN) sono reti wireless che forniscono coperture e servizi tipici di una LAN. Si tratta di reti in area locale in cui le stazioni terminali (e talvolta anche i nodi intermedi) usano collegamenti senza fili. Sono pensate come reti mobili, ma la mobilità è in genere intesa come relativamente lenta , ed il loro scopo principale è quello di liberare gli utenti da postazioni di lavoro fisse. Inoltre sonomolto usate anche come reti di accesso.

3.1 Hiperlan

Mentre si redigevano gli standard americani della IEEE, anche in ambito europeo è stata sviluppata dall'ETSI (European Telecommunications Standards Institute) una famiglia di standard finalizzata ad assicurare la connettività wireless a larga banda. Obiettivo principale dell'ETSI é stato definire HiperLAN (sigla per *high performance radio local area network*) che è una famiglia di standard Wireless LAN, o Radio LAN (RLAN) nel linguaggio ETSI, nelle intenzioni molto simile alla famiglia di standard IEEE 802.11. I lavori di standardizzazione sono stati portati avanti in principio da un comitato tecnico chiamato RES10 che ha emesso nel 1996 lo standard HiperLAN (type 1) e successivamente dal comitato BRAN (Broadband Radio Access Networks) che ha messo a punto gli standard : HiperLAN (type 2), HiperACCESS (type 3), HiperMAN (type 4), HiperLINK (type 5). L'obiettivo del BRAN é la predisposizione di standard wireless che consentano la realizzazione di collegamenti radio a velocità superiori a 25 Mbit/s sia in banda licenziata che in banda esente da licenza. Tali sistemi sono destinati ad utenze residenziali e aziendali che necessitino di collegamenti ad alta velocità di rapida installa-

zione e alternativi alle soluzioni cablate.
Come per altre tecnologie simili, anche gli standard sviluppati dal BRAN si limitano a definire gli aspetti relativi allo strato fisico ed a livello di accesso al mezzo. Inoltre in essi sono approntate alcune specifiche destinate a definire l'interoperabilità con soluzioni cablate basate su ATM e TCP/IP e con le reti mobili 3G.
Le differenti varianti che derivano dallo standard HiperLAN(type 1) sono orientate ad offrire servizi wireless con caratteristiche specifiche. Di seguito sono presentate le caratteristiche principali di ciascuno degli standard riservando particolare attenzione allo standard HiperLAN (type 2) utilizzato da 360srl.

3.1.1 HiperLAN/1.

Lo standard HiperLAN type 1 è stato concepito per fornire comunicazioni alla velocità di 20 Mbit/s tra terminali portatili nella banda dei 5 GHz. Tra gli obiettivi si aveva la creazione di reti wireless flessibili senza necessità di disporre di una struttura cablata preesistente. Analogamente alle soluzioni della famiglia IEEE 802.11, lo standard definisce solo lo strato fisico e lo strato MAC all'interno del quale é previsto un sottostrato, denominato CAC (channel access and control), dedicato alla gestione delle richieste di accesso al canale. Il protocollo EY-NPMA (elimination-yield non-preemptive multiple access) consente alla rete di operare con poche collisioni anche in presenza di numerosi utenti i quali, anche grazie a questo meccanismo, sono in grado di fruire di servizi multimediali. A livello MAC sono definiti i protocolli di instradamento, sicurezza e controllo del consumo energetico. Lo strato fisico utilizza modulazioni FSK e GMSK.

3.1.2 HiperACCESS

Concepito per accesso PMP ad alta velocità (120 Mbit/s) a grande distanza, da parte di utenti residenziali e utenti affari di piccole dimensioni, verso un'ampia varietà di reti, ivi incluso UMTS, le reti ATM e le reti basate sul protocollo IP. Uno scenario tipico nel quale HiperACCESS può essere usato, ad esempio, é per distribuire il traffico tra edifici della stessa azienda. I lavori di standardizzazione per HiperACCESS sono stati finalizzati alla realizzazione di soluzioni wireless operanti a frequenze oltre 11 GHz (ossia a 26, 28, 32 e 42 GHz) con elevata efficienza spettrale in condizioni LOS. Per canali di 28 MHz sono supportate modalità di condivisione del mezzo fisico sia in TDD che in FDD e in HFDD. L'obiettivo é consentire all'operatore di installare rapidamente su vaste aree una rete d'accesso a banda larga. Per questo

motivo HiperACCESS é ritenuto un'attraente alternativa all'accesso cablato (xDSL, cable modem) specialmente per operatori che sviluppino unicamente i servizi mobili e non siano dotati di infrastrutture fisse.

3.1.3 HiperMAN

Progettato per consentire l'accesso a larga banda ad utenza fissa mediante soluzioni wireless è compatibile con lo standard Wi-MAX per le applicazioni in reti MAN; é operativo a frequenze inferiori a 11 GHz (principalmente a 3,5 GHz). Sviluppato in cooperazione con il gruppo che si stava occupando dello standard IEEE 802.16 HiperMAN ed alcuni profili di IEEE 802.16a possono interoperare trasparentemente. Come gli altri standard definiti dall'ETSI anche HiperMAN è in grado di operare con reti ATM e IP. Lo standard prevede numerose categorie di servizio, offre la piena qualità di servizio, meccanismi di sicurezza, meccanismi adattativi per potenza, modulazione e codifica, la possibilità di collegamenti NLOS e meccanismi per la realizzazione di configurazioni Wireless Mesh. L'allocazione di frequenza può essere disciplinata sia in TDD che in FDD e i terminali hanno anche la possibilità di utilizzare la variante realizzativa H-FDD. Come per gli altri standard della famiglia ETSI le raccomandazioni si limitano a definire lo strato fisico e lo strato DLC.

3.1.4 HiperLINK

Finalizzata a ponti radio in grado di realizzare collegamenti dati punto-punto a larga banda tra due siti remoti a velocità di trasmissione fino a 155 Mbit/s che possono operare alle frequenze dei 17 GHz.

3.1.5 HiperLAN/2

Lo standard HiperLAN type 2, emesso nel 2000, é stato progettato per fornire bit rate molto elevati fino a 54 Mbit/s a livello fisico e fino a 25Mbit/s al 3° livello.
L'approccio di HIEPERLAN2, tipicamente orientato alla connessione, lo rende particolarmente adatto a supportare l'assegnazione di parametri specifici ad ogni connessione (larghezza di banda, ritardo, bit error rate, etc), in base alla qualità del servizio da essa richiesta e quindi al tipo di dati che devono essere trasmessi. Inoltre é possibile, seguendo un approccio più semplice, stabilire un livello di priorità tra le varie connessioni. Tali caratteristiche, combinate con le alte velocità raggiungibili, consentono la trasmissione simultanea di diversi tipi di dati (voce, video, dati).
I sistemi di comunicazione basati su HiperLAN/2 operano sulla banda dei 5

GHz (5,150-5,350 GHz , 5,470-5,725 GHz).
Possono essere installati in uffici, scuole, case, aziende industriali o in aree *hot-spot*, come centri comerciali o aeroporti, con l'obiettivo di fornire prestazioni confrontabili con quelle delle reti cablate.

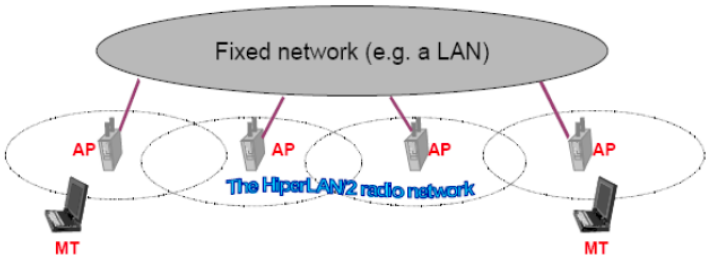

Figura 3.1: *Rete Hiperlan 2*

La struttura tipica di HIPERLAN/2 prevede dei terminali mobili (MT, *mobile terminal*) che comunicano via radio con un solo punto di accesso(AP, *Access point*) della rete fissa ed, in caso di movimento, il passaggio a punti di accesso adiacenti (*handover*) avviene in maniera automatica e in modo da garantire il miglior segnale radio misurato in termini di rapporto segnale/rumore. E' possibile anche la comunicazione diretta tra terminali mobili per mezzo di connessioni ad *hoc* (create al momento, con una durata strettamente necessaria allo scambio di dati). Tuttavia il supporto per la mobilita é garantito anche nel passaggio tra reti di tipo diverso (ad esempio quando il MT si muove da un LAN a WAN o da una rete aziendale privata ad una rete pubblica). Per offrire una copertura continua i punti di accesso devono avere aree di copertura che si sovrappongano.Tali aree presentano un raggio di circa 30 m in ambienti chiusi e di 150 m in ambienti all'aperto privi di ostacoli. Lo standard supporta la mobilità dei MT fino a velocità di 10 m/s. Inoltre in condizioni di LOS e antenne fortemente direttive è possibile raggiungere connessioni punto-punto di diverse decine di kilometri. L'Hiperlan 2 prevede due modalità di funzionamento:

- Centralized mode: Ogni AP si connette alla network core che serve le MT a lui associate. Tutto il traffico dei terminali mobili passa attraverso l'AP, sia che appartengano allo stesso AP, sia che appartengano a due core network differenti.

- Direct mode: L'accesso ai canali di trasmissione è ancora gestito in maniera centralizzata da parte dell'AP, ma il traffico dati avviene direttamente tra i terminali senza passare per l'AP. Questo modo viene

usato in ambiente particolarmente piccoli in cui si aspetta che la maggior parte del traffico avvenga tra terminali associati allo stesso AP. La comunicazione da MT a MT è definita DL (*Direct Link*). L'accesso al mezzo (aria) è gestito in maniera centralizzata con un CC (*central Controller*). Il traffico scambiato tra i terminali invece non passa dal CC.

Uno degli obiettivi delle specifiche tecniche di HIPERLAN2 è quello di far sì che il sistema operi in modalità Plug-and-Play e senza necessità di pianificazione frequenziale ed è a questo fine che le specifiche prevedono un meccanismo di DFS(*Dynamic Frequency Selection*). Lo scopo è quello di evitare le interferenze da parte di altri apparti, sia dello stesso tipo, sia di tipo diverso, che utilizzano lo stesso spettro di frequenze favorendone un uso il più possibile uniforme.
Hiperlan 2 implementa anche un meccanismo per il controllo della potenza di trasmissione (*transmit Power Control-TPC*) sia da parte del MT che da parte dell'AP. Il livello di potenza di trasmissione viene adattato in fuzione delle condizioni del canale. Lo standard definisce i seguenti limiti di potenza di trasmissione:

- 200 mW EIRP media (Equivalent Isotropic Radiated Power) nella banda 5.15-5.35 MHz, con uso esclusivamente in interni ed implementazione di DFS e TPC.

- 1W EIRP media nella banda 5.47-5.725 MHz, con uso in interni ed in esterni ed implementazione di DFS e TPC.

Lo standard si limita a definire i primi due livelli del modello ISO/OSI e le funzionalità necessarie per la convergenza con altre tecnologie di rete.

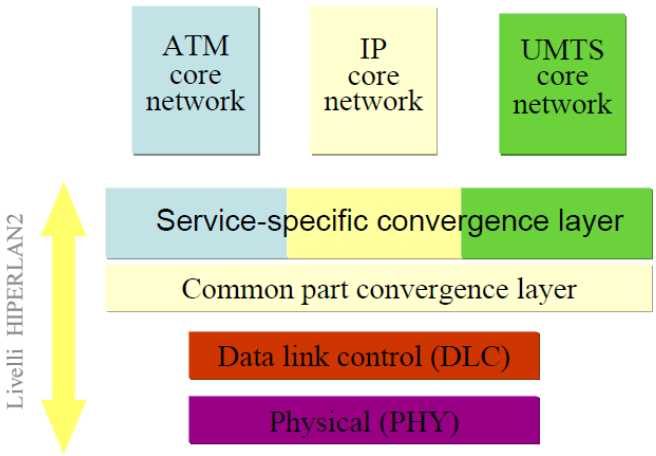

Figura 3.2: *Hiperlan 2 protocol stack*

Mediante il *Convergence Layer* (CL) supporta frame Ethernet, celle ATM, pacchetti IP e PPP. Per ciascun infrastrutura di rete supportata è definito un relativo CL. I vari CL operano uno alla volta. Come si può notare dalla figura 3.2 il CL è composto dal Common Part Convergence Sub-layer (CPCS) e dal Service Specific Convergence Sub-layer (SSCS). L'SSCS mappa, attraverso i suoi sub-layers (ATM SSCS, UMTS SSCS ..), le richieste di servizi dei livelli superiori ai servizi offerti dal livello DLC. Inoltre provede a convertire i pacchetti ricevuti dai livelli superiori in un formato adatto al CPCS e viceversa. Il CPCS converte i pacchetti (di lunghezza fissa o variabile) provenienti dai livelli superiori in Service Data Unit (SDU) di lunghezza fissa secondo le specifiche del DLC.

Il *Data Link Control* DLC costituisce il collegamento logico tra gli AP ed i MT. L'AP effettua uno scheduling centralizzato per assegnare dinamicamente le risorse, supportare la QoS e fornire trasmissioni collision-free.
E' composto da tre sottolivelli: L'Error Controll (EC), il Medium Access Control (MAC) ed il Radio Link Controll (RLC).
Il potocollo per la correzione dell'errore (EC) prevede diverse modalità di funzionamento: *acknowledged* , *unacknowledged* e *repetion*. La prima modalità prevede l'implementazione del protocollo Selective Repeat Automatic Repeat reQuest (SR-ARQ) per migliorare la qualità del collegamento e garantire trasmissioni affidabili.
La seconda modalità, a causa della mancanza del canale di feed-back, non garantisce la consegna dei pacchetti di conseguenza realizza un link inaffidabile.L'ultima modalità si bassa sulla ripetizione dell'intera DLC PDU per

garantire l'affidabilità della trasmissione. Non é disponibile alcun canale di risposta, ma il trasmettitore può arbitrariamente ritrasmettere le PDU per migliorare la ricezione. Il ricevitore accetterà comunque solo LE PDU il cui numero di sequenza rientra nella propria finestra di ricezione. La tecnica di rivelazione e correzione degli errori é di tipo FEC (Foward Error Crrection) basato su un codice Reed Solomon concatenato con un codice convoluzionale di Viterbi , che permette di raggiungere valori di BER (Bit Error Rate) molto bassi.
Il protocollo MAC gestisce l'accesso e la condivisione del canale da parte di una molteplicità di utenti. Si basa sui protocolli Time Division Duplex (TDD) e il Time Division Multiple Access che consentono comunicazioni unicast e multicast AP-MT, e anche comunicazioni MT-MT peer to peer.
L'RLC gestisce principalmente tre funzioni di controllo:

- Association Control Function (ACF): autenticazione, gestione delle chiavi, crittazione e associazione/diassociazione.

- DLC User Connection Control(DCC): setup/release delle connesioni dei MT

- Radio Resource Control(RRC): gestione dell'handover, DFS e TPC

Il *Physical Layer* (PHY) implementa come schema di modulazione l'OFDM. Prevede una funzione di link adaption per migliorare la capacità del collegamento radio alle differenti situazioni di interferenza e distanza tra AP ed MT. Il bit rate può variare tra 6Mbps e 54Mbps in funzione della combinazione dei fifferenti formati di modulazione (BPSK, QPSK, 16QAM, opzionalmente 64QAM) ed il tasso di codifica utilizzati. Questo implica che il PHY è in grado di operare in diverse modalità a secondo i termini di accordo tra MT e AP.

3.2 Ponte radio

Uno degli utilizzi più comuni di questo standard lo troviamo nella realizzazione dei link punto punto. In questa sede ci limiteremmo, data la natura del documento, a presentare le nozioni generali di questo particolare collegamento che altro non è che una rete wireless *ad hoc* tra due utenti che utilizzano antenne direttive.

3.2.1 Ellissoidi di Fresnel

Uno dei presupposti su cui si basa la tecnica dei collegamenti in ponte radio è che le due antenne siano in visibilità ottica (L.O.S, *Line Of Sight*). Per verificare la visibilità della tratta si applicano i principi dell'ottica geometrica, in cui l'onda che si propaga si considera puntiforme (si parla di raggio). In realtà l'energia e.m. occupa un certo spazio e occorre accertarsi che eventuali ostacoli non assorbano quantità di energia tali da compromettere il collegamento. Per fare ciò si analizza il 1° ellissoide di *Fresnel*, quello che racchiude la quantità significativa di energia, ritenendo che se lo spazio da esso circoscritto risulta libero da ostacoli allora il collegamento può essere assimilato a quello in spazio libero. In realtà, è stato dimostrato che se almeno il 60 % del volume generato dal primo ellissoide di Fresnel è libero da ostacoli la propagazione si può considerare di tipo L.O.S.

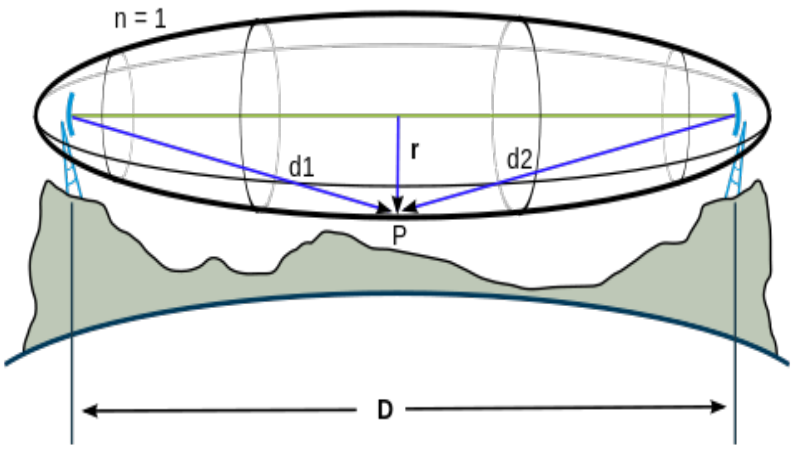

Figura 3.3: *Ellissoide di Fresnel: in verde è indicata la linea di vista, con D è indicata la distanza tra trasmettitore e ricevitore, con r è indicato il raggio sezione circolare.*

Una volta tracciata la line of sight, cioè il segmento che congiunge il dispositivo trasmettitore con il ricevitore, l'ampiezza dell' *n-esima* zona di Fresnel si calcola con la seguentee:

$$F_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}}$$

dove:

- F_n = il raggio dell'n-esima zona di Fresnel, in *metri*
- n = indice delle zone di Fresnel ($n = 1, 2, 3...$)
- d_1 = la distanza del generico punto P dal trasmettitore, in *metri*
- d_2 = la distanza del generico punto P dal ricevitore, in *metri*
- λ = la lunghezza d'onda, in *metri*.

Il raggio della sezione circolare in *metri*, r, della prima zona di Fresnel ($n = 1$) al centro della line of sight è

$$r = 17,32\sqrt{\frac{D}{4f}}$$

dove f è la frequenza espressa in GHz.

3.2.2 Link budget

Il modello di propagazione nello *spazio libero* viene impiegato per caratterizzare la potenza del segnale ricevuto, P_r, in ambienti in cui le onde possono propagarsi dal trasmettitore al ricevitore senza incontrare ostacoli sul loro cammino. La potenza di tale segnale decadrà in funzione del quadrato della distanza d che separa le due antenne secondo la cosidetta formula dell'ottimiso o di *Friss*:

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2$$

dove P_t è la potenza in trasmissione, G_t e G_r sono i guadagni d'antenna rispettivamente del trasmettitore e del ricevitore. Quindi abbiamo che:

$$[P_r]_{dBm} = [P_t]_{dBm} + [G_t]_{dB} + [G_r]_{dB} + [A_{sl}]_{dB}$$

dove :

$$[A_{sl}]_{dB} = 92,44 + 20log\,[d]_{Km} + 20log\,[f]_{GHz}$$

è l'attenuazine nello spazio libero. Come abbiamo specificato in precedenza la formula di Friss è ottimistica in quanto considera la potenza ricevuta in condizioni di massimmo (riguardo a impedenza di carico, direzione e polarizzazione). Nel mondo reale bisogna tenere conto anche di altre attenuazioni . Abbiamo perciò che l'attenuazione totale

$$[A_{tot}]_{dB} = [A_{sl}]_{dB} + [A_{go}]_{dB} + [A_{sup}]_{dB}$$

dove $[A_{go}]_{dB}$ è l'attenuazione introdotta dai dispositivi in guida d'onda mentre $[A_{sup}]_{dB}$ è un'attenuazione supplementare introdotta dal canale radio reale.
Principali cause dell'attenuazione supplementare sono:

- distorsioni dovute a riflessioni e a cammini multipli.
- attenuazione dovuta a fenomeni atmosferici in tratta (pioggia).
- interferenze con altri sistemi radio.

Teniamo presente che nel considerare le condizioni reali di propagazione si trascura il degrado della circuiteria elettronica interna agli apparati radio.

Le distorsioni dovute a riflessioni e a cammini multipli (fading selettivo), sono dovute essenzialmente a due cause:

1. irregolarità dell'indice di rifrazione atmosferico, che determina la curvatura della traiettoria seguita da parte del fascio.
2. la presenza del suolo, che determina la riflessione di parte del fascio.

In sintesi si può schematizzare il fenomeno come l'insorgenza nella propagazione di raggi riflessi e rifratti che si sommano in ricezione al raggio diretto. Poiché tali echi sono trasmessi e ricevuti entro angoli molto piccoli rispetto al raggio diretto, le ampiezze dei segnali sono dello stesso ordine di grandezza: la composizione può risultare distruttiva in fuzione della differenza di fase dei segnali.

L'ttenuazione dovuta a fenomeni atmosferici in tratta come la pioggia determina sull'onda e.m diversi effetti:

- assorbimento di energia e.m e che implica l'aumento dell'attenuazione in tratta.
- depolarizzazione della portante e scattering (cioè diffusione dell'energia nelle varie direzioni).

Il fenomeno dell' assorbimento aumenta rapidamente all'aumentare della frequenza, quando cioè la lunghezza d'onda diviene paragonabile alle dimensioni delle gocce d'acqua. In pratica il fenomeno produce effetti sensibili al di sopra dei 10GHz. Esistono delle carte in cui le zone sono classificate in funzione della piovosità, a seguito di studi per analizzare le proprietà delle precipitazioni: il tipo, la durata, le caratteristiche statistiche dell'intensità, la distribuzione

spaziale delle celle di pioggia.

Riguardo al fenomeno dell'interferenze con altri sistemi radio ci si può concentrare su due tipologie:

- interferenza isocanale.
- interferenza da canale adiacente

L'interferenza isocanale si presenta quando coesistono nello stesso centro radio portanti trasmesse sullo stesso canale, discriminate solo per la polarizzazione. Quando ciò avviene sulla stessa tratta i fadings sui due fasci risultano correlati, quindi il disaccoppiamento di polarizzazione rimane costante. Quando invece si riutilizzano i canali su tratte diverse, i fadings risultano incorrelati e la situazione risulta potenzialmente più pericolosa.
L'interferenza da canale adiacente si presenta quando nello stesso centro radio si utilizzano più canali appartenenti alla stessa gamma, in particolare quelli allocati in banda immediatamente prima ed immediatamente dopo il canale d'interesse.

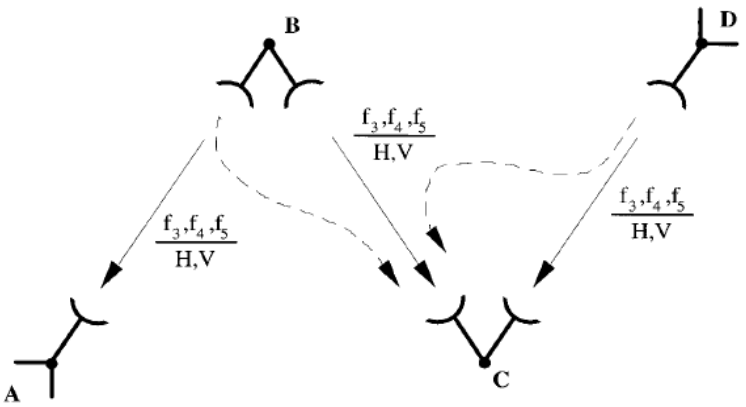

Figura 3.4: *Esempio di interferenti correlati ed incorrelati*

3.2.3 Antenne

Date le caratteristiche dei collegamenti in ponte radio si utilizzano antenne direttive, che sono in grado di concentrare la maggior parte dell'energia e.m. nella direzione voluta.
Le antenne maggiormente utilizzate sono quelle a paraboloide di rivoluzione, che grazie alla forma geometrica sono in grado di trasformare l'onda e.m.

emessa dall'illuminatore posto nel fuoco del paraboloide in un'onda piana nella direzione voluta. Possiamo esprimere il guodagno di queste antene secondo la seguente formula:

$$[G]_{dB} = 10log\left(\frac{\pi^2}{\lambda^2 D^2 \eta}\right)$$

dove D è ildiametro dell'apertura del paraboloide e η è l'efficienza dell'antenna (valori tipici compresi tra 0,5 e 0,65). Quindi il guadagno d'antenna aumenta all'aumentare delle sue dimensioni e della frequenza.

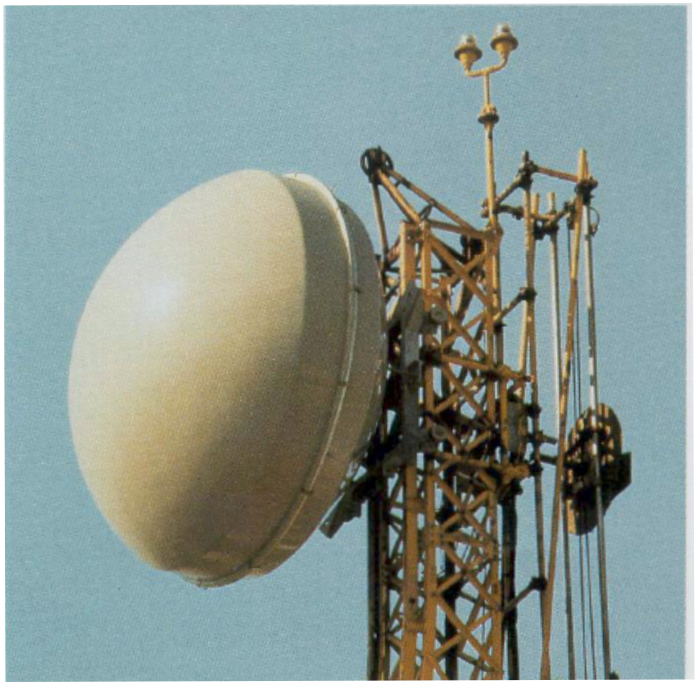

Figura 3.5: *Esempio di antenna*

Altri parametri caratteristici delle antenne (angolo di apertura, attenuazione dei lobi secondari, attenuazione front-back) possono essere ricavati dal diagramma di radiazione che in questa sede non analizzeremo visto il nostro scopo.

3.2.4 Progettazione del link

In seguito viene riportata una sequenza di punti da seguire al fine di realizzare i collegamenti in ponte radio:

- Per prima cosa occorre verificare la visibilità ottica delle due stazioni, valutando eventuali ostacoli intermedi mediante l'ellissoide di Fresnel, stabilendo così la quota ottimale su cui posizionare le antenne.

- Successivamente si sceglie la gamma di frequenza in funzione della distanza del collegamento. Fissata la gamma, si scelgono i canali più opportuni al fine di evitare interferenze con gli altri sistemi attivi e noti.

- A questo punto si calcola l'attenuazione di tratta in spazio libero a cui si aggiunge l'ulteriore attenuazione introdotta dai sistemi in guida d'onda.

- Dalle specifiche di apparato sono noti i valori di potenza in trasmissione e la sensibilità del ricevitore alla soglia P_s. Fissando il margine M per fadings supplementari (multipath e pioggia) che vogliamo ottenere sulla tratta, otterremo la potenza di segnale minima P_{rm} ammessa sul ricevitore:

  $$[P_{rm}]_{dBm} = [P_s]_{dBm} + M$$

  da cui si ottiene quanto deve valere la somma dei guadagni delle due antenne:

  $$[G_t]_{dB} + [G_r]_{dB} = [A_{sl}]_{dB} + [A_{go}]_{dB} - [P_t]_{dBm} + [P_{rm}]_{dBm}$$

In effetti il margine di tratta, fissata la gamma di utilizzo, si ricava da diagrammi che forniscono i valori da rispettare in funzione della distanza e della zona (piovosità).

Capitolo 4

Standard per la sicurezza

Prima di introdurre gli standard di sicurezza utilizzati da 360srl per potteggere i suoi collegamenti presentiamo brevemente i dispositivi utilizzati da questi ultimi.
Data la particolarità dei clienti (*polizia*, *carabinieri*, *procure*) per cui opera l'azienda in questa sede si è convenuto a non rendere noto i modelli dei dispositivi utilizzati per la realizzazione dei collegamenti in ponte radio. Si è piuttosto preferito proporre lo stack protocollare che essi implementano. Attraverso tale stack si andranno ad analizzare i vari protocolli di sicurezza (dichiarati nei datasheet) utilizzati da 360srl.
La figura 4.1 mostra l'architettura dell'Internet Protocol Suite (TCP/IP) a confronto con il modello di riferimento ISO/OSI.

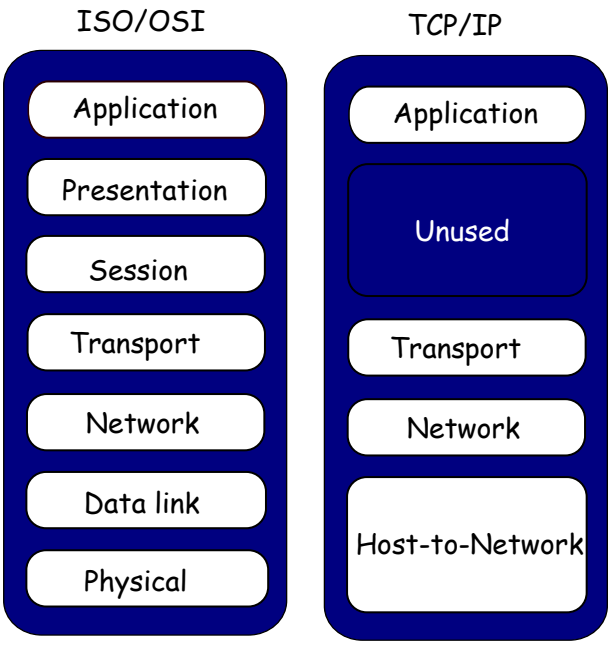

Figura 4.1: *Stack protocollare dei dispositivi*

I nostri dispositivi implementano proprio l'architettura TCP/IP che attualmente risulta la più diffusa. In tale architettura i livelli *Session* e *Presentation* vengono inclusi nel livello *Application*. Il livelo *Network prevede* l'uso del protocollo IP mentre il livello *Transport* TCP/UDP. L'rchitettura TCP/IP non definisce i primi due livelli del modello di riferimento. Lascia ampia possibilità di scelta sulla tecnologia da utilizzare. Infatti tutto ciò che si assume è la capacità del dispositivo di inviare pacchetti IP sulla rete. Qesta capacità nel nostro caso è garantita dallo standard HIPERLAN/2.
I protocolli per la sicurezza che andremmo ad analizzare sono posizionati sotto il livello MAC. Infatti essi agiscono proprio sulle MPDU (*MAC Protocol Data Unit*).

4.1 WEP

A liello di collegamento dati i dispositivi prevedono diverse modalità di protezione. Partiamo con il WEP ossia il *Wired Equivalent Privacy*.
Questo protocollo di sicurezza era stato progettato con l'obiettivo di raggiungere un livello di affidabilità pari alle reti cablate ethernet. Il primo passo dell'algoritmo prevede che la *chiave segreta* K sia concatenata con un *initialization vector* (IV) e che la stringa risultante costituisca il seme (*seed*) di un generatore di numeri pseudocasuali, chiamato *pseudo random number*

generator (PRNG). L'output del PRNG è un *keystream* k la cui lunghezza é esattamente uguale a quella del messaggio che sarà trasmesso in rete. Tale generatore, nel caso dell'WEP, è il famoso algoritmo crittografico $RC4$ (*Rivest Chipher 4*). Per evitare la manipolazione del messaggio durante la trasmissione, gli si applica un algoritmo di controllo di integrità. Nel caso specifico del WEP si usa CRC32. Il risultato di questa operazione viene chiamato *Integrity Check Value* (ICV) ed è concatenato al messaggio stesso. Il processo prosegue eseguendo lo XOR tra il *keystream* k ed il blocco del *plaintext* concatenato con l'ICV dando origine al testo cifrato denominato *ciphertext*. Il messaggio finale, pronto alla trasmissione, sarà ottenuto concatenando al ciphertext l'IV (in chiaro) necessario per la decodifica da parte del destinatario.

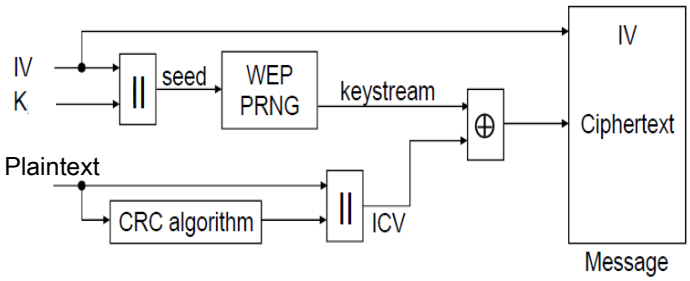

Figura 4.2: *Diagramma della codifica WEP*

La fase di decodifica del protocollo WEP, schematizzata in figura 4.3, prevede una fase iniziale in cui verrà generato lo stesso *keystream* k utilizzato per la codifica. Questo avviene prendendo l'IV del messaggio ricevuto, concatenandolo alla chiave segreta, K, ed infine utilizzando la stringa ottenuta come input del PRNG. Quindi si calcolerà lo XOR tra il *keystream* k ottenuto ed il messaggio cifrato. Infine la corretta decifratura deve essere verificata dall'algoritmo CRC32 sul testo in chiaro recuperato. Si controlla se l'ICV contenuto nel messaggio ricevuto corrisponda esattamente allo all'ICV appena calcolato.

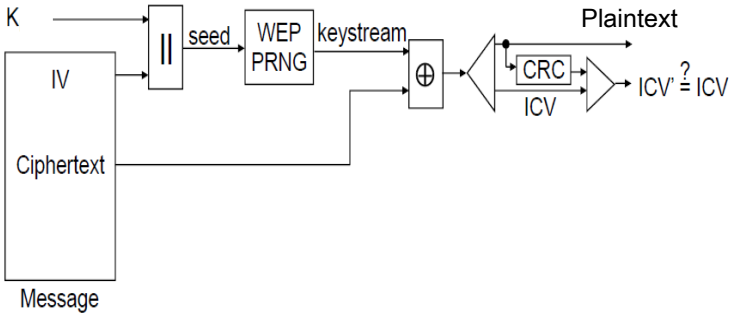

Figura 4.3: *Diagramma della decodifica WEP*

Il protocollo WEP è stato specificatamente progettato per rafforzare tre aspetti delle WLAN:

- **Riservatezza** L'obiettivo fondamentale del WEP è prevenire le intercettazioni casuali. La riservatezza é ottenuta codificando con l'algoritmo RC4 i pacchetti inviati.
- **Access Control** secondo obiettivo del protocollo è di proteggere l'accesso non autorizzato alla rete wireless. Lo standard include infatti una caratteristica che permette di scartare tutti i pacchetti che non sono adeguatamente codificati con WEP.
- **Data Integrity** Un obiettivo correlato è di prevenire l'alterazione dei messaggi trasmessi, il campo di integrity checksum ICV é incluso proprio per questo scopo.

Il protocollo WEP è stato progettato come strumento per salvaguardare la riservatezza dei pacchetti trasmessi in rete. Prevede che ogni pacchetto inviato sia codificato utilizzando una chiave segreta, di 40 bit o 104 bit, preceduta da un *initialization vector IV* di 24 bit e specifico di ogni pacchetto trasmesso. La stringa così ottenuta è utilizzata come chiave per l'algoritmo *RC*4. Attualmente ci sono due metodi di implementazione del WEP: il metodo che prevede l'uso di chiavi da soli 40 bit e una versione estesa sviluppata da molti produttori per permettere l'uso di chiavi lunghe 128 bit (in realtà 104 bit) in modo da rendere più arduo un attacco a forza bruta classico.

4.1.1 RC4

RC4 è stato sviluppato nel 1987 da Ron Rivest per RSA Data Security, Inc. Per sette anni l'algoritmo è stato proprietario e i dettagli venivano resi

disponibili solo dopo la sottoscrizione di un accordo di non divulgazione. Nel settembre del 1994, è stato inviato in modo anonimo il codice sorgente dell'algoritmo sulla mailing list di Cypherpunks, diffondendosi velocemente in tutto il mondo. Alcuni lettori già in possesso del codice originale hanno confermato la compatibilità con il codice online. L'algoritmo $RC4$ è di facile descrizione: esso utilizza una substitution box ad 8 bit: $S_0, S_1, .., S_{255}$. La quale è inizializzata linearmente: $S_0 = 0, .., S_{255} = 255$. Oltre all'array S è necessario un altro array K di 256 byte che viente inizializzato con i valori della chiave, ripetendola nel caso la lunghezza fosse inferiore. L'ultima fase dell'inizializzazione prevede l'esecuzione di questo algoritmo, che permette di distribuire i valori di S:

$$
\begin{aligned}
&j = 0 \\
&\text{for } i = 0 \text{ to } 255 : \\
&\qquad j = (j + S_i + K_i) \bmod 256 \\
&\qquad \text{swap } S_i \text{ and } S_j
\end{aligned}
$$

Per generare un qualsiasi byte del *keystream*, si effettuano le seguenti operazioni:

$$
\begin{aligned}
&i = (i + 1) \bmod 256 \\
&j = (j + S_i) \bmod 256 \\
&\text{swap}(S_i, S_j) \\
&t = (S_i + S_j) \bmod 256 \\
&k = S_t
\end{aligned}
$$

Come si può notare, sono necessari due contatori, indicati con i e j ed inizializzati a zero. Per produrre codice cifrato è sufficiente calcolare lo XOR fra il byte k ed il corrispondente byte del testo in chiaro, così come per ritornare al testo in chiaro si farà lo XOR fra k e il codice cifrato. La substitution box si evolve lentamente con l'uso: i garantisce che ogni elemento sia variato, mentre j assicura che gli elementi cambino in modo casuale. La versione ufficiale di $RC4$ precedentemente descritta è a 8 bit. Mentre si potrebbe pensare di definirne una versione con substitution box a 16 bit. La RSA Data Security ha dichiarato che l'algoritmo $RC4$ è immune da criptanalisi sebbene tale dichiarazione non sia stata verificata.

4.1.2 CRC-32

Il CRC (*Cyclic Redundancy Check*) è codice a rilevazione di errore ed ha lo scopo di permettere al ricevente di determinare se, in modo volontario o meno, c'è stata un'alterazione del messagio durante la trasmissione. Per ottenere ciò, in trasmissione si concatena al messaggio l'ICV (bit di *controllo*) in funzione del messaggio stesso. In ricezione si ricalcola l'ICV sul messaggio e lo si calcola con quello ricevuto. Se risultano uguali il messaggio allora non è stato alterato. Analiziamo un pò più dettagliattamente come avviene tutto il processo. Il CRC considera il *plaintext* sottoforma di polinomio avente come coeficienti le cifre binarie del messaggio. Una sequenza di N bit può essere rappresentata tramite un polinomio a coeficienti binari, di grado pari a $N-1$, tale che i suoi coeficienti siano uguali ai valori dei bit della sequenza. Sia

$$M(x) = b(N-1) * x^{N-1} + .. + b1 * x + b0$$

il polinomio del messaggio. Sia $G(x)$ un polinomio generatore, a coeficienti binari, scelto in base agli standard dettati dal CCITT (*Comité Consultativ International Télégraphique et Téléphonique*). La sigla numerica accanto all'acronimo CRC indica il numero di bit di controllo da aggiugere alla fine del messaggio. Nel caso del protocollo WEP, vista la criticità dei dati inviati, il grado di precisione del controllo deve essere elevato e dunque viene utilizzato un polinomio generatore a 32 bit:

$$\begin{aligned} G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} \\ + x^{12} + x^{11} + x^{10} \\ + x^7 + x^5 + x^4 + x^2 + x + 1 \end{aligned}$$

Vengono aggiunti bit ridondanti in coda al messaggio $M(x)$ per renderlo sempre divisibile per $G(x)$, ovvero tale che il resto $R(x)$ della divisione sia sempre uguale a zero. E' necessario ora moltiplicare il messaggio $M(x)$ per un polinomio di grado 32, ovvero aggiungergli in coda 32 zeri. Chiamiamo tale messaggio $M'(x)$. Dividiamo ora $M'(x)$ per il polinomio $G(x)$. Ovvero

$$M'(x) = G(x) * Q(x) + R(x)$$

dove $Q(x)$ è il quoziente della divisione e $R(x)$ il resto. Dalla formula precedente si ricava il resto

$$R(x) = M'(x) - G(x) * Q(x)$$

e si compone il definitivo messaggio $T(x)$ che verrà trasmesso nel seguente modo:

$$T(x) = M'(x)R(x)$$

concatenando ad $M'(x)$ il resto della divisione per $G(x)$. Il destinatario del messaggio dovrà eseguire le stesse operazioni e confrontare il resto della divisone $R(x)$ con quello ricevuto. Se tale resto risulterà diverso da zero, nel messaggio ricevuto si è verificato un errore, in caso contrario accetterà il pacchetto.

Autenticazione

Prima che la stazione possa comunicare all'interno della rete, ha bisogno di autenticarsi per diventare associata a un determinato Access Point. WEP supporta due tipi di autenticazione:

- *Open system authentication*: qualsiasi stazione che voglia autenticarsi con un determinato AP (sul quale è stato impostato questo tipo di autenticazione) può effettuarlo senza problemi. Il protocollo è costituito semplicemente da un frame di *Authentication Request* e uno di *Authentication Response*.

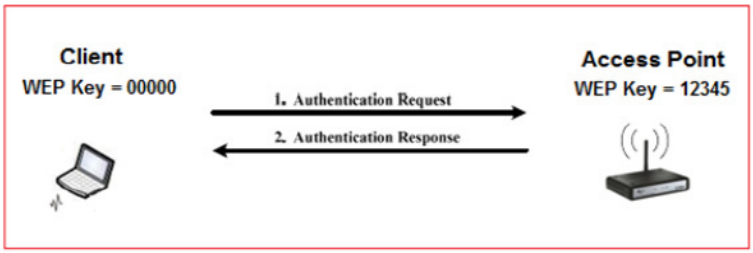

Figura 4.4: *Autenticazione a sistema aperto*

  Consiste semplicemente nello scambio delle reciproche identità tra Access Point e stazione e non offre alcun aiuto dal punto di vista della sicurezza. Se è attiva la crittografia WEP, la stazione può terminare l'autenticazione con l'Access Point ma può inviare/ricevere dati solo se possiede la chiave WEP corretta.

- *Shared key authentication*: fornisce una oneway authentication. E' la stazione ad autenticarsi con l'Access Point e non il contrario. Solamente le stazioni che conoscono la chiave segreta sono capaci di autenticarsi con l'AP.

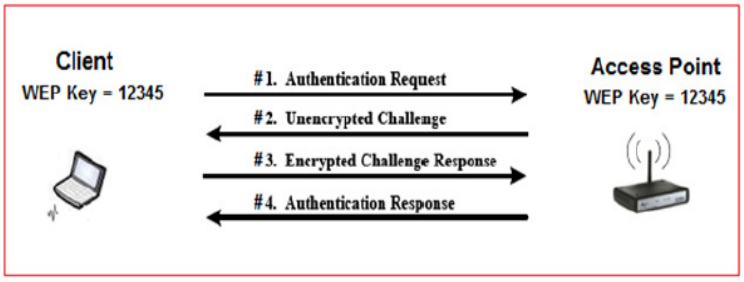

Figura 4.5: *Autenticazione a chiave condivisa*

Il protocollo consiste in un 4-Way Handshake e inizia con un'*Authentication Request* (#1) da parte della stazione. L'Access Point a sua volta risponde con un *challenge* (#2) che contiene un messaggio di 128 byte fornito dal generatore di numeri pseudo casuali. Ricevuto il *challenge* di 128 byte, questo è cifrato usando WEP con la chiave segreta condivisa e quindi inviato (#3) di nuovo all'Access Point. L'Access Point, ricevuto questo messaggio, lo decodifica e ne controlla l'ICV. Se il controllo va a buon fine, il contenuto decifrato è confrontato con il c*hallenge* precedentemente inviato. Se sono identici, l'AP sa che la stazione conosce la chiave condivisa e così invia un'*authentication success* message (#4) alla stazione.

Nonostante questi accorgimenti, come vedremo nei prossimi paragrafi, il protocollo sarà ampiamente violato.

4.2 IEEE 802.11i

Per rispondere alle giustificate preoccupazioni delle aziende in merito alla sicurezza wireless offerta dal protocollo WEP, l'IEEE il 24 Giugno del 2004 ha ratificato il nuovo standard 802.11i. Lo standard IEEE 802.11i definisce un'architettura di sicurezza scalabile che comprende politiche per l'autenticazione, la gestione delle chiavi e la segretezza ed integrità dei dati. Quest'architettura di rete wireless é utilizzabile sia in grandi reti aziendali che in reti domestiche ed é chiamata *Robust Security Network* (RSN). Anche se la nuova architettura é molto complessa essa però fornisce delle soluzioni sicure per le comunicazioni wirless. IEEE 802.11i definisce anche un'architettura Transitional Security Network (TSN) in cui possono partecipare sia RSN sia i sistemi WEP, consentendo agli utenti di aggiornare i loro strumenti senza

doverne acquistare dei nuovi. Se le procedure di autenticazione o di associazione usate tra le stazioni utilizza una handshake a 4 vie, l'associazione è detta *Robust Security Network Association*(RSNA). La comunicazione, come si può notare nella figura 4.6 avviene attreverso 4 fasi :

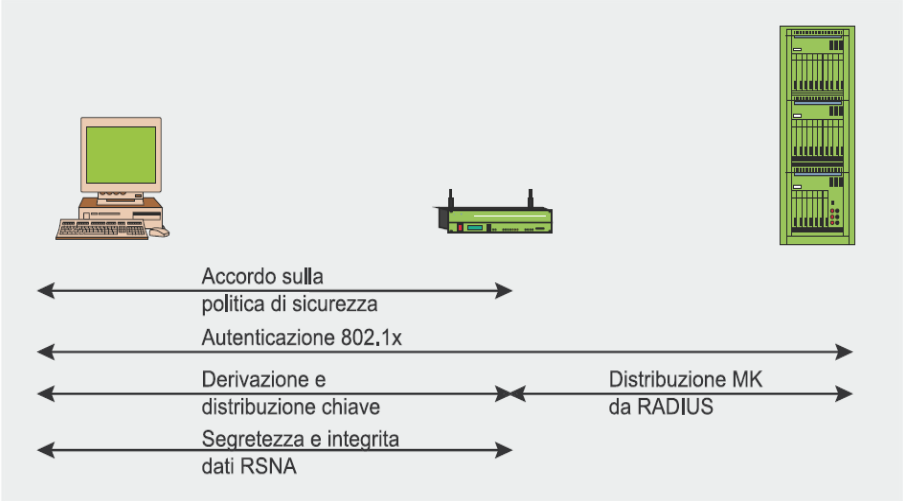

Figura 4.6: *Le fasi operative del 802.11i*

- accordo sulla politica di sicurezza
- autenticazione (802.1X o Pre-Shared Key (PSK))
- derivazione e distribuzione di chiave
- segretezza e integrità dei dati RSNA.

In seguito si procede all'analizi di ciascuna fase.

Fase 1: accordo sulla politica di sicurezza

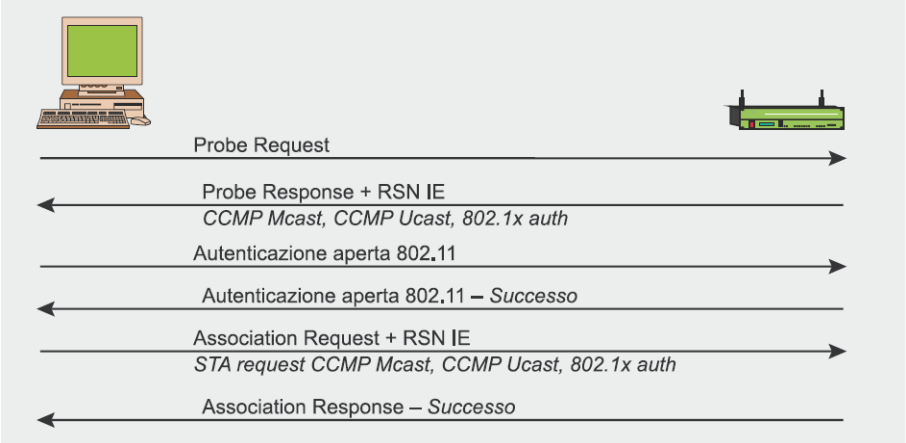

Figura 4.7: *Accordo sulla politica di sicurezza*

La prima fase richiede alle parti coinvolte nella comunicazione di stabilire un accordo sulla politica di sicurezza da adottare. Le politiche di sicurezza supportate dal punto di accesso sono pubblicizzate su *Beacon* o in un messaggio Probe Respond (seguendo un Probe Request dal client). Ne segue un'autenticazione aperta standard (dove l'autenticazione va sempre a buon fine e non offre alcun vantaggio in termini di sicurezza ma semplicemente fornisce retro compatibilità con lo standard IEEE 802.11). La risposta del client viene incluso nel messaggio di tipo Association Request confermato dalla risposta Association Response dal punto di accesso. Le informazioni sulla politica di sicurezza è inviata nel campo RSN IE (*Information Element*) con i dettagli su :

- metodi di autenticazione supportati (802.1X, chiave segrete precondivisa (PSK))
- protocolli di sicurezza per traffico unicast (CCMP, TKIP ecc.)
- protocolli di sicurezza per il traffico multicast (CCMP, TKIP etc.)
- supporto per la pre-autenticazione, permettendo agli utenti di preautenticarsi prima di passare ad un nuovo punto di accesso della stessa rete per un passaggio senza interruzioni

Fase 2 : autenticazione
Concordato le politiche di sicurezza da adottare ha inizio il processo di autenticazione fondamentale per prevenire accessi non autorizzati alle risorse della rete.
Prima di analizzare la seconda fase facciamo una piccola digressione su questo standard.
Lo standard IEEE 802.11i prevede l'uso di IEEE 802.1X (noto anche come Port-Based Network Access Control) per l'autenticazione. 802.1X è un framework sviluppato in origine per le reti wired. Fornisce inoltre meccanismi di distribuzione delle chiavi, autorizzazione e cotrollo degli accessi. L'architettura IEEE 802.1X si compone di tre entità principali:

- il supplicant che entra nella rete (client)
- l'autenticatore che fornisce il controllo sugli accessi (access point)
- il server di autenticazione che esegue le autorizzaz (RADIUS)

La figura 4.8 illustra come queste tre entità comunicano tra loro.

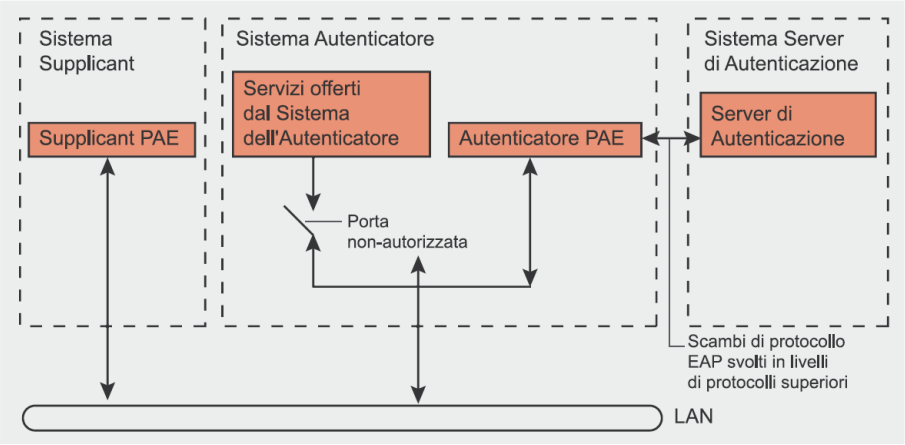

Figura 4.8: *Architettura 802.1X*

Ogni porta fisica (porta virtuale nelle reti wireless) si divide in due porte logiche che compongono il PAE (Port Access Entity). L'autenticazione del PAE è sempre aperta e permette di far passare le trame di autenticazione mentre il servizio PAE viene aperto solo in seguito ad una autenticazione andata a buon fine (come in uno stato autorizzato) per un certo periodo di tempo (3600 secondi di default). La decisione di permettere l'accesso è eseguita di solito dalla terza entità, cioè il server di autenticazione (che può

essere un server Radius dedicato o, per esempio nel caso di reti domestiche, un processo che gira su un acces point). Lo standard 802.11i esegue piccole modifiche al IEEE 802.1X per le reti wireless per far fronte alla possibilità di furto di identità. L'autenticazione del messaggio deve essere incorporato affinché sia il supplicant sia l'autenticatore calcolino le chiavi segrete e permettono la cifratura prima di accedere alla rete. Il supplicant e l'autenticatore comunicano usando un protocollo EAP. EAP è una framework per il trasporto di diversi metodi di autenticazione, accettando solo un numero limitato di messaggi (Request, Response, Success, Failure), mentre gli altri messaggi intermediari dipendono dal metodo di autenticazione scelto: EAP-TLS, EAP-TTLS, PEAP, Kerberos V5, EAP-SIM ecc. Quando l'intero processo è completato (a causa del gran numero di possibili metodi non ci addentreremo nei dettagli), entrambe le entità (cioè il supplicant e il server di autenticazione) hanno una chiave principale segreta. La comunicazione tra l'autenticatore e il server di autenticazione procede utilizzando il protocollo EAPOL (EAP Over LAN), usato nelle reti wireless per il trasporto di dati EAP con i protocolli di livello superiore come Radius.
Dopo la digressione possiamo comprendere più chiaramente la fase due riassunta nella figura 4.9.

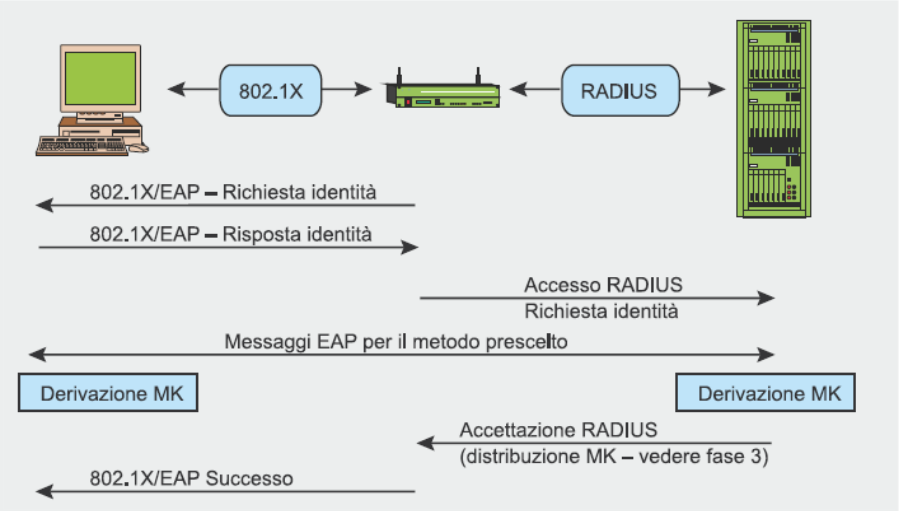

Figura 4.9: *Autenticazione 802.1X*

La seconda fase è l'autenticazione 802.1X basata sul EAP e il metodo di autenticazione concordato in precedenza: EAP/TLS con certificati client e server (che richiedono una infrastruttura per chiave pubblica), EAP/TTLS o PEAP per l'autenticazione ibrida (con certificato richiesti solo per i server)

ecc. L'autenticazione 802.1X viene iniziata quando il punto di accesso richiede i dati di identità del client, con la risposta del client contenente il metodo di autenticazione scelto. I messaggi adatti sono poi scambiati tra il client e il server di autenticazione per generare una chiave maestra comune (MK). Alla fine della procedura, un messaggio Radius Accept è inviato dal server di autenticazione al punto di accesso che contiene la MK e il messaggio EAP Success per il client.
Nelle reti in cui non é presente un server di autenticazione, un'alternativa all'autenticazione 802.1X é quella attraverso Pre-Shared Key (PSK). La PSK é unica e viene condivisa tra tutti i MT e l'AP. In pratica, molti AP utilizzano una singola PSK per tutti i MT. Ciò significa che, piuttosto che autenticare il MT, l'AP verifica che il MT sia un membro di un gruppo autorizzato (il gruppo che condivide la chiave). L'effettiva identità del MT non viene stabilita per cui questa non é un'autenticazione vera e propria. Tipicamente, la fase di autenticazione in un'RSNA, prevede la mutua autenticazione tra un MT ed un server di autenticazione, e la consegna di una comune master key (MK) di sessione all'AP. Tuttavia, in una RSNA che ha negoziato PSK durante la fase di accordo sulle politiche di sicurezza, la fase di autenticazione non é necessaria perché la chiave PSK é già stata distribuita ed installata in modo da rendere implicita la condizione di autenticazione.

Fase 3: derivazione e distribuzione delle chiavi
La sicurezza nelle connessioni dipende molto dalle chiavi segrete. Nel RSN, ogni chiave ha una vita limitata e la sicurezza generale è garantita da un'insieme di diverse chiavi, organizzate in una gerarchia. Quando viene stabilito un contesto di sicurezza dopo una autenticazione andata a buon fine, le chiavi temporanee (di sessione) vengono create e aggiornate regolarmente fino a quando il contesto di sicurezza non viene chiuso.

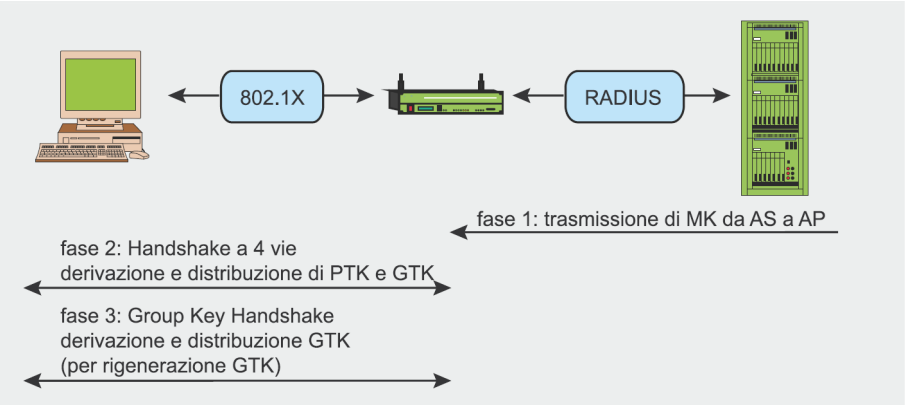

Figura 4.10: *Derivazione e distribuzione di chiave*

Come possiamo notare nella figura 4.10 abbiamo due handshake durante le derivazione delle chiavi:

- Handshake a quattro vie per la derivazione della PTK (*Pairwise Transient Key*) e della GTK (*Group Transient Key*)

- Group Key Handshake per il rinnovo della GTK

La derivazione della PMK (*Pairwise Master Key*) dipende dal metodo di autenticazione usato:

- se viene usata una PSK (PreShared Key), PMK = PSK. La PSK viene generata dalla passphrase (da 8 a 63 caratteri) o da una stringa di 256 bit e fornisce una soluzione per le reti domestiche e di piccole imprese che non hanno un server di autenticazione;

- se viene usato un server di autenticazione, la PMK è derivata dall'autenticazione 802.1X MK.

La PMK stessa non è mai usata per il controllo di cifratura o integrità. Inveceè usato per generare una chiave di cifratura temporanea PTK (Pairwise Transient Key) utilizzata per il traffico unicast. La funzione utilizzata per la derivazione della PTK in IEEE 802.11i é la seguente:

802.11i-PRF(*K*, *A*, *B*, *Len*)

$R \leftarrow$ ""

for $i \leftarrow 0$ to ((Len+159)/160) – 1) do

$R \leftarrow R \parallel$ HMAC-SHA1(K, $A \parallel B \parallel i$)

return Truncate-to-len(R, Len)

- K = PMK (Pairwise Master Key)
- A = una stringa fissa (Pairwise key expansion)
- B = min(AP-Addr, STA-Addr) || max(AP-Addr, STA-Addr) || min(ANonce, SNonce) || max(ANonce, SNonce)
- Len = lunghezza della chiave da generare (TKIP=512; CCMP=384)

La lunghezza della PTK dipende dal protocollo di cifratura: 512 bit per TKIP e 384 bit per CCMP. La PTK consiste di diverse chiavi temporanee:

- KCK (*Key Confirmation Key*, 128 bit): Chiave per i messaggi di autenticazione (MIC) durante la Handshake a 4 vie e la Group Key Handshake;
- KEK (*Key Encryption Key*, 128 bit): Chiave per garantire la segretezza durante la Handshake a 4 vie e la Group Key Handshake;
- TK (Temporary Key, 128 bit): Chiave per la cifratura dei dati (usata per TKIP o CMMP);
- TMK (*Temporary MIC Key*, 2x64 bit): Chiave per l'autenticazione dei dati (usata solo da Michael con TKIP). Una chiave dedicata è usata per ogni lato delle comunicazione;

Questa gerarchia è riassunta nella figura 4.11.

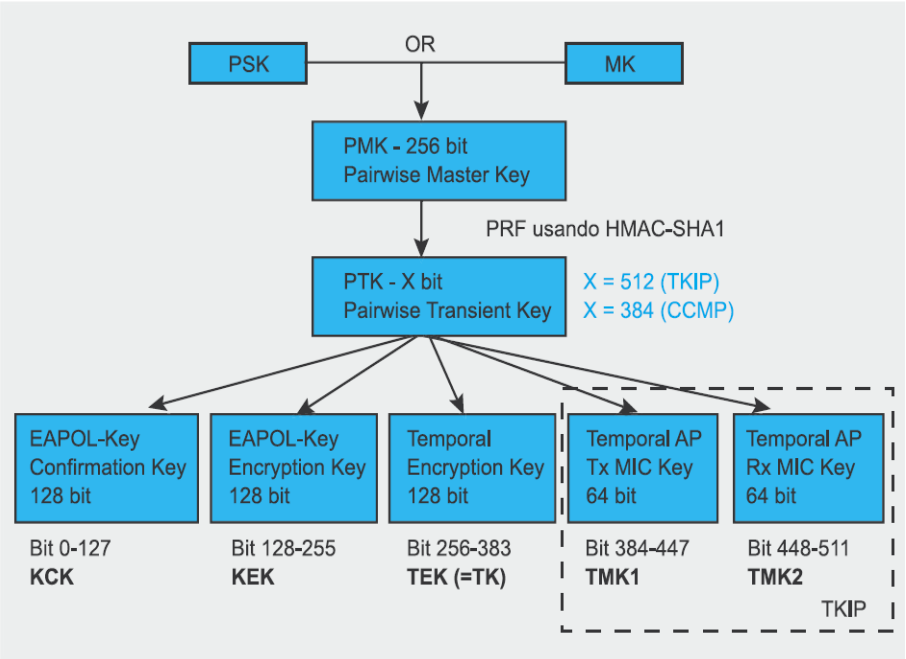

Figura 4.11: *Pairwise Key Hierarchy*

Tramite la Handshake a 4 vie, iniziata dal punto di accesso, è possibile:

- confermare la conoscenza del client della PMK;
- installare chiavi di cifratura e di integrità;
- cifrare il trasporto della GTK;
- confermare la selezione di cifratori;

Durante l'handshake a 4 vie vengono scambiati quattro messaggi EAPOL-Key tra il client ed il punto di accesso .

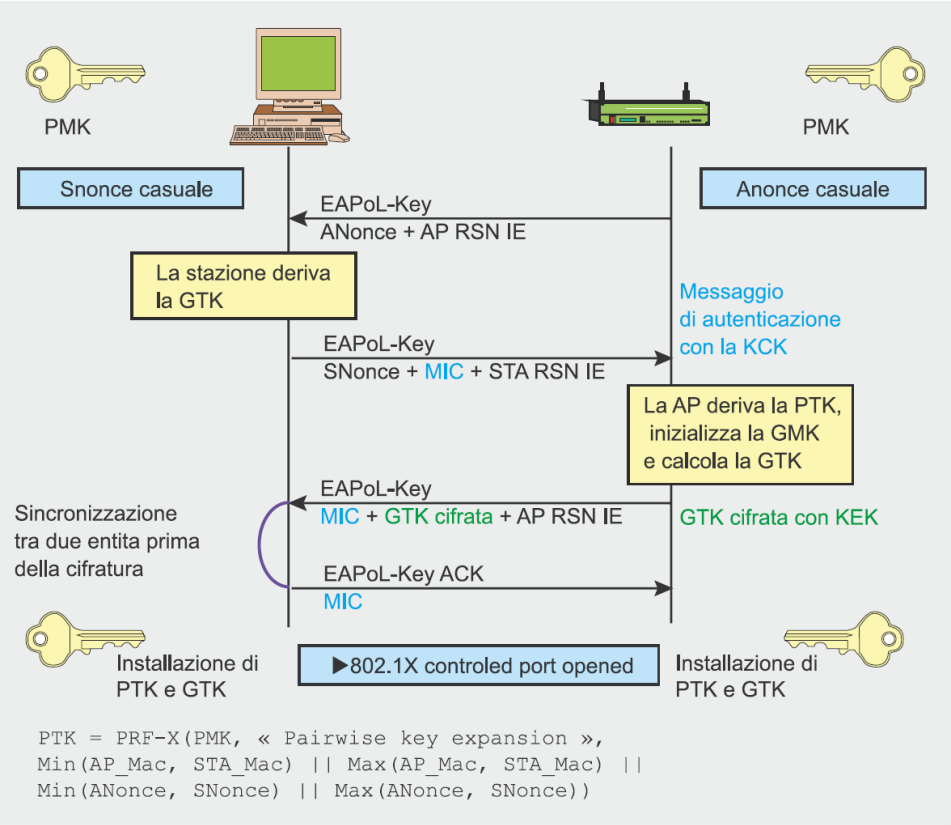

Figura 4.12: *Handshake a 4 vie*

La PTK viene derivata dalla PMK, una stringa fissa, l'indirizzo MAC del punto di accesso, l'indirizzo MAC del client e due numeri casuali (ANonce e SNonce, generati rispettivamente dall'autenticatore e il supplicant). Il punto di accesso inizializza il primo messaggio selezionando il numero casuale ANonce e lo invia al supplicant, senza cifrare il messaggio o proteggendolo in alternativa contro il miscuglio. Il supplicant genera il proprio numero causale SNonce e adesso può calcolare una PTK e le chiavi di derivazione temporanee, quindi invia SNonce e la chiave MIC calcolata dal secondo messaggio usando la chiave KCK. Quando l'autenticatore riceve il secondo messaggio, può estrarre SNonce (perché il messaggio non è cifrato) e calcolare la PTK e le chiavi di derivazione temporanee. Adesso può verificare il valore della MIC nel secondo messaggio e essere sicuro che il supplicant conosca la PMK e che ha calcolato correttamente la PTK e le chiavi di derivazione temporanee. Il terzo messaggio inviato dall'autenticatore al supplicant contiene la GTK (cifrata con la chiave KEK), derivata da una GMK casuale e GNonce (vedere figura 15 per dettagli), insieme ad una MIC calcolata dal terzo mes-

saggio usando la chiave KCK. Quando il supplicant riceve questo messaggio, la MIC viene controllata per essere sicuri che l'autenticatore conosca la PMK e ha calcolato correttamente la PTK e le chiavi di derivazione temporanee. L'ultimo messaggio riconosce il completamento dell'intera handshake e indica che il supplicant adesso installerà la chiave e iniziare la cifratura. Dopo la ricezione, l'autenticatore installa le proprie chiavi dopo aver verificato il valore MIC. Quindi, il dispositivo mobile e il punto di accesso hanno ottenuto, calcolato e installato le chiavi di cifratura e adesso sono in grado di comunicare su un canale sicuro per traffico unicast e multicast. Il traffico multicast è protetto da un'altra chiave, la GTK (*Group Transient Key*) generata da una chiave principale detta GMK (*Group Master Key*), una stringa fissa, l'indirzzo MAC del punto di accesso e un numero casuale GNonce. La lunghezza GTK dipende dal protocollo di cifratura , 256 bit per la TKIP e 128 bit per la CCMP. La GTK si divide in diverse chiavi temporanee:

- GEK (*Group Encryption Key*): Chiave per la cifratura dei dati (usata da CCMP per l'autenticazione e la cifratura e dalla TKIP);

- GIK (*Group Integrity Key*): Chiave per l'autenticazione dei dati (usata solo da Michael con TKIP);

Questa gerarchia è riassunta nella figura 4.13.

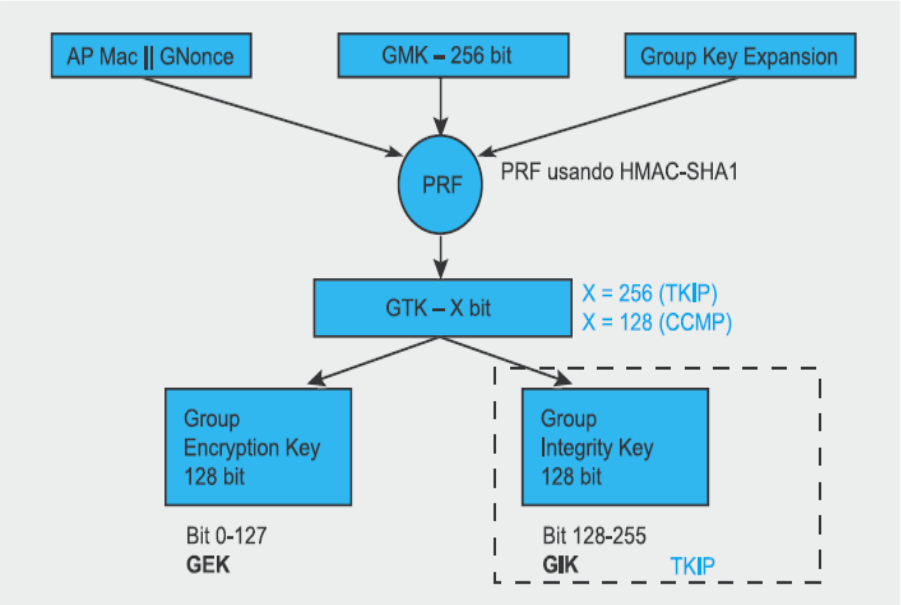

Figura 4.13: *Group Key Hierarchy*

Durante il Group Key Handshake vengono scambiati due messaggi EAPOL-Key tra il client e il punto di accesso. Questa handshake fa uso di chiavi temporanee generate durante la Handshake a 4 vie (KCK e KEK).Questo processo è illustrato nella figura 4.14.

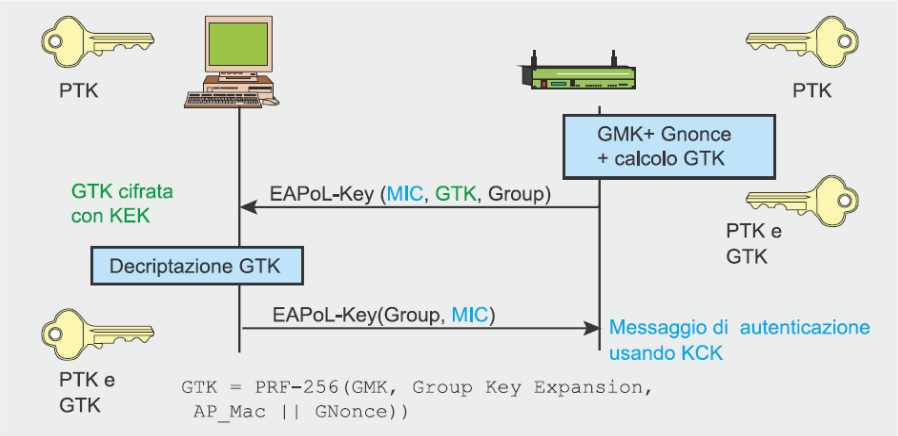

Figura 4.14: *Group Key Handshake*

La Group Key Handshake é richiesta solo per dissociare un host e per rigenerare la GTK su richiesta di un client. L'autenticatore inizializza il primo messaggio scegliendo il numero causale GNonce e calcolando una nuova GTK. Poi invia al supplicant la GTK cifrata (usando KEK), il numero di sequenza GTK e la MIC calcolata da questo messaggio usando KCK. Quando il supplicant riceve il messaggio, la MIC viene verificata e la GTK può essere decifrata. Il secondo messaggio riconosce il completamento della Group Key Handshake inviando un numero di sequenza GTK e la MIC calcolata su questo secondo messaggio. Dopo la ricezione, l'autenticatore installa la nuova GTK (dopo aver verificato il valore MIC).

Fase 4 :segretezza ed integrita dei dati

Tutte le chiavi generate in precedenza sono usate nei protocolli che supportano la segretezza e l'integrità dei dati RSNA:

- TKIP (*Temporal Key Hash*);
- CCMP (*Counter-Mode / Cipher Block Chaining Message Authentication Code Protocol*);

Prima di analizzare in dettaglio questi protocolli dobbiamo capire un concetto importante: la differenza che passa tra un MSDU (MAC Service Data

Unit) ed un MPDU (MAC Protocol Data Unit). Entrambi si riferiscono ad un singolo pacchetto di dati, ma l'MSDU rappresenta i dati prima della frammentazione, mentre i MPDU sono unità di dati multipli dopo la frammentazione. La differenza è importante nella cifratura TKIP e CCMP, dal momento che in TKIP la MIC viene calcolata dal MSDU, mentre nel CCMP viene calcolata dal MPDU. Proprio come WEP, la TKIP si basa sull'algoritmo di cifratura RC4, ma esiste per una sola ragione: per permettere ai sistemi WEP di essere aggiornati e implementare protocolli più sicuri. La TKIP è richiesta per la certificazione WPA ed è anche inserita nel RSN 802.11i come opzione facoltativa. La TKIP aggiunge anche misure correttive per tutte le vulnerabilità WEP:

- integrità del messaggio: una nuova MIC (M*essage Integrity Protocol*) chiamata Michael che può essere implementata nei software che girano su microprocessori lenti;

- IV: una nuova selezione di regole per i valori IV, re-utilizzando il vettore IV come contatore replay (TSC, o TKIP Sequence Counter) e aumentando le dimensioni del vettore IV per evitare il riutilizzo;

- Funzione *Per Packet Key Mixing*: per generare chiavi di cifratura apparentemente non legate;

- gestione chiave: nuovo meccanismo modifica delle chiavi;

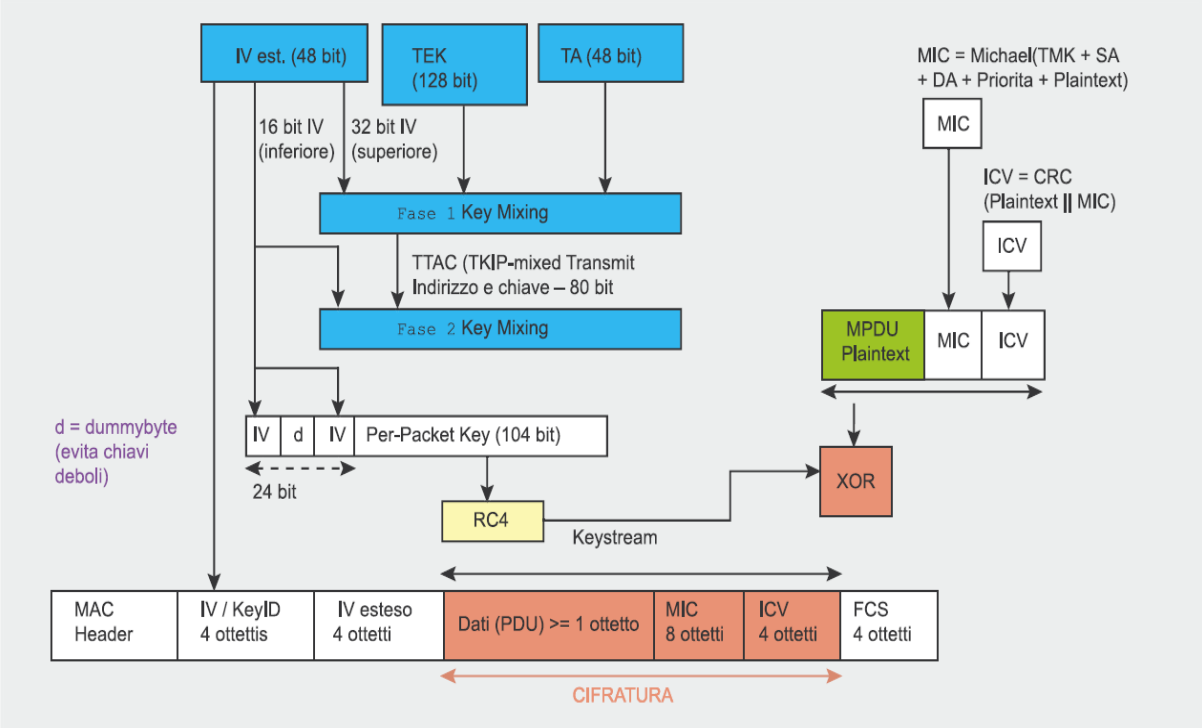

Figura 4.15: *Modello TKIP Key-Mixing e cifratura*

Il modello TKIP Key-Mixing si divide in due fasi:

- La Fase 1 riguarda i dati statici, la chiave di sessione segreta TEK, l'indirizzo trasmettitore MAC TA (per prevenire le collisioni tra vettori IV) e i 32 bit più alti del vettore IV. L'output di questa fase è una chiave intermedia che servirà da input per la fase 2.

- La Fase 2 dipende dall'output della Fase 1 e comprende i 16 bit più bassi del vettore IV, modificando tutti i bit del campo Per Packet Key per ogni nuovo IV. Il valore IV inizia sempre con 0 ed è aumentato di 1 per ogni pacchetto inviato, con il rifiuto di qualsiasi messaggio il cui TSC non è più grande dell'ultimo messaggio.

L'output della Fase 2 e parte del IV esteso (più un dummy byte per evitare chiavi deboli) sono l'input per l'RC4, e generano un keystream con un operatore XOR con un MPDU in testo in chiaro, la MIC calcolata dalla MPDU ed il vecchio ICV di WEP. La figura 4.16 mostra come avviene la fase di codifica e decodifica utilizzando TKIP.

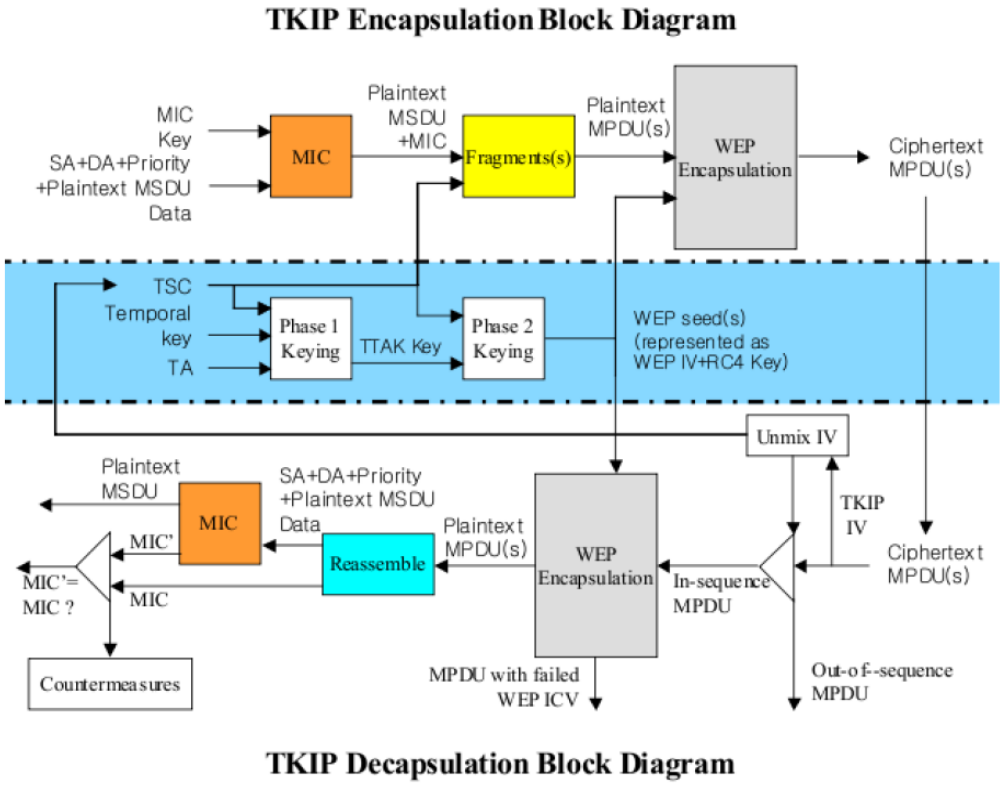

Figura 4.16: *TKIP block diagram*

Il MIC utilizza l'algoritmo Michael di Niels Ferguson. Vienne creato per la TKIP ed ha un livello di sicurezza di 20 bit (l'algoritmo non usa la moltiplicazione per ragioni di prestazioni, poiché deve essere supportato da hardware wireless di vecchia generazione che deve essere aggiornato per il WPA). A causa di questi limiti, delle contromisure sono necessaire per evitare alterazioni MIC. I guasti MIC devono essere ridotti a due al minuto, altrimenti viene applicato un blackout di 60 secondi e le nuove chiavi (GTK e PTK) devono essere ristabilite in un secondo momento. Micheal calcola un valore di controllo di 8 ottetti chiamato MIC e lo aggiunge al MSDU prima della trasmissione.

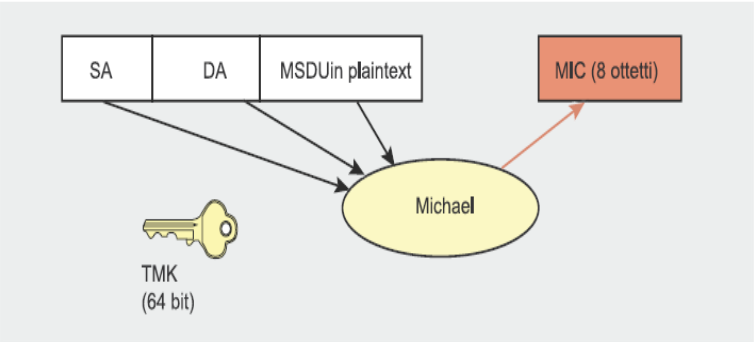

Figura 4.17: *Calcolo MIC con l'algoritmo Michael*

Il MIC viene calcolato dall'indirizzo di origine (SA), indirizzo di destinazione (DA), MSDU in testo in chiaro e la TMK appropriata (a seconda dei casi, viene usata una chiave diversa per la trasmissione e la ricezione.
Il CCMP si basa sulla suite del cifrario a blocchi AES (Advanced Encryption Standard) in modalità CCM con le chiavi e i blocchi di 128 bit. AES è per CCMP quello che RC4 è per TKIP, ma al contrario di TKIP che era stato creato per accogliere l'hardware WEP esistente, CCMP non è un ibrido bensì un nuovo protocollo. Il CCMP utilizza Counter Mode (CTR) per la confidenzialità combinato con un metodo di autenticazione di messaggio chiamato Cipher Block Chaining (CBC-MAC) che produce un MIC. Esso aggiunge altre funzionalità interessanti, come l'uso di una singola chiave per la cifratura e l'autenticazione (con diversi vettori di inizializzazione) e l'autenticazione dei dati non cifrati.

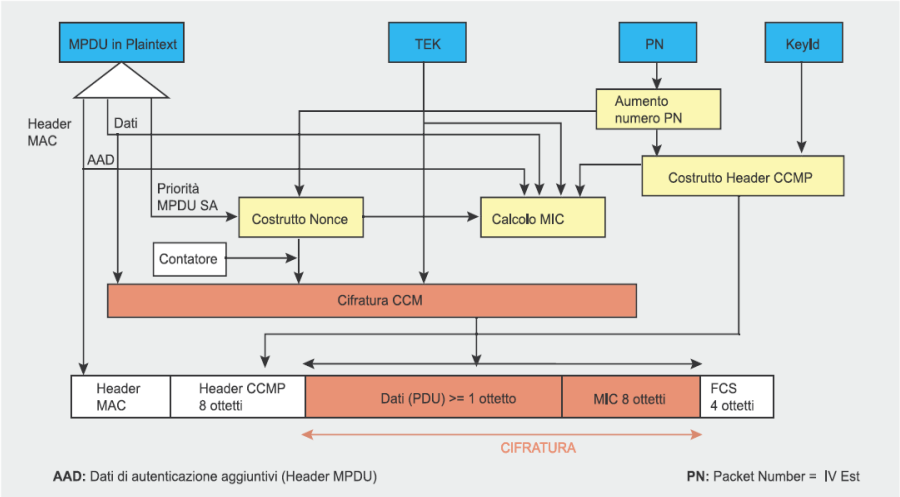

Figura 4.18: *Cifratura CCMP*

Il protocollo CCMP aggiunge 16 byte alla MPDU: 8 byte per l'intestazione CCMP e 8 byte per il MIC. L'intestazione CCMP è un campo non cifrato incluso tra l'intestazione MAC ed i dati cifrati, ed include il PN di 48 bit (Packet Number = IV esteso) ed il KeyID (un byte che contiene: Ext IV (bit 5), Key ID (bits 6-7), ed un campo riservato (bits 0-4). Il PN viene aumentato di uno per ogni MPDU successivo. Il calcolo MIC utilizza l'algoritmo CBC-MAC che cifra il blocco nonce di partenza (calcolato dai campi Priority, l'indirizzo di origine dell'MPDU ed il PN aumentato) ed esegue lo XOR con i blocchi successivi per ottenere una MIC finale di 64 bit (la MIC finale è un blocco di 128 bit però i 64 bit più bassi vengono ignorati). Il MIC viene poi aggiunto ai dati in testo in chiaro per la cifratura AES in modalità contatore. Il contatore è costruito da un nonce simile a quello del MIC, ma con un campo contatore in più inizializzato ad 1 ed incrementato per ogni blocco.

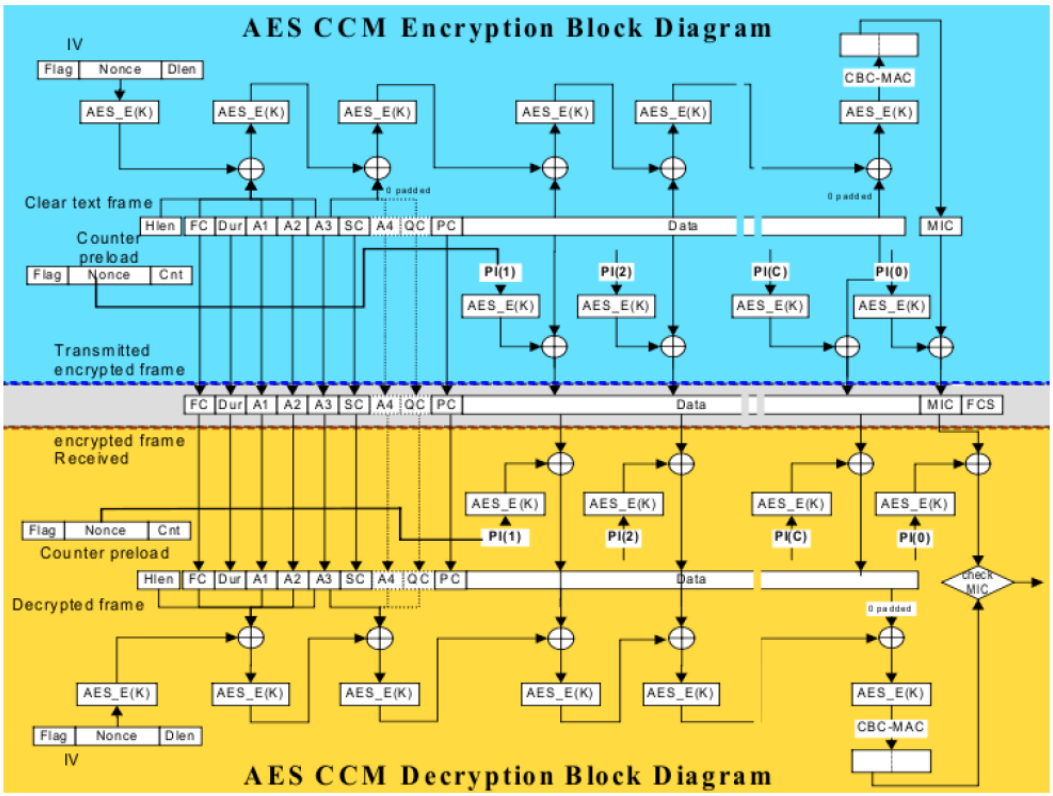

Figura 4.19: *AES CCMP block diagram*

L'ultimo protocollo è WRAP, basato anch'esso su AES, ma che utilizza il modello di cifratura autenticato OCB (Offset Codebook Mode, cifratura e autenticazione in un unico calcolo). Il modello OCB fu il primo scelto dal gruppo di lavoro di IEEE 802.11i, ma venne poi abbandonato a causa di

problemi legati alla proprietà intellettuale e le tariffe della licenza. Il CCMP venne allora adottato come metodo obbligatorio.

4.2.1 WPA

Prima che lo standard 802.11i venisse completamente definito, il gruppo Wi-Fi Alliance per contrastare il prima possibile le lacune di sicurezza del WEP fornì un protocollo di sicurezza chiamato *Wireless Protected Access* (WPA) basato su un sott'insieme di funzionalità presenti nello standard 802.11i. Questa evoluzione adotta le seguenti funzionalità di IEEE 802.11i:

- autenticazione mediante IEEE 802.1X;
- Generazione e distribuzione delle chiavi basati sul 4-Way Handshake;
- Cifratura TKIP
  - Raddoppio della lunghezza del vettore di inizializzazione IV (da 24 bit a 48);
  - IV utilizzato come un contatore (TSC, *TKIP Sequence Counter*) per previene la ripetizione del keystream;
  - Chiave calcolata ad ogni pacchetto (per packet) usando il vettore IV;
  - PMK (*Pairwise Master Key*) non viene MAI usata direttamente ma viene ricavata sempre una nuova chiave PTK (*Pairwise Transient Key*) usando la PMK;
  - Utilizzo di **Michael** come MIC (*Message Integrity Check*) per garantire l'integrità del messaggio.

4.2.2 WPA2

Completato e definito la standard 802.11i nel Settembre del 2004 viene presentato WPA2. WPA2 (a differenza di WPA) implementa tutte le funzionalità definite dallo standard IEEE 802.11i. Così come il suo predecessore, WPA2 utilizza 802.1x ed EAP per l'autenticazione (oltre all'alternativa PSK). La differenza sostanziale tra WPA e WPA2 risiede nell'algoritmo crittografico.

WPA2 utilizza CCMP con AES (*Advanced Encryption Standard*) mentre WPA uilizza TKIP con RC4. AES è un block cipher e richiede capacità computazionali più elevate a livello hardware. Questo implica che non tutti i dispositivi che supportano WPA possono essere aggiornati via software per essere compatibili con WPA2, mentre invece è garantita l'interoperabilità di WPA2 con WPA. Inoltre WPA2 aggiunge un'altra funzionalità interessante: l'uso di una singola chiave per la cifratura e l'autenticazione (con diversi vettori di inizializzazione).
Entrambi WPA e WPA2 hanno due modalità operative: *Personal* ed *Enterprise*. La prima implica l'utilizzo di una chiave pre-condivisa per l'autenticazione, mentre la seconda utilizza lo standard IEEE 802.1X ed EAP.

Capitolo 5

Attacchi e vulnerabilità

Prima di analizzare gli attacchi una certa attenzione merità la reale fattibilità di essi. Una delle più grandi barriere alla realizzazione pratica di questi attacchi è l'accesso ai dati del sistema di comunicazioni. Nonostante questi siano trasmessi mediante onde radio occorre un certo equipaggiamento per monitorare ed iniettare traffico in una comunicazione. Le cose si complicano in particolar modo nel caso dei collegamenti in ponte radio in cui vengono utilizzate antenne fortemente direttive posizionate in ambienti poco favorevoli. In questa sede suporremmo che l'attaccante si strovi nelle condizioni ideali e che possega le risorse necessarie per acedere al canale.
Inoltre, un'altra nota di attenzione merita il fatto che gli attacchi più significativi in letterattura sono stati e vengono sviluppati basandosi su un architettura come quella rappresentata in figura 5.1.

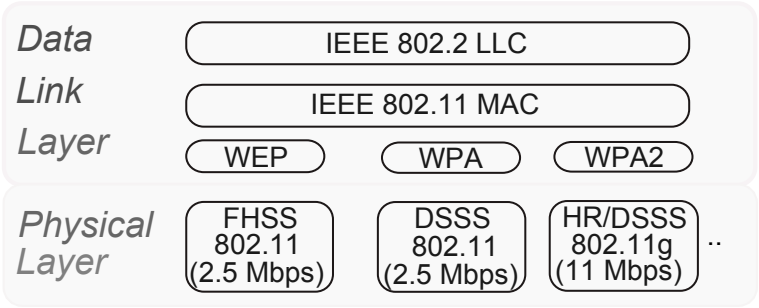

Figura 5.1: *Architettura 802.11 WLAN*

Tale architettura sarà utile per la comprensione di una parte degli attachi. In figura 5.2 è rappresentata la cronologia temporale per quanto riguarda l'introduzione degli standard di sicurezza per la protezione delle 802.11 WLAN,

la scoperta delle vulnerabilità e lo sviluppo degli attacchi. Lo scopo di questo capitolo è verificare l'applicabilità di essi però su un architettura in cui il livello DLL e PHY è occupato da Hiperlan 2.

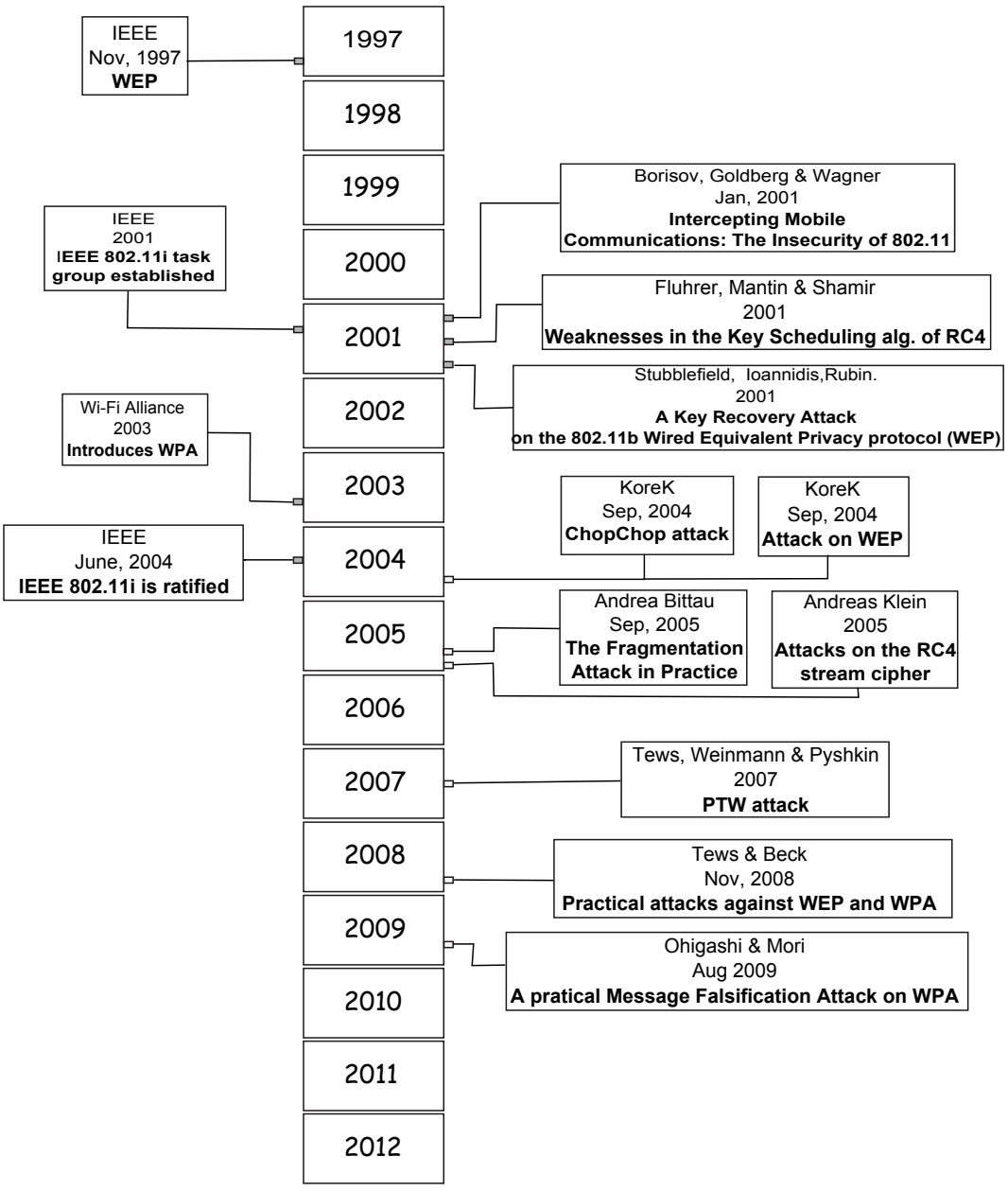

Figura 5.2: *Cronologia temporale dello svilluppo degli standard di sicurezza delle reti Wi-Fi confrontato con lo sviluppo degli attacchi e le vulnerabilità scoperte*

5.1 Debolezze del WEP

In questa sezione analizzeremo quelle che sono le vulnerabilità di WEP. Si potrà osservare di seguito che alcune prescindono dalla architettura in cui è collocato il protocollo WEP. L'unico requisito richiesto per sfruttarle è la capacità di codifica/decodifica di pacchetti WEP o pacchetti appartenenti a livelli superiori al DLL. Altre debolezze invece sono fortemente legate a livello DLL al protocollo LLC quindi gli attacchi nascono ad hoc sulle caratteristiche di questo specifico protocollo.

5.1.1 Vulnerabilità non legate all'algoritmo $RC4$

Ad iniziare l'opera di smantellamento, dal punto di vista della sicurezza, furono i ricercatori Nikita Borisov, Ian Goldberg e David Wagner che nel 2001 dimostrarono una serie di vulnerabilità intrinseche al protocollo WEP [22].

Rischio di riutilizzo dello stesso *keystream*

WEP alla base del proprio sistema di sicurezza prevede l'utilizzo dell'algoritmo $RC4$, il quale appartiene alla categoria degli *stream chipher*. Ogni algoritmo di questo tipo agisce espandendo una chiave segreta, nel caso del WEP si tratta di un IV pubblico concatenato ad una chiave segreta, in uno stream arbitrariamente lungo di bit pseudo-casuali. La codifica avviene poi calcolando lo XOR tra lo stream generato ed il testo in chiaro. La decodifica avviene seguendo le stesse operazioni: si genera uno stream identico al precedente ed eseguendone lo XOR con il testo codificato, si otterrà il testo in chiaro. Un noto punto debole degli algoritmi di stream chypher consiste nel fatto che il codificare due messaggi con una stessa chiave (nel caso del WEP, con uno stesso IV e con la stessa chiave), può rivelare informazioni riguardo ad entrambi i messaggi. Infatti, se definiamo i due messaggi codificati $C1$ e $C2$ come:

$$C1 = P1 \oplus RC4(v, k)$$
$$C2 = P2 \oplus RC4(v, k)$$

si ottiene:

$$C1 \oplus C2 = (P1 \oplus RC4(v, k)) \oplus (P2 \oplus RC4(v, k)) = P1 \oplus P2$$

In altre parole, calcolando lo XOR tra i due messaggi cifrati si riesce ad eliminare l'effetto del keystream, ottenendo come risultato lo XOR tra i due

messaggi in chiaro. Questa proprietà può rendere possibile diversi tipi di attacchi: se infatti il testo in chiaro di uno dei due messaggi fosse conosciuto, allora si otterrebbe immediatamente anche il secondo:

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

Più in generale, nelle situazioni reali, spesso i messaggi in chiaro hanno abbastanza ridondanza tra loro da permettere di risalire ad entrambi semplicemente analizzandone lo XOR che si ha a disposizione. Queste tecniche diventano notevolmente più semplici da applicare avendo a disposizione un elevato numero di pacchetti codificati con la stessa chiave.
Per evitare questi attacchi, WEP utilizza un IV che varia in ogni pacchetto trasmesso. In questo modo si cambia il processo di generazione del *keystream* in ogni messaggio. L'IV è inviato in chiaro negli header del pacchetto in modo che il ricevente possa generare lo stesso *keystream* necessario per la decodifica. In questo modo però si espone l'IV ad eventuali attaccanti, anche se la parte rimanente della chiave segreta rimane sconosciuta. La variazione dell'IV in ogni pacchetto trasmesso è stata introdotta per evitare il riutilizzo della stessa chiave, ma purtroppo il protocollo WEP non riesce a raggiungere questo scopo. Occorre infatti notare che lo standard WEP si limita a raccomandare, ma non ad obbligare, che lo IV sia cambiato in ogni pacchetto ed inoltre non indica alcun modo in cui tale variazione debba avvenire. A causa di questa mancanza molti produttori hanno realizzato schede che inizializzano lo IV a 0 per il primo pacchetto e lo incrementano di uno in ogni pacchetto successivo.
Un altro aspetto da considerare riguarda la lunghezza dell'IV, essa è di soli 24 bit, alcuni studi hanno mostrato come un normale Access Point sottoposto ad elevato traffico esaurisca tutti gli IV disponibili in meno di mezza giornata. Altri costruttori, utilizzano invece IV scelti in modo casuale, ma in questo caso la situazione può addirittura peggiorare: a causa del cosiddetto *paradosso del compleanno* ci si può aspettare un riutilizzo dello stesso IV dopo la trasmissione di soli 5000 messaggi, cioè dopo solo pochi minuti.

Modifica dei messaggi

Il protocollo WEP prevede che all'interno di ogni pacchetto sia inserito un campo contenente il checksum calcolato con CRC32, questo campo sarà criptato insieme al messaggio da trasmettere. Il checksum CRC non è però sufficiente ad assicurare che un attaccante non possa in alcun modo modificare un messaggio inviato. CRC è stato infatti progettato per identificare errori casuali durante la trasmissione del messaggio, ma non è in grado di resistere ad attacchi effettuati da utenti malevoli. Questo tipo di vulnerabilità è

addirittura ampliata dal fatto che il corpo del messaggio sia codificato utilizzando un algoritmo di *stream cypher*. Per dimostrare questa debolezza bisogna considerare che il checksum è una funzione lineare del messaggio. Questo significa che se c è la funzione di checksum, si ha che:

$$c(x \oplus y) = c(x) \oplus c(y).$$

Questa è una proprietà generale di tutte le funzioni CRC. Per mostrare come possa essere sfruttato da un attaccante, consideriamo un messaggio codificato C che viene intercettato e chiamiamo M il corrispondende messaggio in chiaro, si ha che:

$$C = RC4(v; k) \oplus \langle M; c(M) \rangle$$

Lattaccante può modificare il messaggio M in un nuovo messaggio M', semplicemente facendo lo XOR tra il messaggio cifrato intercettato C ed un nuovo messaggio Δ tale che $M' = M \oplus \Delta$. Si consideri infatti la seguente equazione:

$$\begin{aligned} C' = C \oplus \langle \Delta, c(\oplus) \rangle &= RC4(v; k) \oplus \langle M, c(M) \rangle \oplus \langle \Delta, c(\Delta) \rangle \\ &= RC4(v; k) \oplus \langle M \oplus \Delta, c(M) \oplus c(\Delta) \rangle \\ &= RC4(v; k) \oplus \langle M', c(M \oplus \Delta) \rangle \\ &= RC4(v; k) \oplus \langle M', c(M') \rangle \end{aligned}$$

Essa mostra la validità del messaggio modificato. Infatti il destinatario decodificherà il messaggio e troverà un checksum corretto per M' e considererà quindi valido quanto avrà ricevuto. Si noti che questo attacco può essere applicato anche senza conoscere nessuna parte del messaggio in chiaro M. L'attaccante deve solamente intercettare il messaggio cifrato C e decidere la differenza con l'originale, cioè Δ. Questo implica che si possano effettuare modifiche arbitrarie su un pacchetto senza che nessuno riesca ad identificare il cambiamento. La protezione dell'integrità dei dati è uno degli obiettivi che si era prefisso il protocollo WEP, ma il checksum adottato non permette assolutamente di raggiungerlo.

Mesagge Injection

L'utilizzo di checksum CRC all'interno del protocollo WEP comporta, oltre alla potenziale modifica di un messaggio trasmesso, anche la possibilità di inviare messaggi in una rete senza essere autorizzati ad accedervi. Quindi il WEP non provede ad uno dei suoi fondamentali obbiettivi cioè un controllo

degli accessi sicuro. La funzione di checksum non è infatti una funzione crittografica. Conseguentemente ogni attaccante che conosca il messaggio originale può calcolarne il checksum in modo autonomo. Quindi se riuscisse ad ottenere un messaggio cifrato ed il corrispondente messaggio in chiaro potrebbe, come si è visto nel paragrafo precedente , ottenerne il relativo *keystream*. Il quale potrebbe cosi servire per creare nuovi pacchetti semplicemente riutilizzando lo stesso IV. Questa possibilità è dimostrata dalla seguente equazione in cui P è il messaggio in chiaro e C è il corrispondente cifrato:

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

Avendo ottenuto il valore del *keystream* RC4(v, k), lo si può utilizzare per creare qualsiasi altro messaggio cifrato:

$$C' = \langle M', c(M') \rangle \oplus RC4(v, k)$$

E' necessario utilizzare sempre lo stesso IV in ogni pacchetto, ma dal momento che il protocollo WEP non indica alcun modo su come esso debba variare, lo si può usare senza che questo crei nessun tipo di allarme da parte del destinatario. Una volta ottenuto un *keystream* lo si potrà utilizzare in modo indefinito ed in questo modo si riuscirà ad aggirare l'intero meccanismo di *Access Control* previsto dal WEP.

Falsificare l'autenticazione

Un caso particolare dell'invio non autorizzato dei messaggi può essere utilizzato per aggirare il meccanismo di autenticazione previsto dal WEP. Questo meccanismo è utilizzato dagli Access Point per autorizzare gli host mobili prima che formino un'associazione con esso. Infatti il procedimento previsto dal WEP indica che un Access Point, dopo aver ricevuto una richiesta di autenticazione da parte di un host mobile, gli invii un messaggio in chiaro di 128 byte casuali chiamato *challenge*. L'host mobile risponderà con lo stesso challenge dopo averlo codificato usando WEP. L'autenticazione avrà successo se la decodifica corrisponderà esattamente al messaggio inviato inizialmente dallAccess Point. La capacità di poter generare una versione codificata correttamente è considerata una dimostrazione di possesso della chiave WEP. Come descritto però nel precedente paragrafo è possibile inviare in una rete wireless dei pacchetti codificati correttamente senza dover conoscere la chiave WEP. Infatti è necessaria solamente la conoscenza di un messaggio in chiaro e del corrispondente codificato. Tale coppia di messaggi può essere facilmente ottenuta monitorando una legittima sequenza di autenticazione. In questo modo diventa estremamente semplice poter ricavare il *keystream*

utilizzato. Inoltre, dato che i messaggi di autenticazione sono tutti della stessa lunghezza, il *keystream* ottenuto potrà essere usato per codificare un qualsiasi *challenge* ricevuto.

Decodifica mediante redirezione dei pacchetti

La possibilità di modificare i pacchetti inviati in una rete wireless può essere sfruttata da un attaccante per decodificarli. L'idea si basa sul fatto che anche se l'attaccante non conosca la chiave WEP, in una rete ci sarà sempre almeno l'Access Point che può decodificare i pacchetti. Si cercherà quindi di fare in modo che, modificando opportunamente alcuni pacchetti, l'Access Point li decodifichi e li trasmetta ad un host sotto il controllo dell'attaccante. L'attacco consiste in un *reindirizzamento IP* ed è possibile in tutte le reti wireless in cui l'Access Point sia anche un router con accesso ad Internet. Questa situazione è sicuramente molto comune. L'attaccante utilizzerà le tecniche descritte per cambiare l'indirizzo destinazione di un pacchetto e costringere così l'Access Point a reindirizzarlo verso un host sotto controllo. Per effettuare una modifica efficace è necessario conoscere il reale indirizzo di destinazione del pacchetto, ma questa operazione non costituisce una difficoltà. Infatti la maggior parte dei pacchetti sarà diretta verso la rete wireless stessa, i cui indirizzi sono facilmente identificabili. L'unica difficoltà consiste nel mantenere un valore di checksum del pacchetto valido, anche dopo la modifica dell'indirizzo destinazione. In questo caso si possono utilizzare diversi accorgimenti per poter ottenere checksum validi. Il più semplice di essi consiste nel mantenerne il valore, cercando ad esempio di modificare anche l'indirizzo sorgente del pacchetto.

Reaction attack

Un modo alternativo per manipolare l'Access Point, al fine di rompere la protezione al traffico fornita del WEP, si può ottenere attravero l'analisi dei pacchetti TCP/IP protetti con WEP. L'attacco non richiede connettività a Internet e si può utilizzare quando l'attacco mediante *reindirizzamento IP* non è applicabile. L'attacco si basa sul fatto che i pacchetti TCP sono accettati dall'Access Point solamente se il TCP checksum è corretto. In questo caso un pacchetto di acknowledgement (ACK) viene spedito come risposta al client. Il pacchetto ACK è facilmente riconoscibile dalla sua dimensione senza la necessità di decrittazione. Perciò la reazione alla ricezione del pacchetto inviato ci rivela quando il suo checksum è valido. L'attacco ha inizio con l'intercettazione di un ciphertext $\langle v, C \rangle$ con un plaintext sconosciuto:

$$A \mapsto (B) : \langle v, C \rangle$$

Successivamente si flippano alcuni bit in C e si aggiusta il CRC crittato in moddo da ottenere un nuovo ciphertext C' con un valido WEP checksum. Dopo di che bisogna trasmettere C' in un nuovo pacchetto all'Access Point :

$$(B) \mapsto A : \langle v, C' \rangle$$

Infine, si vede se il destinatario risponde con un pacchetto TCP ACK. Questo permette di capire se il controllo del TCP checksum viene superato con successo e se il pacchetto viene accettato dal destinatario. Si noti che potremmo scegliere quali bit di C flippare nella maniera che preferiamo usando le tecniche precedentemente esposte. Il punto chiave della tecnica è il seguente: attraverso una scelta intelligente della posizione del bit da flippare possiamo assicurare che il TCP checksum rimanga indisturbato esattamente quando la condizione di un bit soddisfa $P_i \oplus P_{i+16} = 1$. Perciò la presenza o l'assenza del ACK rivelerà un bit di informazione sul plaintext P sconosciuto. Ripetendo l'attacco per molte scelte di i possiamo scoprire quasi tutto del plaintext P, e poi dedurre i pochi bit sconosciuti rimasti sarà semplice uttilizzando le tecniche classiche.
Spiegheremo in seguito precisamente quali bit scegliere da flippare.
Il punto principale, per ora, è che abbiamo sfruttato la disponibilità dell'Access Point per decriptare arbitrariamente testi criptati per poi passarli ad un altro componente del sistema che fa emergere un esiguo numero di bit di informazione relativo all'input. Quindi abbiamo usato l'Access Point come un *oracolo* che senza saperlo decripta per noi i testi cifrati intercettati. Questo è conosciuto col nome di *reaction attack* in quanto lavora monitorando la reazione ai pacchetti da noi manomessi.
Passiamo ora a spiegare i dettagli tecnici su come scegliere i pacchetti manomessi C' per indurre il ricevente a rivelare informazioni sul plaintext P sconosciuto. Sia $C' = C + \Delta$ dove Δ specifica la posizione del bit da fippare. Possiamo scegliere Δ come segue: preso i arbitrariamente settiamo la posizione dei bit i ed $i + 16$ di Δ a 1 ponendo il resto dei bit a 0. Una proprietà della somma modulo $(2^{16} - 1)$ stabilisce che:

$$P \oplus \Delta \equiv P mod(2^{16} - 1)$$

se si verifica la condizione $P_i \oplus P_{i+16} = 1$. Sicome assumiamo che il TCP checksum è valido per il pacchetto originale $(P \equiv 0 mod(2^{16} - 1))$,questo significa che il TCP checksum del nuovo pacchetto sarà valido $(P \oplus \Delta \equiv 0 mod(2^{16} - 1))$non appena $P \oplus \Delta \equiv P mod(2^{16} - 1)$. Questo ci dà un bit di informazione sul plaintext.

Attacco ChopChop

A settembre del 2004 un hacker sotto lo pseudonimo di KoreK rilascio presso un forum su internet due attacchi [25, 26] denominati poi *KoreK attack* e *ChopChop attack*. L'attacco *ChopChop* appartiene alla categoria degli attacchi non crittografici. Questo attacco non rivela la chiave k e non è basato su alcuna proprietà speciale di *RC4*. Come illustrato in figura 5.3, esso lavora troncando (da qui il nome *ChopChop*) l'ultimo byte di un pacchetto criptato. L'obiettivo che si prefigge è quello di individuare il valore di questo byte. Un modo per realizzare ciò consiste nell'iniettare il pacchetto troncato (criptato) nella rete. Tale pacchetto risulta ora non valido dal momento che l'ICV non corrisponde al resto del pacchetto. Tuttavia eseguendo l'operazione di XOR con una certa quantità *Mod* il pacchetto tornerebbe di nuovo valido. Tale quantità *Mod* dipende esclusivamente dal byte troncato. Provando tutti i valori tra 0 e 255 (in media 128) di questo byte alla fine troveremmo il valore cercato. A questo punto l'attaccante potrebbe usare l'access point come *oracolo* per capire quando il pacchetto trasmesso è corretto. Ora l'attaccante conoscerà sia il *plaintext* del byte troncato sia la *keystream*. Ripetendo il procedimento è possibile decriptare l'intero pacchetto cosi come la *keystream* senza la neccessità della chiave segreta k.

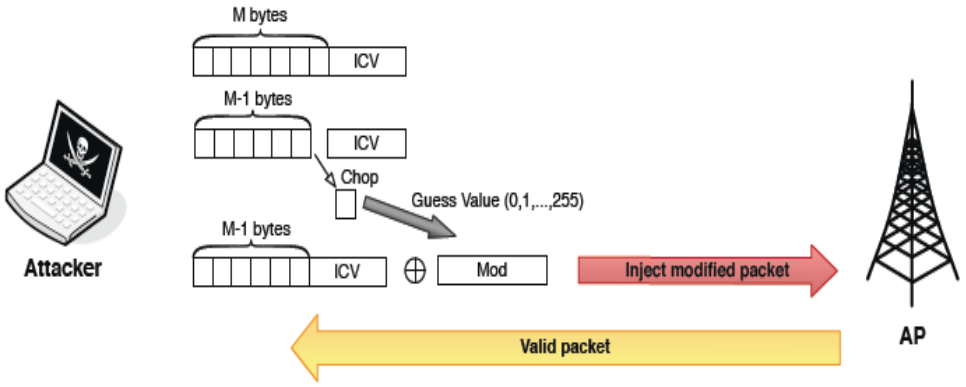

Figura 5.3: *ChopChop attack*

Diamo ora una breve descrizione della matematica che ci sta dietro l'attacco *ChopChop*. Sia P il *plaintext* troncato. Affinche il checksum sia corretto la seguente equazione deve essere verificata:

$$PmodR_{CRC} = P_{ONE}$$

dove R_{CRC} è il polinomio di $CRC32$ e P_{ONE} è un polinomio i cui coefficienti da X^0 a X^{31} sono uguali 1. Possiamo scrivere P come

$$P = QX^8 + P_7$$

con P_7 tutti gli elemnti di P con esponenti minori di 8. Adesso, allo scopo di avere un checksum corretto per $P = QX^8 + P_7$ occore che

$$Q \times X^8 = P_{ONE} + P_7 mod R_{CRC}$$

sia verificata. Invertendo X^8 abbiamo:

$$(X^8)^{-1} = R_{INV}$$

Ora sapiamo che

$$Q = R_{INV}(P_{ONE} + P7) mod R_{CRC}$$

Per avere un checksum valido, Q deve avere il valore

$$Q = P_{ONE} mod R_{CRC}$$

Adesso, aggiungendo $P_{COR} = P_{ONE} + P_{INV}(P_{ONE} + P_7)$ a Q abbiamo un nuovo messaggio per P che avrà un checksum corretto. Come si può notare P_{COR} dipende esclusivamente da P_7 che può assumere al massimo 256 valori possibili. L'attaccante può iniziare supponendo un valore di P_7, ridurre il messaggio originale di un byte, aggiungere il byte di correzione ed interogare l'*oracolo* se il valore supposto è corretto. In media occorrono 128 interogazioni per byte. Quindi per decriptare m byte di un *ciphertext* ocorrono in media m*128 interogazioni all'*oracolo*.

Attacco a Frammentazione

A settembre del 2005 Andera Bittau descrisse in un articolo un attacco molto originale noto come il *Fragmentation attck* [28]. L'originalità di questo attacco risiede nel fatto che catturando semplicemente un singolo pacchetto dalla rete si è in grado di iniettare nella stessa un numero arbitrario di pacchetti. La proprietà che sta alla base del attacco è la simmetria dell'operazione XOR. Come mostrato in precedenza eseguendo lo XOR tra *plaintext* e il corrispondente *ciphertext* si conosce il relativo *keystream*. Quindi si possono trasmettere pacchetti con lo stesso IV del pacchetto catturato. Lo stesso

keystream può essere usato, in futturo, per decriptare i pacchetti che riuserano lo stesso IV. Quini il problema principale è venire a conoscenza di una coppia (*plaintext*,*ciphertext*).
Tutti i pacchetti in una WLAN 802.11 hanno gli header simili. Ogni pacchetto, secondo l'architettura 802.11, vienne incapsulato dal LLC (*Logical Link Control*). L'header LLC è seguito sempre dall'intestazione SNAP (*Sub-Network Access Protocol*). L'intestazione LLC/SNAP è rappresentata nella figura seguente.

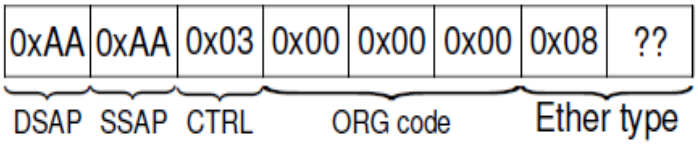

Figura 5.4: *Intestazione LLC/SNAP dei pacchetti IEEE 802.11*

Come si può notare essa è composta da 8 byte è contiene costanti in tutti i campi tranne per l'*Ethernet Type* che definisce il protocollo che segue nello stack. I candidati ad occupare quest'ultimo campo sono IP o ARP. Il valore per entrambi i protocolli comincia con $0x08$. I pacchetti ARP sono facilmente individuabili grazie alla loro lunghezza fissa di 36 byte. Qundi essendo in grado di distinguerli si è in grado di conoscere 8 byte di p*laintext* ed il relativo *ciphertext* e di conseguenza il relativo *keystream* o PRGA (*Pseudo Random.Generation Algorithm*). Con 8 byte di PRGA si è in grado di spedire alla rete pacchetti dati di 4 byte dal momento che 4 byte occorono per l'ICV. Trasmettere pacchetti di questa dimensione è assolutamente inutile dal momento che il livello che segue li scarta immediatamente. E' a questo punto che l'attacco a frammentazione entra in gioco. L'802.11 supporta la frammentazione dei pacchetti a livello MAC. Ciò significa che un pacchetto può essere suddiviso in framenti più piccoli, fino ad un massimo di 16, a acui WEP viene applicato in modo indipendentemente.Niente vieta a ciascun pacchetto di avere lo stesso IV Questo significa che spedendo 16 pachetti di 8 byte ciascuno (4 byte dati + 4 byte ICV) è possibile iniettare in rete un pacchetto di 64 byte. Questa parte dell'attacco è riassunta nella figura 5.5.

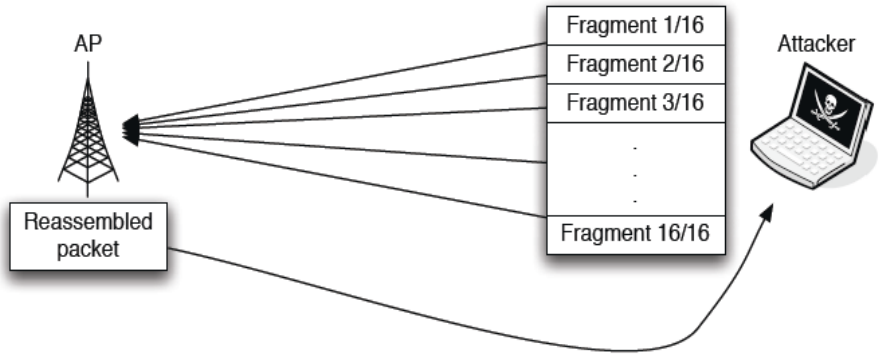

Figura 5.5: *Fragmentation attack*

Idealmente un attaccante vorrebbe 1500 byte di keystream equivalente a una MTU (*Maximum Transmission Unit*) di Ethernet. Sfruttando la framentazione prevista dall'802.11 questo è possibile in pocchissimo tempo. L'attaccante genera unpacchetto broadcast di 64 byte e lo invia all'AP in 16 frammenti. L'AP aspetta finché riceve tutti i pacchetti e poi li riassembla in un unico pacchetto. Dal momento che si tratta di un pacchetto broadcast esso viene prima decriptato e poi criptato nuovamente, ma con un IV nuovo, per poi esere diffuso in rete in un unico frammento. Questo pacchetto avrà una dimensione di 68 byte (64 byte di dati e 4 byte di ICV). L'attaccante può catturare questo pacchetto e conoscendo il plaintext puo ottenere 68 byte di *keystream* per il nuovo IV. Ripetendo la tecnica a questo punto si possono ottenere 1500 byte di *keystream*. Usando questo metodo un attacante ha bisogno solamente di trasmettere 34 pacchetti e riceverne 4 per ottenere 1500 byte di *keystream*. Questo metodo è molto più veloce del ChopChop attack che richiede 128 pacchetti in media per recuperare un byte.

A differenza degli attacchi precedenti, in cui nessun requisito legato all'architettura era richiesto per la loro esecuzione, in questo attacco le cose sono differenti. La vulnerabilità qui nasce grazie a specifiche caratteristiche dei protocolli LLC e SNAP che in Hiperlan 2 non esitono. In aggiunta, cosa fondamentale, Hiperlan 2 non supporta la frammentazione a livello MAC. Possiamo quindi sotenere che il *Fragmentation attack* non è applicabile nelle reti Hiperlan 2.

5.1.2 Vulnerabilità legate all'algoritmo$RC4$

In questa sezione analizzeremmo le debolezze di WEP derivanti dall'algoritmo di *key scheduling* di $RC4$. Riguardo l'analisi di $RC4$ il lettore appassionato che desidera approfondire i dettagli troverà con molta facilità in rete anche paper risalenti precedentemente alla pubblicazione dell'articolo per eccellenza [23] da cui avranno origine tutti gli attacchi basati sulle vulnerabilità di $RC4$. Come vedremmo in seguito tutti questi attacchi avranno un requisito di partenza comune che sarà evidenziato di seguito.

Attacco FMS

Nello stesso anno in cui furono evidenziate le vulnerabilità specifiche del protocollo WEP, i ricercatori Scott Fluhrer, Itsik Mantin e Adi Shamir hanno dimostrato l' esistenza di un' importante debolezza nell'algoritmo di *key scheduling* di $RC4$. Essa riguarda l'esistenza di larghe classi di *chiavi deboli*, in cui una piccola parte della chiave segreta determina un grande numero di bit di output del *key stream*. Questo attacco è noto come l'*FMS attack*.
Esso lavora basandosi solamente sul primo byte del *keystream* di $RC4$. L'equazione di questo byte può essere scritta come [23] dove $S[i]$ rappresenta il byte $i-esimo$ del vettore di stato di $RC4$. Osservando questi valori quando viene usata una *chiave debole* possiamo dedurre informazioni sulla chiave. In $[refart]$ gli autori elencano un insime di condizioni mediante cui si possono ricercare gli IV che determinano le *chiavi deboli*. Queste condizioni vengono definite *resolved condition*. Poco tempo dopo la pubblicazione dell'articolo di Fluhrer, Mantin e Shamir riguardante la scoperta di vulnerabilità nell'algoritmo $RC4$, fu condotto un primo esperimento, da parte dei ricercatori Stubblefield, Ioannidis e Rubin, per verificare concretamente quanto scoperto [24]. Essi osservarono che in uno stato di *resolved condition* il valore del prossimo byte della chiave era dato con una probabilità non trascurabile(5%) dalla seguente equazione:

$$K[B] = S_{B+2}^{-1}[Out] - j_{B+2} - S_{B+2}[B+3]$$

dove K è la chiave, B il byte corrente da indovinare, Out è il primo output del PRGN e S^{-1} è la posizione in S dove appare.
L'esperimento condotto si poneva principalmente due obiettivi: prima di tutto si voleva verificare che l'attacco descritto potesse essere funzionante anche nel mondo reale e non solo in teoria. In secondo luogo si voleva capire quanto potesse essere facile ed economico condurre questo tipo di attacco. La prima fase dell'esperimento ha previsto la simulazione di un attacco a RC4. Dopo appena due ore dedicate alla scrittura del codice, si fece partire

la simulazione che ne mostrò l'effettivo funzionamento. La parte che invece richiese meggiore tempo riguardò la cattura dei pacchetti codificati con WEP. A questo scopo fu acquistata una scheda di rete dotata di chipset Prism II del costo di 100 $, la quale permetteva di effettuare molte computazioni via software e inoltre consentiva la cattura di pacchetti grezzi. L'ultima parte dell'esperimento consistette nell'individuare il valore del primo byte del messaggio in chiaro. Questa operazione fu estremamente semplice, infatti il traffico maggiormente presente nella rete locale era di tipo IP e ARP. I quali erano però tutti incapsulati con un header Ethernet, come previsto dal protocollo SNAP. Tale header iniziava sempre con il valore $0xAA$ ed esso costituiva il primo byte del messaggio in chiaro.
Avendo scoperto il valore del primo byte, si usò il seguente algoritmo per individuare la chiave WEP utilizzata durante l'esperimento:

RecoverWEPKey()
  $Key[0 \dots KeySize] = 0$
  for $KeyByte = 0 \dots KeySize$
    $Counts[0 \dots 255] = 0$
    foreach packet $\rightarrow P$
      if $P.IV \in \{(KeyByte + 3, \texttt{0xFF}, N) \mid N \in \texttt{0x00} \dots \texttt{0xFF}\}$
        $Counts[\text{SimulateResolved}(P, Key, KeyByte)] \mathrel{+}= 1$
    $Key[KeyByte] = \text{IndexOfMaximumElement}(Counts)$
  return Key

SimulateResolved$(P, Key, KeyByte)$
  $K = P.IV \cdot Key$
  for $i = 0 \dots N - 1$
    $S[i] = i$
  for $i = 0 \dots KeyByte$
    $j = j + S[i] + K[i \bmod l]$
    swap$(S[i], S[j])$
  return $S^{-1}_{B+2}[P.Out] - j_{B+2} - S_{B+2}[B + 3]$

Figura 5.6: *Algoritmo key recovery*

Questo tipo di attacco può essere utilizzato qualunque sia la lunghezza dello IV; si presta quindi ad essere effettuato sia per attaccare il WEP, che usa un IV di 24 bit, ma anche contro la sua proposta di estensione, a volte

chiamata WEP2, che invece utilizza un IV di 128 bit. Con questo esperimento, Stubblefield, Ioannidis e Rubin dimostrarono l'effettiva applicabilità dell'attacco all'algoritmo $RC4$, mostrando come fosse possibile ricavare la chiave WEP utilizzata in una determinata rete wireless. Riuscirono inoltre a condurre l'esperimento con mezzi economici ed alla portata di chiunque, potendo quindi concludere che il protocollo WEP fosse completamente insicuro.

Attacchi KoreK

Come affermato in precedenza, nel 2004, vengono publicati su internet gli attacchi KoreK. Il secondo attacco descrive 17 differenti attacchi a WEP classificati secondo le seguenti categorie:

- Recupero della chiave basato sul primo byte del *keystream* del PRNG (FMS attack).
- Recupero della chiave basato sul primo e secondo byte del *keystream* del PRNG.
- Attacchi inversi.

Come possiamo notare la prima categoria ha un approccio è simile al FMS attack. L'FMS e in effetti considerato parte del attacco KoreK ed è stato nominato con la sigla A_s5_1. In aggiunta alla correlazione trovata nel FMS attack, KoreK ne trovo molte altre che aumentarono la probabilità di indovinare il prossimo byte della chiave fino al 14%. La seconda categoria di attacchi trovati da KoreK è molto simile alla prima . La sola differenza è che in questo caso l'attacco si basa anche sul secondo byte del *keystream.* piuttosto che solo sul primo. La terza categoria consiste in uno specifico attacco noto come A_neg *attack.* L'riginalità di questo attacco consiste nella riduzione dello spazio di ricerca delle chiavi riuscendo ad eliminarne alcune secondo particolari congetture. Una trattazione più dettagliata che , esula dallo scpo di questa tesi, la si può trovare in Chaabonui [30]. Con gli attacchi KoreK si riusci ad abbassare il numero di pacchetti catturati necessari da 4,000,000/6,000,000 per una probabilità di sucesso dell' attacco del 5% a 700,000 per una probabilità di sucesso del 50%.

Attacco di Klein

Nel 2005 Andreas Klein presentò nel suo paper [27] una nuva serie di correlazioni tra il *keystream* e la chiave in aggiunta a quelle scoperte precedentemente da KoreK. L'articolo descrive come l'ggetto del suo attacco sia

migliorare l'attacco FMS in modo tale che sia applicabile anche alle reti in cui non sono presenti *chiavi deboli*. Anche in questo caso il punto di partenza è la conoscenza del primo byte del *keystream*.

Attacco PTW

Nel 2007 Tews, Weinmann e Pyshkin pubblicarono quello che sarà ricordato come il *PTW attack*. L'attacco si basa su una estensione dell'analisi di Klein. Nel loro paper intitolato *Breaking 104-bit WEP in less than 60 seconds*, l'attacco recupera completamente la chiave con una probabilità di successo de 50% con meno di 40,000 frame. La probabilità arriva al 95% ne caso i frame catturati siano 85,000.

Attacco PTW migliorato

Nel 2008, Beck e Tews presentano la bozza di due attacchi relativi a WEPe WPA/TKIP [29]. Al momento non esistono molte informazioni su questo attacco sebbene esista una implementazione nel famoso tool Aircrack. Il PTW si basa fondamentalmente su una riscrittura più efficace ed efficiente delle correlazioni scoperte da KoreK. L'effetto di questo attacco si può osservare nell figura seguente dove per una probabilità di successo del 50%occorrono appena 24,200 pacchetti.

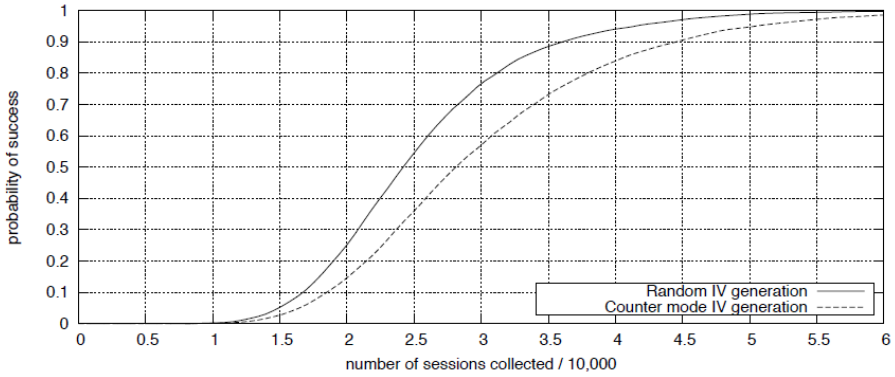

Figura 5.7: *Probabilità di successo del attacco PTW migliorato*

Come è facile notare tutti gli attacchi presentati precedentemente non sono che una specializzazione del *FMS attack*. Essi hanno una caratteristica in comune. Affinché possano funzionare nel mondo reale neccessitano della conoscenza del primo byte del *keystream*. Tale conoscenza avviene sfrttando, come già osservato in precedenza, la particolarità dei frame IEEE 802.11.

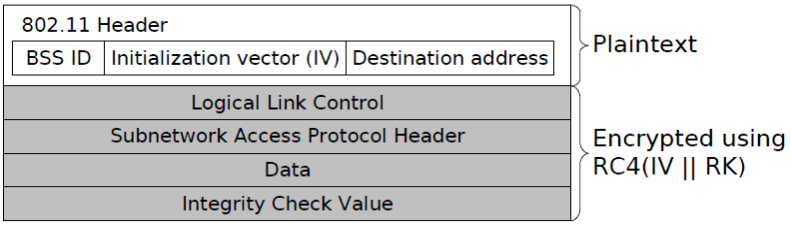

Figura 5.8: *Frame 802.11 criptato con WEP*

Come si può notare dalla figura 5.8 le costanti contenute nei campi dell'intestazione LLC/SNAP vengono criptate dal protocollo WEP. Quindi siamo in presenza di una coppia di *plaintext* noto e *ciphertext* e di conseguenza possiamo conoscere il primo byte del *keystream*.

L'analisi compiuta sullo standard Hiperlan 2 non ha portato alla luce campi conteneti valori fissi attraverso cui si possa applicare il procedimento visto precedentemente per ottenere il primo byte del *keystream*. Di conseguenza possiamo affermare che tali attacchi non possano essere trasposti su Hiperlan 2. Naturalmente questo risultato potrebbe essere rivalutato nel caso di un'analisi ancora più accurata su tale standard.

5.2 Debolezze del WPA

Per analizzare le eventuali debolezze di questo standard è neccessaria una conoscenza un pò più approfondita di alcune delle sue componenti. Wi-Fi Protected Access (WPA)/Temporal Key Integrity Protocol (TKIP) nasce con lo scopo di colmare le lacune di WEP. Come detto in precedenza in WPA viene condivisa una *master key* MK mediante cui vengono generati due tipi di chiavi: una chiave K^* di 64 bit per il Mes*sage Integrity Check* (MIC) e una K di 128 bit per la criptazione dei pacchetti. In figua 4 è presentato uno schema riassuntivo delle componenti di WPA in trasmissione. L'algoritmo Micheal ricevendo in ingresso la MIC *key* K^* e l'MSDU (*MAC Service Data Unit*) calcola il MIC che viene in seguto concatenato alla MSDU come segue:

$$MSDU||micheal\,(K^*, MSDU)$$

dove $micheal\,(K^*, MSDU)$ è il MIC di 64 bit e $||$ è l'operatore di concatenazione. L'MSDU con il MIC accodato viene poi framentata nelle MPDU (*MAC Protocoll Data Unit*) ed a ciascuna di essa è applicato il CRC32. Si forma quindi:

$$MPDU||CRC32(MPDU)$$

Il $CRC32(MPDU)$ è costituito da 32 bit.
La criptazione in WPA e esguita su ciascuna $MPDU||CRC32(MPDU)$. Una *chiave di pacchetto* PK è generata dal vettore di inizializzazione di 48 bit (IV), dalla chiave K e dall'indirizzo MAC usando una specifica funzione *hash* per WPA. Gli IV per ciascun pacchetto sono differenti. In WPA l'IV è usato come contatore ed è chiamato *TKIP secuence counter* (TSC). Come sappiamo in WPA l'algoritmo usato per la criptazione è l'$RC4$. Questo genera il *keystream*, ricevendo in ingresso PK e l'IV, $Z = (Z_1, Z_2, .., Z_L)$ dove Z_i è un byte variabile ed L è la lunghezza del *plaintext*. Il *keystream* e il *plaintext* $P = (P_1, P_2, .., P_L)$ generano il ciphertext $C = (C_1, C_2, .., C_L)$ secondo:

$$C_i = P_i \oplus Z_i$$

per $(i = 1, 2, .., L)$. Possiamo quindi esprimere la ciptazione di WPA come:

$$C = (MPDU||CRC32(MPDU)) \oplus RC4(PK, IV)$$

e al ricevente viene trasmesso $C||IV$.
Passiamo ora ad analizzare quello che sucede in ricezione aiutandoci con lo schema di figura 5.9.

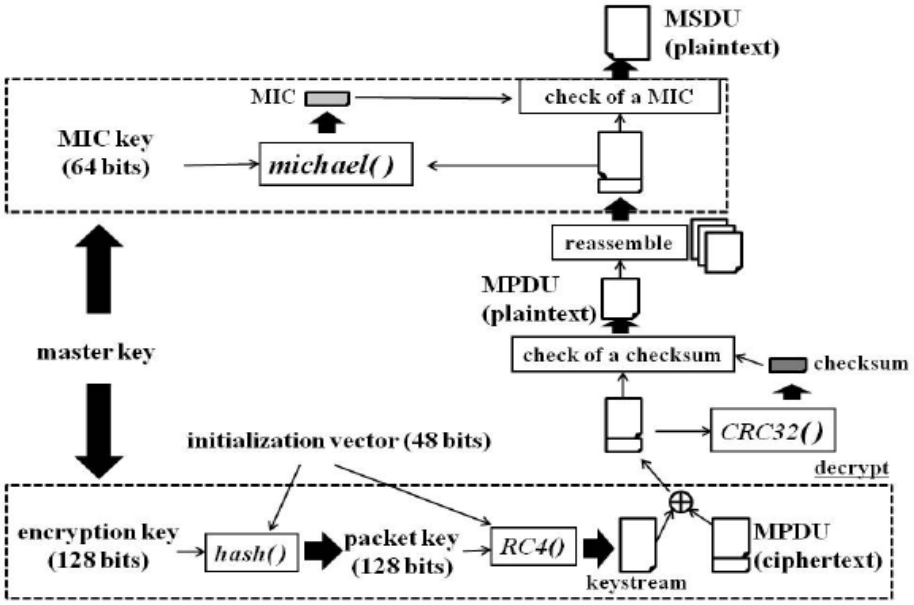

Figura 5.9: *Componenti WPA in ricezione*

In ricezione quindi ci troviamo le MPDU criptate ed a ciascuna di esse il proprio IV in chiaro. L'IV è confrontato con il TSC *counter* il quale ha il valore corrispondente all'IV della più recente MPDU accettata. Se l'IV ricevuto ha un valore ninore o uguale al TSC *counter* allora la MPDU criptata viene scartata. In ricezione viene generato un *keystream* $\hat{Z}$ mediante l'IV ricevuto e la PK uguale a quello in trasmissione Z. Il *plaintext* P è ottenuto usando $\hat{Z} = Z$ come segue:

$$P_i = P_i \oplus Z_i \oplus Z_i = C_i \oplus Z_i$$

per $(i = 1, 2, .., L)$. Possiamo allora scrivere la decriptazione in WPA come:

$$(MPDU||CRC32(MPDU)) = C \oplus RC4(PK, IV)$$

Il ricevente calcola il checksum dalla MPDU ricevuta e lo confronta con quello concatenato. Se i due differiscono la MPDU ricevuta viene scartata. Notiamo che il ricevente non spedisce un messaggio di errore al trasmitente quando si verifica questo fatto.
Quando tutte le MPDU sono ricevute vengono riassemblate formando cosi la MSDU. A questo punto il ricevente calcola il MIC sulla MSDU usando la K^*. Il nuovo MIC viene confrontato con il MIC concatenato alla MSDU. Se i due MIC risultano differenti tutte le MPDU ricevute corrispondenti alla MSDU vengono scartate e viene inviato un messaggio di errore al trasmitente. In WPA la K^* è cambiata non appena si verificano due errori di questo tipo in meno di un minuto. Quando l'MSDU è accettata il TSC *counter* viene aggiornato al valore più grande dell'IV relativo alle MPDU.

5.2.1 Attacco di Beck e Tews

Nel 2008, precisamente a novembre, Beck and Tews rilasciano un articolo intitolato *Practical Attacks Against WEP and WPA*. In questo paper oltre a una proposta migliorativa del *PTW attack* , come visto in precedenza, essi presentano una versione modificata dell'attacco *ChoChop* diretta al TKIP. E' importante sottolineare che questo attacco non provede al recupero della encription key e non è quindi paragonabile agli attacchi FMS o PTW. Esso abilita l'attaccante a decriptare un pacchetto ARP request diretto dall'AP verso una station (STA). Per fare questo l'attaccante ottiene la *keystream* e la MIC key che possono essere usate per creare ed iniettare pacchetti modificati dall'AP verso le stazioni della rete. L'attacco presenta però un limite: è applicabile solo alle reti IEEE 802.11 che supportano QoS.

WPA/TKIP come spiegato in precedenza per rispondere ai *replay attack* implementa il TSC *counter*. Questo significa che l'attacco *ChopChop* non può più essere riproposto. Beck e Tews hanno però scoperto che nelle reti in cui è abilitata la QoS questo non è del tutto vero. Lo standard IEEE 802.11e QoS permette 8 canali diversi per flussi di dati differenti e ciascun canale ha un TSC *counter* indipendente. Supponiamo che un attaccante abbia catturato un pacchetto criptato proveniente dal canale 0 che ha un $IV = 15$. In questo caso non si potrebbe eseguire il *ChopChop* su tale canale in quanto il TSC *counter* di quel canale è aggiornato a 15. In una condizione del genere si può eseguire l'attacco in un canale diverso ma che abbia un TSC *counter* minore di 15.

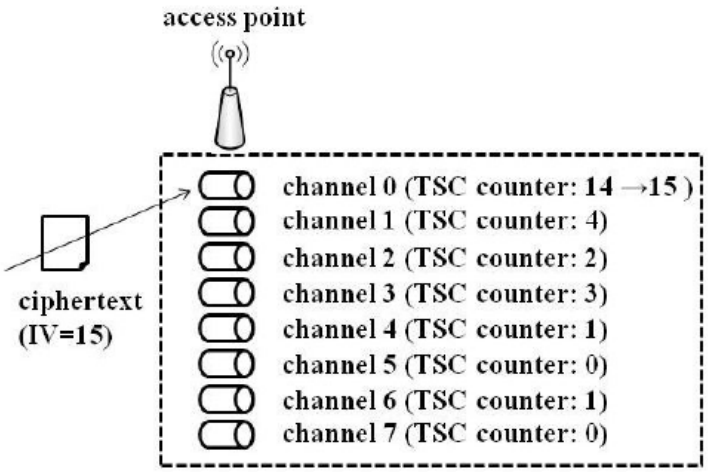

Figura 5.10: *Implementazione di WPA che supporta le funzionalità di IEEE 802.11e QoS*

Nella maggior parte delle reti in cui è abilitata la QoS i dati di tendenza vengono trasmessi sul canale 0, ciò significa che il resto dei canali ha molto probabilmente un TSC *counter* più basso del canale 0 e di conseguenza sono vulnerabili al *ChopChop attack*.
La figura seguente riassume i passi fondamentali dell'attacco.

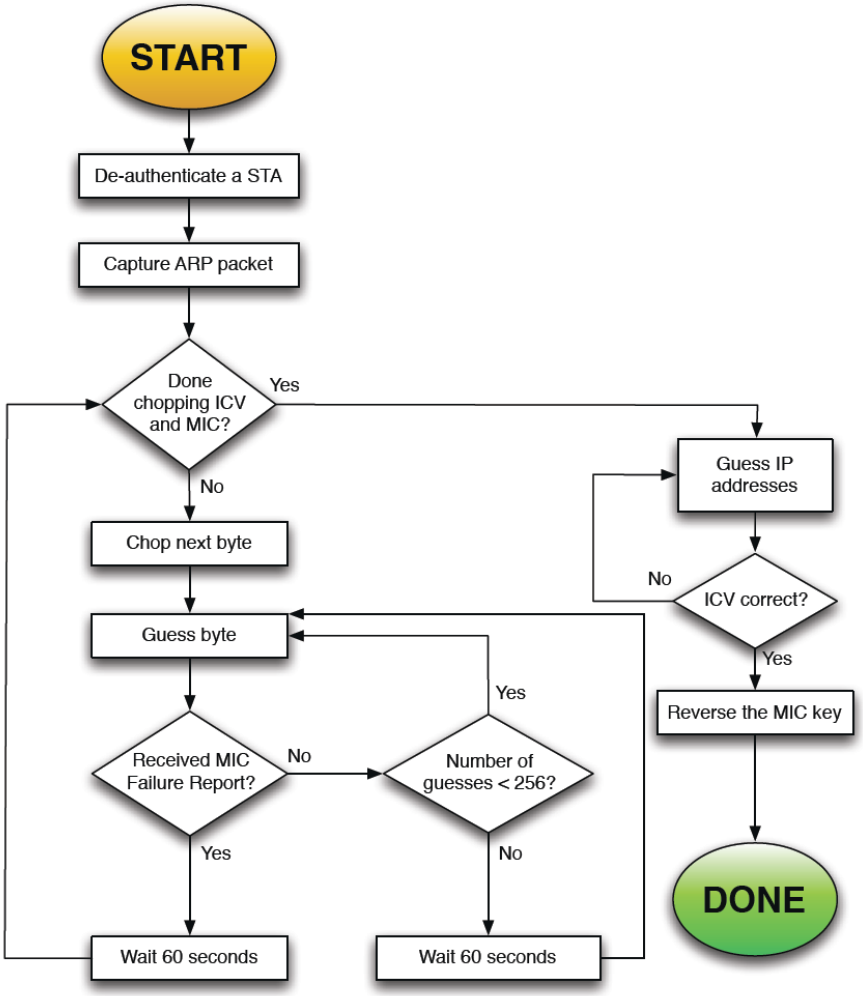

Figura 5.11: *Flowchart dell'attacco*

Possiamo dividerlo sostanzialmente in quattro fasi:

- Deautenticazione del client
- Attacco *ChopChop* modificato
- Individuazione byte rimanenti
- Inversione algoritmo di MICHAEL

Deautenticazione del client

Prima che l'attacco inizi si provvede a deautenticare la STA. Eseguendo questa operazione si costringe il client e l'AP ad eseguire tutte quelle operazioni neccessarie alla nuova associazione tra cui lo scambio di pacchetti ARP su cui gli autori del paper consigliano di focalzzarel'attenzione per la riuscita dell'attacco. Questa scelta non è casuale in quanto tali pacchetti sono facili da individuare grazie alla loro lunghezza fissa e molti dei dati in essi contenuti sono acilmente predicibili. Inoltre grazie alla loro ridotta dimensione non vengono frammentati.

5.2.2 Attacco *ChopChop* modificato

Una volta catturato un pacchetto ARP trasmesso dall'AP al client una versione modificata del *ChopChop attack* può avere luogo. Il *Chopchop attack* modificato lavora troncando l'ultimo byte del pacchetto cioè alla stessa maniera del *Chopchop attack* convenzionale. In contrasto con l'attacco convenzionale si ha che l'attacco modificato funziona con il traffico diretto da AP a STA. Il nostro *oracolo* ora è il client. La ragione di questo risiede nel fatto che solo il client è in grado di trasmettere il MIC *failure report*. Lo scopo per cui nasce l'attacco modificato è dovuto sostanzialmente alla contromisura inserita su WPA per contrastare al sua versione origiraria. Infatti, se un pacchetto presenta un ICV non valido il pacchetto viene scartato senza report di errore (diversamente da WEP). Nel caso contrario, con ICV corretto, ma con MIC scorretto un report di errore viene notificato all'AP e il TSC *counter* non viene incrementato.
Bisogna tenere in considerazione due punti fondamentali :

- Si possono avere al massimo 2 report di errore in 60 secondi prima che l'AP interrompa la comunicazione e riparta la rinegoziazione delle chiavi cioè il *Key Renewal Interval* che definisce il tempo di rinovamento della *Pairwise Temporal Key* (PTK).

- L'attacco viene lanciato su un canale diverso da quello da cui il pacchetto proviene e che presenta un TSC *counter* minore del IV del pacchetto.

Quindi nel caso in cui partano le contromisure del MIC l'attaccante deve attendere 60 secondi prima di inoltrare il pacchetto contenente la supposizione del nuovo byte troncato.

Individuazione byte rimanenti

Come spiegato nel paragrafo precedente il *Chopchop attack* modificato deve aspettare 60 secondi tra ogni byte troncato in modo da evitare che le contromisure MIC vengano attivate. Considerando che di solito la *Key Renewal Interval* dura 60 minuti, mediante l'attacco *ChoChop* modificato possiamo decriptare al massimo 60 byte.
Per calcolare invece il il MIC e il checksum occorono invece 12 minuti dal momento che il MIC è di 8 byte e l'ICV di 4 byte. Il resto dei byte del pacchetto ARP (esattamente gli indirizzi IP del tasmettitore e del rievitore) è calcolato senza l'utilizzo del *ChoChop attack*. Basta infatti formare tutti i candidati pachetti ARP con i restanti 2 byte sconosciuti (2^{16}). Per ciascuno di essi si calcola il checksum e lo si confronta con quello recuperato attraverso il *ChopChop attack*. Se i due ICV coincidono significa che si è indovinato il pachetto giusto.

Inversione algoritmo di MICHAEL

Al fine di iniettare pacchetti modificati in rete l'attaccante deve in qualche maniera recuperare la MIC key. Dal momento che l'algoritmo di MICHAEL non fu progettato per essere una funzione a senso unico come le funzioni hash esso è facilmene invertibile. Avendo a disposizione il MIC e il *plaintext*, come nel nostro caso, e facile recuperare la MIC *key* K^*.

Quindi, ome menzionato in precedenza, l'attacco di WPA non recupera la chiave K. L'attacco è in grado di recuperare il *keystream* di un pacchetto ARP e la chiave K^*. Conoscendo questi dati l'attaccante è in grado di creare pachetti ARP ed inoltrarli in rete. Non può inoltrare pacchetti di lunghezza maggiore in quanto la dimensione del *keystream* corrisponde a quella del pacchetto ARP. Una possibile applicazione di questa tecnica è quella di cambiare l'ARP cache che è una parte di vitale importanza per il routing e l'indirizzamento nella rete.

Non rimane ora che capire se questo tipo di attacco è applicabile alle reti Hiperlan 2. L'unico requisito affinchè tale attacco abbia successo è che la rete supporti la QoS. Hiperlan 2 supporta la QoS ed implementa tale concetto secondo lo standard 802.1p (dichiarato anche nei data sheet dei dispositivi), come aviene nelle reti 802.11e. Di conseguenza possiamo affermare che questo attacco può essere applicato nelle reti Hipelan 2.

5.2.3 A pratical messagge falsification attack

Allo scopo di mitigare la limitazione derivante dal vincolo della QoS, ad agosto del 2009, i ricercatori Ohigashi e Morii presentarono nel loro paper[31] una soluzione a tale problema. La soluzione proposta consiste nell'applicazione del *Beck-Tews attack* al MITM (*Man-in-the-middle*) attack.
Quest'ultimo tipo di attacco consiste nel dirottare il traffico generato durante la comunicazione tra due entità verso una terza (attaccante) la quale fingera' di essere l'end-point legittimo della comunicazione.
Nel loro articolo i due autori presentano come requisito che la comunicazione tra access point e station avvenga solo attraverso l'attacante. Nel caso contrario tale comunicazione non può avere luogo a causa della distanza tra i leggittimi utenti della comunicazione.

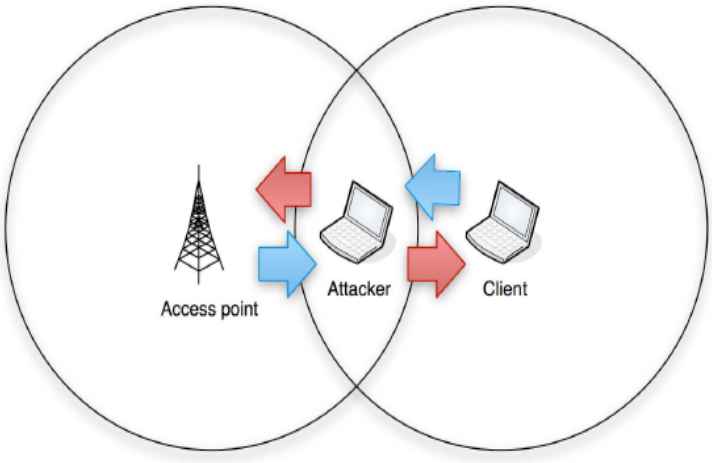

Figura 5.12: *Scenario del MIMT*

In questo scenario l'attaccante, per ottimizzare le tempistiche, può aggire in tre modalità differenti:

- Ripetitore
- Recupero MIC key
- Falsificazione di messaggio

Nella prima modalità l'attacante non fa altro che inoltrare i pacchetti tra STA ed AP e viceversa.
Nella seconda modalità l'attaccante tenta di ottenere la MIC *key* K^* eseguendo il *ChopChop attack* basato sul MIMT. Di solito l'attacante entra in

questa modalità quando c'è uno scmbio di pacchetti ARP che non hanno una reale influenza sulla comunicazione. Nel caso in cui non venga eseguità questa modalità l'attaccante rientra nella prima. In questo modo si riduce il tempo di interruzione della comunicazione aumentando l'invisibilità dell'attacco.
Una volta che il recupero della MIC *key* ha avuto successo l'attaccante entra nell'ultima modalità dove può falsificare ed inoltrare pacchetti criptati nella comunicazione.
Allo stato attuale non esiste un'implementazione di tale attacco. Gli autori sostengono che usando questa strateggia l'interruzzione della comunicazione tra AP ed STA risulterebbe nel migliore dei casi di soli 4 minuti.

Sicuramente per quanto riguarda il nostro caso questo attacco potrebbe essere funzionante nelle reti WLAN basate su Hiperlan 2. Infatti cadendo il vincolo della QoS il *ChopChop attack* è eseguibile in quanto richiede solo la capacità di modifica di pacchetti TKIP.

5.3 Debolezze del WPA2

Per quanto riguarda questo standard al momento in letteratura non sono stati pubblicati attacchi che mettano a pericolo la sicureza delle informazioni degli utenti che lo utilizzano.
E' importante notare che usare il protocollo WPA2 o WPA non fornisce nessuna protezione contro le tecnologie sottostanti, come la radio frequency jamming, DoS attraverso deautenticazione, de-associazione in cui anche Hiperlan2 è soggetta.

Capitolo 6

Conclusioni

In questo elaborato abbiamo compiuto una lunga analisi degli standard di sicurezza, utilizzati per proteggere le informazioni che circolano nelle reti wireless, ed i loro relativi punti di debolezza. Abbiamo conosciuto gli attacchi più significativi presenti in letteratura seguendo un'evoluzione cronologica che ci ha portato fino ai tempi attuali. Attraverso tale caratterizzazione temporale si è potuto constatare come tali standard ed attacchi siano piuttosto recenti. Questo fatto ha indirettamente protetto i sistemi che usano Hiperlan 2. Essendo definito lo standard Hiperlan2 antecedentemente alla nascita di questi attacchi non è stato sottoposto al vaglio della comunità scientifica per quanto riguarda la sicurezza ed in particolar modo in abbinamento con il WEP e l'802.11i. L'ttenzione della comunità scientifica si è concentrata, dato il forte radicamento, alle reti wireless che utilizzano lo standard 802.11. L'obbiettivo della tesi consisteva nell'individuare se e quali attacchi noti fino a questo momento potessere essere applicati allo standard Hiperlan 2. Come abbiamo potuto constatare gli attacchi più devastanti (FMS, PTW) sono fortemente legati all'architettura definita dallo standard utilizzato per le comunicazioni wireless. Non è possibili eseguire una trasposizione di tali attacchi su uno standard diverso anche se tale possibilità non si può escludere completamente. In Hiperlan 2, come abbiamo visto, possiamo escludere tale possibilità anche se una ricerca più approfondita potrebbe contradire tale affermazione. Una certezza sicuramente maggiore abbiamo nel caso del *ChopChop attack*. Esso può essere applicato sia su WEP che su WPA ed in entrambi gli standard (802.11, Hiperlan 2). Tale possibilità è garantita dalle caratteristiche generali che lo distinguono e che non si basano sull'architettura su cui si applica.
Per quanto riquarda WPA2 la nostra ricerca ha datto esito positvo. Positivo nel senso che non si sono ancora scoperte falle in questo standard. Sicuramente molta della forza di questo standard si fonda sulla robustezza

dell'algoritmo di crittografia utilizzato (AES).
Possiamo quindi concludere, da un punto di vista protocollare, che i collegamenti in ponte radio realizzati dalla 360srl presentano un grado di sicurezza molto elevato specialmente nel caso di utilizzo di WPA2 che in questa azienda è sicuramente privilegiato.

Bibliografia

[1] www.wikipedia.it

[2] www.wi-fi.org

[3] www.clusit.it

[4] www.digitpa.gov.it

[5] W.Stallings. *Crittografia e sicurezza delle reti.* Mc Graw Hill, 2004

[6] G.Schaefer. *Security in fixed and wireless networks.* Wiley, 2003

[7] ETR0230002 V0.2.0: Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; *System Overview.*

[8] ETSI TC-RES/RES10: Radio Equipment and Systems (RES): HIgh PErformance Radio Local Area Networks (HIPERLANs) *Requirements and Architecture*; January 1997.

[9] ETSI TS 101 475-2 V1.3.1, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; *Physical(PHY) Layer*; October 2001.

[10] ETSI TS 101 761-1 V1.3.1, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; *Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions*, January 2002.

[11] ETSI TS 101 493-2 V1.1.1: Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; *Packet Based Convergence Layer; Part 1:Common Part*,January 2002.

[12] ETSI TS 101 761-2 V1.3.1, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; *Data Link Control (DLC) Layer; Part 2: Radio Link Control (RLC)* January 2002.

[13] TS 101 493-2, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; *Packet based Convergence Layer; Part 2: Ethernet Service Specific Convergence Sublayer (SSCS).*

[14] J.Jush, G.Malmgren, P. Schramm, J. Torsner, *Hiperlan type 2 for broadband wireless communication*, November 2000.

[15] K. Oikonomou, J. Tenidis, I Stavrakakis, *A Mechanism to Enable Differentiated Services QoS in HIPERLAN/2*, June 2001, Bucharest, Romania.

[16] J. Jush1, P. Schramm1, U. Wachsmann1, F. Wenger2 *Structure and Performance of the HIPERLAN/2 Physical Layer*, April 2000.

[17] C.N.I.P.A. *Linee guida per l'introduzione delle tecnologie wireless nella Pubblica Amministrazione.* Marzo 2008.

[18] S. Paggi. *Progettazione ponti radio numerici terrestri.* Telecom Italia, Roma 20 Maggio 2004

[19] O. *Mirabella. Principi di progettazione di reti geografiche di trasporto wireless.* Medianet Comunicazioni S.R.L.

[20] G.Lehembre. *Wi-fi security: Wep, Wpa and Wpa2.* hakin9, (2), 2006.

[21] Arunesh Mishra, William, A. Arbaugh, *An Initial Security Analysis of The IEEE 802.1X Standard*, University of Meryland, 2002

[22] N. Borisov, I. Goldberg, and D.Wagner. *Intercepting Mobile Communications: The Insecurity of 802.11.* In Proc. ACM Mobicom, Rome, Italy, July 2001

[23] S. Fluhrer, I. Mantin, and A. Shamir. *Weaknesses in the Key Scheduling Algorithm of RC4.* Lecture Notes in Computer Science, 2259:1-24, 2001.

[24] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. *A key recovery attack on the 802.11b wired equivalent privacy protocol (wep).* ACM Trans. Inf. Syst. Secur., 7(2):319-332, 2004.

[25] KoreK. *Chopchop (Experimental WEP attacks)*, 2004. http://www.netstumbler.org/showthread.php?t=12489.

[26] KoreK. *Next generation of WEP attacks?*, 2004. http://www.netstumbler.org/showpost.php?p=93942postcount=35.

[27] Andreas Klein. *Attacks on the RC4 stream cipher*. Submitted to Designs, Codes and Cryptography, 2007.

[28] Andrea Bittau. *The Fragmentation Attack in practice*. September 2005

[29] E. Tews and M. Beck. *Practical attacks against WEP and WPA*. In Proceedings of the second ACM conference on Wireless network security, WiSec '09, pages 79-86, New York, NY, USA, 2009. ACM

[30] R. Chaabouni. *Break WEP Faster with Statistical Analysis*,June 2006.

[31] Ohigashi,M. Morii. *A practical message falsification attack on wpa*. In: Proceedings of 2009 Joint Workshop on Information Security. (2009)

[32] E. Tews, R. Weinmann, and A. Pyshkin, *Breaking 104 bit WEP in less than 60 seconds*, Cryptology ePrint, 2007

[33] A. Stubblefield, J. Ioannidis, and A. Rubin. *Using the Fluhrer, Mantin, and Shamir Attack to Break WEP*. In Proc. Symposium on Network and Distributed System Security, San Diego, California, Feb 2001. Internet Society.

[34] Andrea Bittau, Mark Handley, and Joshua Lackey. *The final nail in WEP's cofin*. In IEEE Symposium on Security and Privacy. IEEE Computer Society, 2006.

[35] W. A. Arbaugh, N. Shankar, and Y. J. Wan. *Your 802.11 Wireless Network has No Clothes*, 2001.

[36] Erik Tews. *Attacks on the wep protocol*. Cryptology e Print Archive, Report 2007/471, 2007.

[37] N. C.Winget, R. Housley, D. Wagner, J. Walker. *Security flaws in 802.11 data link protocols*. Comunication of the ACM,Vol. 46, No. 5,May 2003.

[38] Frank H. Katz. *WPA vs. WPA2: Is WPA2 Really an Improvement on WPA?*, November 2009.

[39] A. H. Laskari, M. M. S. Danesha. *Survey on Wireless Security protocols (WEP,WPA and WPA2/802.11i)*, September, 2009.

[40] C. He J. Mitchell. *Security Analysis and Improvements for IEEE 802.11i*, September 2007.

[41] D.Schauenberg, *Wi-Fi Protected Access: Overview and State of the Art Attacks*, August 2010.