Università degli Studi di Salerno

Facoltà di Scienze Matematiche Fisiche e Naturali

Corso di Laurea Specialistica in Informatica

Sicurezza in Bluetooth

CORSO DI SICUREZZA SU RETI 2

A. A . 2008/09

PROF. ALFREDO DE SANTIS

Carmine Spiniello [0521000562]

Sommario

1. Introduzione

1.1 Storia dello Standard

La storia di questo appellativo ci rimanda indietro nei secoli. Harald Bluetooth era il nome di un antico re vichingo, vissuto tra il 940 ed il 981 D.C. La particolarità di Harald Bluetooth fu l'unione di due paesi tanto diversi tra loro, come Danimarca e Norvegia, sotto un unico regno, con lui a capo. Inoltre Harald portò a termine la completa cristianizzazione dei due paesi. In tal modo il nome Bluetooth è rimasto a significare l'unione di due realtà diverse, ma con la volontà di collaborare per un futuro migliore. Così Bluetooth è stato preso come simbolo per la nuova tecnologia, che porterà ad unire differenti dispositivi, anche molto diversi tra loro, in un'unica grande rete senza fili. Anche il simbolo del Bluetooth rimanda all'antico re vichingo. Consiste, infatti, nei due simboli celtici raffiguranti la H e la B, le iniziali di Harald Bluetooth. Le basi dell'odierno Bluetooth le getta Ericsson nel 1994. L'azienda svedese iniziò una ricerca per trovare una nuova interfaccia di comunicazione ad onde radio a basso costo, tra i cellulari ed i diversi accessori, che soppiantasse l'IRDA. L'idea era quella che un minuscolo ricevitore radio immesso sia nel cellulare sia nell'eventuale dispositivo da unire, avrebbero sostituito i vari fili usati all'epoca. Dopo circa un anno dal progetto iniziale, lo sviluppo della nuova tecnologia entrò nella fase operativa ed il lavoro passò alla sezione ingegneristica dell'Ericsson. Il progetto iniziale di fornire nuovi accessori per telefonia mobile ad onde radio, fu soppiantato da un progetto più ambizioso, la creazione di una vera e propria nuova tecnologia per far comunicare qualsiasi tipo di dispositivo a breve distanza. Ericsson capì l'importanza degli studi effettuati e decise di allargare il gruppo di sviluppo ad altri partner. Si arriva quindi al 1998, con la formazione del SIG (Special Interest Group) insieme a Nokia, Intel, IBM e Toshiba. Inizialmente Ericsson ha fornito un importante contributo per quanto riguarda la tecnologia radio. Toshiba ed IBM hanno lavorato per sviluppare un protocollo comune per l'integrazione del Bluetooth all'interno di dispositivi portatili. Nokia si è occupata della trasmissione dati e del software ed Intel della progettazione dei nuovi chip necessari. Gli obiettivi iniziali del gruppo erano aiutare lo sviluppo di una nuova tecnologia ad onde corte per far comunicare a breve distanza, creando uno standard fisso, ma aperto a tutte le aziende che ne volessero far parte. Gia nel mese di aprile del 1999 il consorzio contava ben 600 membri. Nel mese di luglio dello stesso anno uscirono le prime specifiche tecniche del neonato Bluetooth. Da quel momento varie versioni del Bluetooth sono state ratificate dal SIG, tutte rispondenti ai requisiti di interoperabilità, armonizzazione della banda e promozione della

tecnologia. Obiettivo principale del SIG è appunto garantire il perfetto funzionamento di apparati Bluetooth costruiti da differenti.

1.2 Versioni

- *Bluetooth 1.0:* offriva una velocità di connessione di 1Mbps suddiviso tra dati e voce, se non fosse che soltanto circa 700Kbps vengono utilizzati per il trasferimento e come se non bastasse poteva comunicare con un solo dispositivo per volta.

- *Bluetooth 1.1:* fissati alcuni bug della versione precedente e permette la comunicazione su canali non cifrati.

- *Bluetooth 1.2:* viene adottata la Adaptive Frequency Hopping per rendere la comunicazione tra dispositivi più resistente alle interferenze esterne, e Enhanced Voice Processing per migliorare la qualità audio, soprattutto in ambienti rumorosi.

- *Bluetooth 2.0 + EDR:* la velocità di trasmissione dati passa a 2.1Mbps e scendono i consumi di energia.

- *Bluetooth 2.1 + EDR:* migliorata la velocità di associazione ed i passaggi necessari. Migliora ancora il consumo di energia, che varia in base all'utilizzo che si fa.

- *Bluetooth 3.0+HS, o Bluetooth High Speed Technology:* capace di raggiungere velocità di connessione che sfiorano i 24Mbps appoggiandosi sulla tecnologia WI-FI. La specifica finale è stata pubblicata nel mese di aprile del 2009.

1.3 Caratteristiche dei Dispositivi

La potenza massima di trasmissione dei dispositivi è di 100 metri ma per avere un risparmio energetico le case produttrici limitano la potenza dei dispositivi ad una trasmissione di 10 metri. Ogni dispositivo ha un indirizzo di 48 bit diviso in 3 parti con i seguenti compiti:

- per identificare la casa costruttrice

- per identificare il dispositivo

- per identificare la rete di cui fa parte, che prende il nome di "piconet", verrà spiegato in seguito come è composta e come viene creata.

Ogni dispositivo ha un clock a 28 bit che scatta 3200 volte al secondo ed è utilizzato per la sincronizzazione dei dispositivi e la generazione delle chiavi per l'autenticazione. Sono supportati due tipi di canali:

- Canali Sincroni

- Canali Asincroni

I canali Sincroni vengono utilizzati per la trasmissione della voce offrendo una trasmissione bilaterale con velocità di trasmissione massima di 64 Kb/s in tutte le versioni di Bluetooth.

Fig 1.1 – Connessione tra auricolare e uno Smartphone.

I canali Asincroni vengono utilizzati per la trasmissione dei dati e vengono usati in due modalità, simmetrica e asimmetrica. Ecco come si comporta lo standard nelle prime versioni cioè nella 1.0 e nella 1.1.

- Simmetrica, si può disporre di una velocità di trasmissione simmetrica massima di 433.9 Kb/s per direzione

Fig 1.2 – Canale asincrono con connessione simmetrica.

- Asimmetrica, si può disporre di una velocità di trasmissione simmetrica massima di 732.2 Kb/s in una direzione e 57.6 Kb/s nell'altra

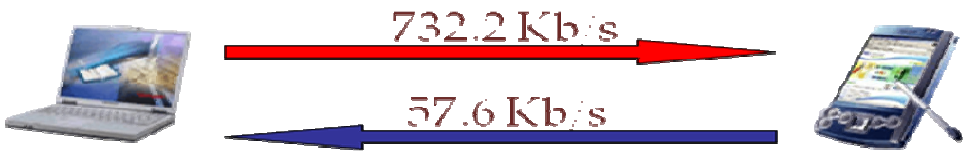

Fig 1.3– Canale asincrono con connessione asimmetrica.

Successivamente nella versione 2.0 e 2.1 si può arrivare a velocità di trasmissione pari a 3 Mb/s in entrambe le direzioni, mentre nella versione 3.0 HS si hanno i seguenti miglioramenti:

- Aumento nella velocità di trasferimento

Con la tecnologia Wi-Fi attuale, il Bluetooth 3.0 ha una velocità di trasferimento di 480 megabit (60 megabyte) al secondo per brevi distanze. Ad una distanza superiore a 10 metri, la velocità di trasferimento crolla a 100 megabit (12.5 megabyte) al secondo. Questa velocità di trasferimento dati è sufficiente per flussi audio e video ad alta definizione. Videocamere con tecnologia Bluetooth 3.0 saranno in grado di inviare contenuti in alta definizione a computer e televisioni con Bluetooth.

- Meno interferenze

Per evitare possibili interferenze, il Bluetooth 3.0 funzionerà nell'intervallo 6-9 GHz invece che 2,4 Ghz, come nei precedenti standard Bluetooth. Questo diminuisce la quantità di interferenze del segnale da altre tecnologie wireless, come ad esempio reti Wi-Fi e telefoni cordless. Bluetooth 3.0 sarà compatibile con i vecchi standard. Comunque, i dispositivi esistenti non saranno in grado di sfruttare la maggiore velocità che il Bluetooth 3.0 può offrire.

2. Protocolli

Nella figura sottostante viene fatto vedere lo stack dei protocolli Bluetooth.

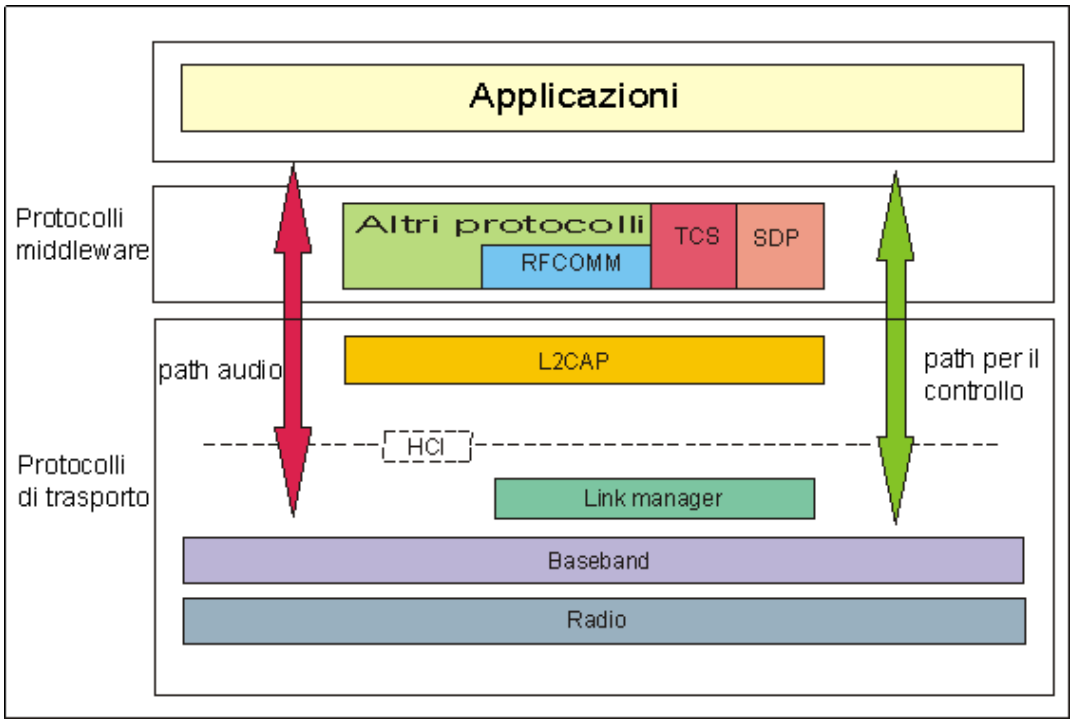

Fig 2.1 – Stack protocolli Bluetooth.

Esaminiamone ogni singola parte.

2.1 Radio Layer

La parte radio delle specifiche riguarda le caratteristiche dei transceivers e le specifiche di progettazione come l'accuratezza della frequenza, l'interferenza del canale e le caratteristiche di modulazione. Il sistema Bluetooth opera nella banda di frequenza ISM, globalmente disponibile, e la modulazione di frequenza è la GFSK. Supporta canali voce a 64 Kbps e canali dati asincroni con un tasso di picco di 24 Mbps nella versione 3.0. I canali dati sono asimmetrici (in una direzione) o simmetrici (in entrambe le direzioni). Il transceiver Bluetooth è un sistema FHSS operante su un insieme di *m* canali ognuno di 1 Mhz. Nella maggioranza dei paesi il valore di *m* è 79. Viene usato il frequency hopping e gli hop sono effettuati rapidamente sui possibili 79 hops nella banda, che inizia a 2.4 Ghz e termina a 2.480 Ghz. La scelta del frequency hopping è stata fatta per fornire protezione dalle interferenze. L'interfaccia aerea di Bluetooth è basata su una potenza d'antenna nominale di 0 dBm (1 mW) con estensioni per operare fino a 20 dBm (100 mW) in tutto il mondo. L'intervallo di collegamento nominale va da 10 centimetri a 10 metri, ma può essere esteso a più di 100 metri aumentando la potenza di trasmissione (usando l'opzione 20 dBm). Si può notare che una WLAN non può usare una potenza d'antenna minore di 0 dBm (1 mW) e dunque una soluzione 802.11 potrebbe non essere adottata per dispositivi vincolati dalla potenza.

2.2 Baseband Layer

Le funzioni chiave di questo layer sono la selezione della frequenza di hop, la creazione della connessione e il controllo di accesso al mezzo. La comunicazione Bluetooth prende luogo grazie alla creazione di una rete ad hoc chiamata *piconet*. L'indirizzo e il clock associato ad ogni dispositivo Bluetooth sono i due elementi fondamentali che governano la formazione di una piconet.
Ad ogni dispositivo è assegnato un singolo indirizzo a 48-bit che è simile all'indirizzo dei dispositivi LAN IEEE 802.xx. Il campo indirizzo è diviso in tre parti e la parte bassa (LAP) è usata in molte operazioni baseband come l'identificazione della piconet, il controllo degli errori, e i controlli di sicurezza. Le restanti due parti sono indirizzi proprietari delle organizzazioni produttrici. LAP è assegnato internamente da ogni organizzazione. Ogni dispositivo ha anche un clock a 28 bit (chiamato *clock nativo*) che scatta 3200 volte al secondo o una volta ogni 312.5 μs. Si noti che è il doppio dell'hopping rate di 1600 hops al secondo.

2.2.1 Piconet

Chi dà inizio alla formazione della rete assume il ruolo di *master* (della piconet). Tutti gli altri membri sono detti *slaves* della piconet. Ad ogni istante una piconet può avere fino a sette slaves attivi. A scopi d'identificazione, ad ogni slave attivo viene assegnato indirizzo di membro attivo, localmente univoco, AM_ADDR. Anche altri dispositivi potrebbero essere parte della piconet nella modalità parked. Un dispositivo Bluetooth non associato ad alcuna piconet è detto in modalità standby.

Fig 2.2 – Esempio di Piconet.

2.2.2 Scatternet

Quando un dispositivo appartiene a più reti si viene a creare una Scatternet.

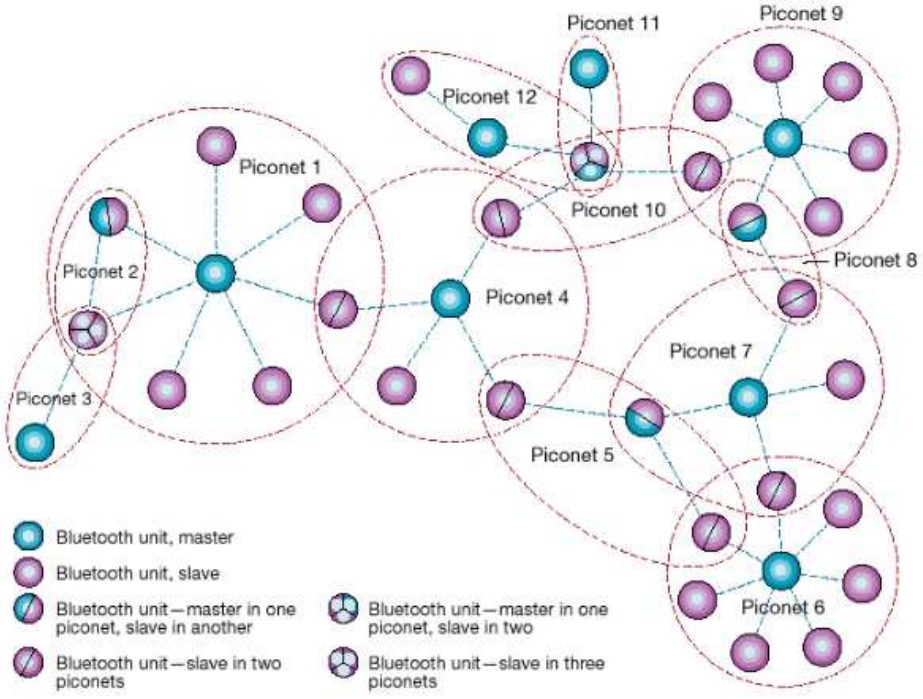

Fig 2.3 – Esempio di Scatternet.

Questo approccio può creare un gran numero di collisioni. Le piconet potrebbero sovrapporsi sia spazialmente che temporalmente, vale a dire che molte piconet potrebbero operare nella stessa area allo stesso tempo. Ogni piconet è caratterizzata da un master univoco e dunque le piconet eseguono gli hop in modo indipendente, ognuna con la propria sequenza di hopping di canale, determinata dal rispettivo master. In aggiunta i pacchetti trasportati sul canale sono preceduti da differenti channel access codes determinati dagli indirizzi dei dispositivi master. Nel momento in cui si aggiungono ulteriori piconet la probabilità di collisioni aumenta, e si verifica una degradazione delle performance, cosa che avviene comunemente nei sistemi FHSS. In questo scenario un dispositivo può partecipare a due o più piconet sovrapposte tramite il processo di condivisione del tempo. Per partecipare sul canale opportuno esso dovrebbe utilizzare l'indirizzo del dispositivo master associato ed il clock offset adeguato. Una unità Bluetooth può agire come slave in molte piconet, ma come master soltanto in una singola piconet. Un gruppo di piconet in cui esistono connessioni tra diverse piconet è detto *scatternet*.
Quando un dispositivo cambia ruolo e prende parte a differenti piconet è portato nella situazione in cui alcuni slot restano inutilizzati (per la sincronizzazione). Ciò implica che la completa utilizzazione della larghezza di banda non è ottenuta. Un'interessante proposta sarebbe quella di unire i timings dell'intera scatternet. Ma questo potrebbe portare ad un aumento della probabilità di collisione di pacchetti.
Un altro importante problema è il timing che un dispositivo perderebbe partecipando a più di una piconet. Un master assente da una piconet (essendo diventato slave momentaneamente in un'altra piconet) potrebbe mancare il polling degli slaves e deve assicurarsi di non perdere beacons dai suoi slaves. In maniera simile uno slave (diventando master o slave in un'altra piconet) assente da una piconet potrebbe far sembrare al master di essere uscito dall'intervallo o di essere connesso tramite un collegamento di scarsa qualità.

2.2.3 Stati Operativi

Inizialmente tutti i dispositivi si troveranno nella modalità standby. Qualche dispositivo (chiamato master) potrebbe iniziare la fase di inquiry (ricerca) e venire a conoscenza dei dispositivi nelle vicinanze e, se necessario, unirli nella propria piconet. Dopo la ricerca il dispositivo potrebbe formalmente associato tramite il paging, un processo di scambio di pacchetti tra il master ed uno slave per informare lo slave del clock del master. Se il dispositivo è stato già ricercato, il master potrebbe passare allo stato page bypassando lo stato inquiry. Una volta che il dispositivo finisce di essere paginato entra nello stato connected. Questo stato ha tre sottostati a preservazione di energia – hold, sniff, e park. Un dispositivo nello

stato connected può partecipare alla trasmissione dati.

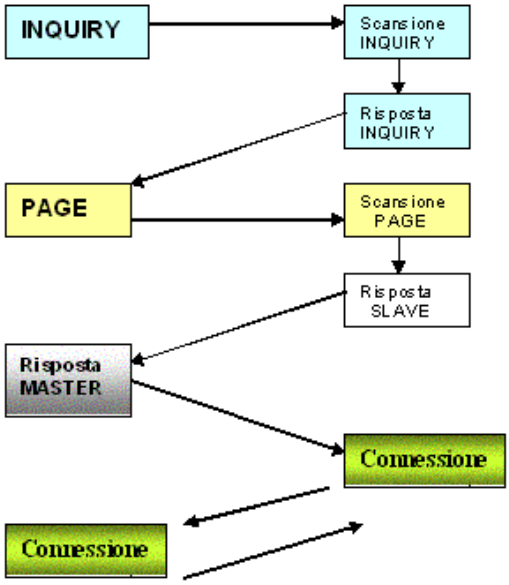

Fig 2.4 – Stati Operativi in Bluetooth.

Stato Inquiry:
Un dispositivo inizialmente nello stato standby entra nello stato di inquiry. Come suggerisce il nome, l'unico scopo di questo stato è collezionare informazioni sugli atri dispositivi Bluetooth nelle vicinanze. Tali informazioni includono l'indirizzo Bluetooth e il valore di clock, poiché queste formano il punto cruciale della comunicazione tra i dispositivi. Questo stato è classificato in tre sottostati: inquiry, inquiry scan ed inquiry response. Un potenziale master nello stato inquiry invia un pacchetto d'inquiry alla sequenza di hop di frequenze di inquiry. Questa sequenza è determinata dando un indirizzo comune come uno degli input dell'FSM. Un dispositivo (slave) che vuole essere scoperto entrerà periodicamente nello stato inquiry scan e ascolterà i pacchetti inquiry. Quando viene ricevuto un messaggio d'inquiry nello stato inquiry scan, deve essere inviato un pacchetto di risposta chiamato frequency hopping sequence (FHS), contenente l'indirizzo del dispositivo che ha risposto. I dispositivi rispondono dopo un jitter casuale per ridurre le possibilità di collisioni.

Stato Page:
Un dispositivo entra in questo stato per invitare altri dispositivi ad unirsi alla propria piconet. Un dispositivo potrebbe invitare soltanto dispositivi che esso

stesso conosce. Quindi normalmente l'operazione d'inquiry precederà tale stato. Anche questo stato è classificato in tre sottostati: page, page scan e page response.
Nella modalità page il master stima il clock dello slave basandosi sulle informazioni ricevute durante lo stato di inquiry, per determinare in quale punto della sequenza di hop lo slave dovrebbe rimanere in ascolto nella modalità page scan. Per tenere conto delle inesattezze nella stima il master trasmette il messaggio di page attraverso le frequenze immediatamente precedenti e successive a quella stimata. Ricevendo il messaggio di page lo slave entra nel sottostato page response. Restituisce un page response consistente nel suo ID packet che contiene il proprio device access code (DAC). Infine il master (dopo aver ricevuto il response dallo slave) entra nello stato page response ed informa lo slave del proprio clock ed indirizzo in modo che slave può proseguire e partecipare alla piconet. Lo slave ora calcola un offset per sincronizzarsi con il clock del master, e lo utilizza per determinare la sequenza di hopping per la comunicazione nella piconet. È evidente (in ogni comunicazione wireless) che, affinchè la comunicazione abbia luogo, il trasmittente ed il ricevitore dovrebbero utilizzare la stessa frequenza. In ogni dispositivo è presente un frequency selection module (FSM) per selezionare la prossima frequenza da utilizzare sotto varie circostanze. Nello stato connected il clock e l'indirizzo del dispositivo (master) determinano completamente la sequenza di hopping. Diverse combinazioni di input (clock, indirizzo) sono usate a seconda dello stato operativo. Durante l'operazione di inquiry l'indirizzo in input all'FSM è un indirizzo di inquiry comune. Questo indirizzo comune è necessario perché al momento della ricerca nessun dispositivo ha informazioni sulla sequenza di hopping da seguire. L'indirizzo del dispositivo paginato è fornito in input all'FSM per lo stato di paging.

2.2.4 Canale Fisico

Il canale è suddiviso in slots di tempo, ognuno della durata di 625 μs. Gli slots di tempo sono numerati secondo il clock Bluetooth del master della piconet. Viene utilizzato uno schema time division duplex (TDD) in cui il master e lo slave trasmettono alternativamente. Il master inizia la trasmissione soltanto negli slots di tempo aventi numero pari, e lo slave inizia la trasmissione soltanto negli slots di tempo aventi numero dispari. L'inizio del pacchetto sarà allineato all'inizio dello slot. Un dispositivo Bluetooth determina la parità dello slot osservando il bit meno significativo (LSB) nella rappresentazione binaria del suo clock. Se l'LSB è pari a 1 esso è il possibile slot di trasmissione soltanto per lo slave. In circostanze normali ad uno slave è consentito di trasmettere soltanto se nello slot precedente ha ricevuto un pacchetto dal master. Uno slave dovrebbe conoscere il clock e

l'indirizzo del master per determinare la frequenza successiva (dall'FSM). Questa informazione è scambiata durante il paging.

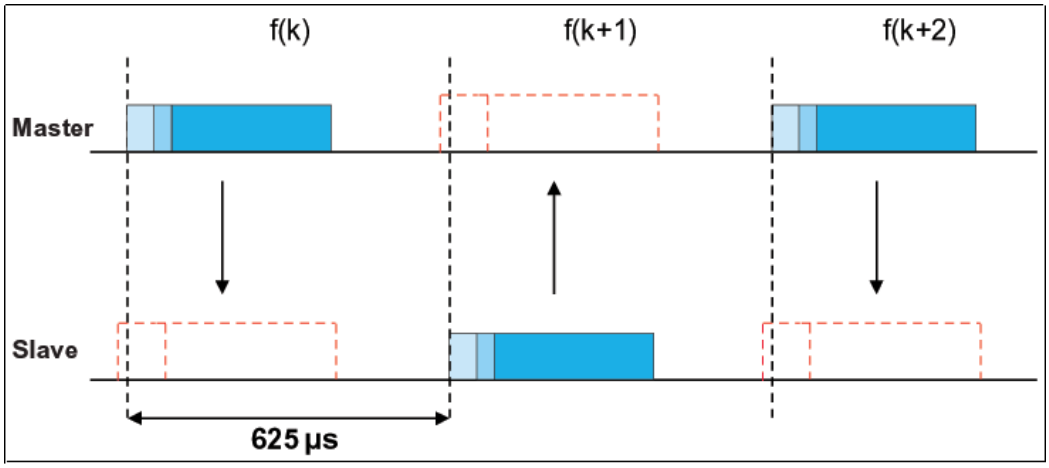

Fig 2.5 – Comunicazione Packet Based.

Bluetooth utilizza una comunicazione packet-based in cui i dati da trasmettere sono frammentati in pacchetti. Soltanto un singolo pacchetto può essere trasmesso in ogni slot. Un tipico pacchetto utilizzato in queste comunicazioni ha tre componenti: access code, header, payload. Il componente principale dell'access code è l'indirizzo del master della piconet. Tutti i pacchetti scambiati sul canale sono identificati dall'identità del master. Il pacchetto verrà accettato dal destinatario soltanto se l'access code combacia con il corrispondente access code del master della piconet. Ciò aiuta anche la risoluzione di conflitti nel caso in cui due piconets stanno operando concorrentemente sulla stessa frequenza. Uno slave che riceve due pacchetti nello stesso slot può identificare il suo pacchetto esaminando l'access code. L'header del pacchetto contiene molti campi, come un active slave address a 3 bit, un ACK/NACK ad 1 bit per lo schema ARQ [Automatic Repeat reQuest – ogni volta che un errore è riconosciuto, viene restituito un acknowledgment negativo (NACK) e i frame specificati sono ritrasmessi], un packet type a 4 bit per distinguere i tipi di payload, e un header error check code a 8 bit per riconoscere errori nell'header. A seconda della dimensione del payload potrebbero essere utilizzati uno, tre, o cinque slots per la trasmissione del pacchetto. La frequenza di hop utilizzata per il primo slot è usata per il resto del pacchetto. Quando si trasmettono pacchetti in slot multipli è importante che le frequenze utilizzate negli slots di tempo successivi siano quelle assegnate a tali slot, e che non seguano la sequenza di frequenza che avrebbe dovuto essere applicata normalmente. Quando un dispositivo utilizza cinque slots per la trasmissione di un pacchetto, la prossima trasmissione di pacchetto è consentita in F(k+6) e non in F(k+2). Si noti inoltre che il time slot del ricevitore diventa F(k+5) invece che F(k+1). Su questo canale suddiviso

in slot, sono supportati sia collegamenti sincroni che asincroni. Tra un master ed uno slave c'è soltanto un asynchronous connectionless link (ACL) supportato. Questo è il link di default esistente una volta che si stabilisce una connessione tra un master ed uno slave. Ogni volta che un master vorrà comunicare, lo farà, e lo slave risponderà. Opzionalmente una piconet potrebbe anche supportare collegamenti synchronous connection oriented (SCO). Il collegamento SCO è simmetrico tra master e slave con larghezza di banda riservata e regolare scambio periodico di dati sottoforma di slots riservati. Questi collegamenti sono essenziali ed utili per informazione ad alta priorità e con limiti di tempo come audio e video.

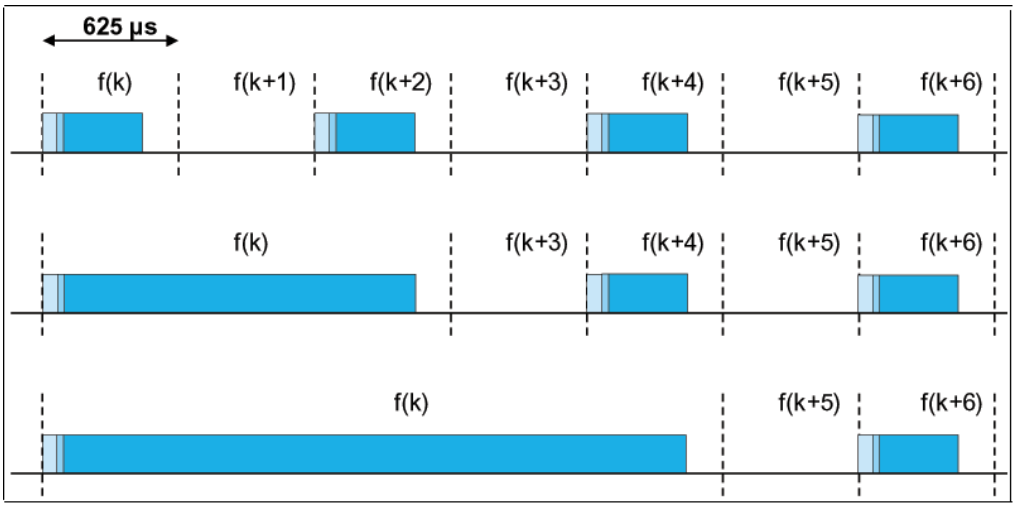

Fig 2.6 – Pacchetti multislot.

2.3 Link Manager Layer

Il link manager protocol (LMP) è responsabile dell'impostazione e del mantenimento del collegamento Bluetooth. Al momento le principali funzionalità di questo layer sono la gestione dell'energia e la gestione della sicurezza. Fornisce inoltre un minimo supporto QoS permettendo il controllo di parametri come delay e delay jitter. Normalmente un dispositivo di paging è il master della piconet, ma, a seconda degli scenari d'uso, i ruoli del master e dello slave potrebbero essere interscambiati, e ciò è coordinato tramite lo scambio di pacchetti LMP.

2.3.1 Gestione dell'energia

Le unità Bluetooth possono trovarsi in diverse modalità operative durante lo stato di connessione, ossia active mode, sniff mode, hold mode e park mode. Queste modalità sono discritte di seguito.

- **Active mode:** in questa modalità l'unità Bluetooth partecipa attivamente all'interno della piconet. Sono previste varie ottimizzazioni per risparmiare energia. Ad esempio se il master informa lo slave nel momento in cui verrà indirizzato, lo slave potrebbe restare in sleep fino a quel momento. Gli slave attivi sono sottoposti a polling dal master per le trasmissioni.

- **Sniff mode:** questa è una modalità a basso consumo in cui l'attività di listening è ridotta. L'LMP nel master invia un comando allo slave affinchè entri nella modalità sniff, fornendogli un intervallo di sniff, e lo slave ascolta le trasmissioni soltanto in tali intervalli fissi.

- **Hold mode:** in questa modalità lo slave temporaneamente non supporta i pacchetti ACL sul canale (eventuali collegamenti SCO saranno ancora supportati). In questa modalità la capacità è resa disponibile per effettuare altre funzioni, come scanning, paging, inquiring, o partecipazione ad altre piconet.

- Park mode: questa è una modalità a consumo molto basso. Lo slave cede il proprio active member address e riceve un parked member address a 8 bit. Lo slave ciononostante resta sincronizzato con il canale. Tutti i messaggi da inviare ad un membro parked sono spedite sul canale di broadcast caratterizzato da un active member address di tutti zero. Oltre a risparmiare energia la modalità park aiuta il master a gestire più di sette slave (limitati dallo spazio d'indirizzo dell'active member address a 3 bit) nella piconet.

2.4 Host Controller Interface Layer

Questo è il layer d'interfaccia opzionale fornito tra i layer superiori (sopra LMP) ed inferiori dello stack di protocolli Bluetooth, per l'accesso alle capacità hardware di Bluetooth. Questo layer è necessario ogni qualvolta i layer superiori sono implementati sulla scheda madre del dispositivo host. Un approccio di questo tipo apporterebbe benefici su come le risorse libere del sistema potrebbero essere impiegate. La specifica definisce dettagli come i differenti tipi di pacchetto trattati da questo layer. I pacchetti di comando che sono utilizzati dall'host per controllare il dispositivo, i pacchetti di evento che sono utilizzati per informare l'host dei cambiamenti, e i pacchetti di dati cadono in questa categoria.

2.5 Logical Link Control and Adaptation Protocol Layer

Questo è il protocollo con il quale molte applicazioni interagiranno a meno che venga utilizzato un host controller. L2CAP supporta il multiplexing di protocollo per fornire astrazione ad ognuna delle molte applicazioni in esecuzione nei layers superiori come se fossero in esecuzione da sole. Poiché i pacchetti di dati definiti dal protocollo baseband sono limitati nelle dimensioni, L2CAP segmenta anche i pacchetti grandi provenienti dai layer superiori come RFCOMM o SDP in pacchetti multipli più piccoli prima della loro trasmissione sul canale. In maniera simile più pacchetti baseband ricevuti potrebbero essere riassemblati in un unico grande pacchetto L2CAP. Questo protocollo fornisce QoS su alcuni parametri come larghezza di banda di picco, latenza, e variazione di ritardo quando viene stabilito un collegamento tra due unità Bluetooth.

2.6 Protocolli Middleware

La funzionalità di base del middleware protocol group è quella di presentare agli application layers un'interfaccia standard che potrebbe essere utilizzata per la comunicazione al di sopra del layer di trasporto, sarebbe a dire che le applicazioni non hanno bisogno di sapere le complessità del layer di trasporto, ma devono soltanto utilizzare le application program interfaces (APIs) o funzioni di livello superiore fornite dai protocolli di middleware. Questo gruppo consiste nel layer RFCOMM, service discovery protocol (SDP), IrDA interoperability protocols, telephony control specification (TCS), e audio. Il layer RFCOMM presenta una porta seriale virtuale alle applicazioni che utilizzano l'interfaccia seriale. Ogni applicazione che sta usando la porta seriale può lavorare ininterrottamente sui dispositivi Bluetooth. RFCOMM utilizza una connessione L2CAP per instaurare un collegamento tra due dispositivi. Nel caso dei dispositivi Bluetooth non ci sarà nessun dispositivo statico e pertanto i servizi offerti dagli altri dispositivi devono essere scoperti. Ciò è effettutato utilizzando il service discovery protocol (SDP) dello stack di protocolli Bluetooth. La scoperta dei servizi rende il dispositivo autoconfigurante senza necessità di intervento manuale. L'IrDA interoperability protocol non serve per la comunicazione tra dispositivi Bluetooth ed Infrarossi. È utilizzato soltanto per permettere alle applicazioni IrDA di funzionare sui dispositivi Bluetooth senza alcuna modifica. I protocolli principali nell'insieme IrDA sono IrOBEX (IrDA object exchange) per lo scambio di oggetti tra due dispositivi ed IrMC (infrared mobile communications) per la sincronizzazione. L'audio è la parte rilevante di Bluetooth. All'audio è data la massima priorità ed è inviato direttamente sulla baseband a 64 Kbps in modo da fornire una qualità della voce molto buona. Un altro importante punto da notare è che l'audio non è attualmente un layer dello stack di protocolli, ma soltanto un formato di pacchetto

specifico che può essere trasmesso direttamente sui collegamenti SCO del layer baseband. Il controllo telefonico è implementato utilizzando il protocollo telephony control specification – binary (TCS-BIN). TCS definisce tre tipi principiali di aree funzionali: call control, group management, e connectionless TCS. Il call control è utilizzato per impostare chiamate che possono essere utilizzate sequenzialmente per trasportare traffico voce e dati. TCS opera sia configurazione point-to-point che point-to-multipoint. Uno dei concetti principali di TCS è quello del wireless user group (WUG). Il group management abilita estensioni telefoniche multiple, inoltro di chiamate, e gruppi di chiamate. Si considerino ad esempio headsets multipli ed un singolo base set. Quando arriva una chiamata nel base set, tutti gli headsets possono riceverla. In maniera simile le chiamate possono anche essere inoltrate. Le funzionalità di TCS includono *configuration distribution* e *fast intermember access*. La configuration distribution è il meccanismo utilizzato per trovare informazioni sugli altri membri in un gruppo. Il fast intermember access è un metodo con cui due slave creano una nuova piconet. Un membro WUG usa le informazioni ottenute dal configuration distribution e determina un altro membro che vuole contattare. In seguito invia le informazioni del dispositivo al master, che le inoltra a tale dispositivo. Poi il dispositivo contattato risponde con le proprie informazioni di device address e clock e si pone nello stato di page scan. A questo punto il master contatta il dispositivo che inizia la comunicazione. Questo dispositivo pagina il dispositivo contattato e forma una nuova piconet. Ciò mostra come viene formata una nuova piconet tra due slave con l'aiuto del master. In tutti i casi sopra esposti viene stabilito un canale connection-oriented. Per scambiare semplici informazioni come regolazione del volume o informazioni di segnalazione l'instaurazione di un canale di questo tipo è distruttivo e pertanto è previsto un TCS connectionless per utilizzare un canale connectionless.

3. Profili

Per facilitare l'utilizzo dei dispositivi Bluetooth sono stati definiti una serie di profili, questi profili identificano una serie di possibili applicazioni. Questi profili sono stati sviluppati per promuovere l'interoperabilità tra le varie implementazioni dello stack di protocolli Bluetooth. Ogni specifica di profilo Bluetooth è stata definita per fornire uno standard chiaro e trasparente che può essere utilizzato per implementare una specifica funzione utente. Due dispositivi Bluetooth possono ottenere una funzionalità comune solo se entrambi i dispositivi supportano profili identici. Ad esempio un telefono cellulare ed un headset devono supportare entrambi il profilo headset Bluetooth affinchè l'headset funzioni con il telefono. I profili Bluetooth derivano dai modelli d'uso. In tutto sono stati elencati 13 profili

che possono essere classificati nelle seguenti quattro categorie:

- **General profiles**: il profilo di accesso Generico, che non è una reale applicazione, fornisce una via per stabilire e mantenere collegamenti efficienti tra il master e gli slave. Il profilo di scoperta dei servizi abilita gli utenti ad accedere all'SDP per scoprire quali applicazioni (servizi Bluetooth) sono supportate da uno specifico dispositivo.

- **Telephony profiles:** il profilo di telefonia cordless è progettato per i telefoni tre-in-uno. Il profilo Intercom supporta la comunicazione vocale a due vie tra due dispositivi Bluetooth ognuno all'interno del range dell'altro. Il profilo Headset specifica come Bluetooth può fornire una connessione wireless con un headset (con auricolari/microfoni) per l'utilizzo con un computer o un telefono cellulare.

Fig 3.1 – Telephony profiles.

- **Network profiles:** il profilo LAN Access abilita i dispositivi Bluetooth sia a connettersi ad una LAN attraverso APs che a formare una piccola LAN wireless tra di loro. Il profilo dial-up networking è progettato per fornire connessioni dial-up attraverso telefoni cellulari abilitati Bluetooth.

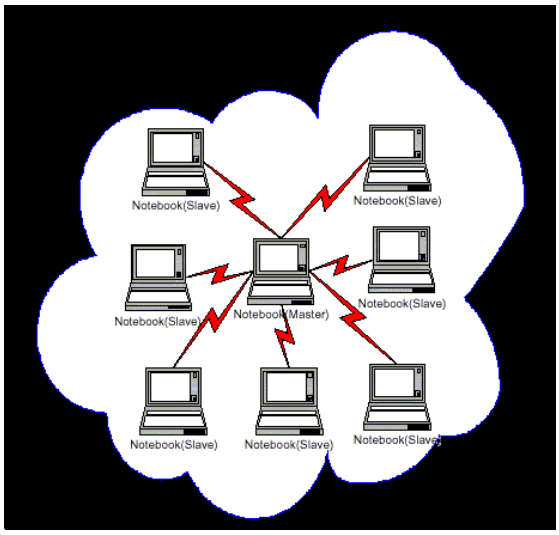

Fig 3.2 – Network profiles.

- **Serial and object exchange profiles:** il profilo porta seriale emula una linea seriale (porte seriali RS232 e USB) per applicazioni (legacy) che richiedono una linea seriale. Gli altri profili, generic object exchange, object push, file transfer, e synchronization servono per scambiare oggetti tra due dispositivi wireless.

Bluetooth è l'unica tecnologia wireless ad aver tentato di far seguire a tutti i dispositivi elettronici domestici un particolare paradigma comune. Ciò ha avuto parzialmente successo, ma ha le sue limitazioni. La comunicazione Bluetooth non supporta attualmente il routing. Sono in corso alcuni sforzi di ricerca per sistemare questo aspetto nelle specifiche Bluetooth. Una volta fornito il supporto al routing, la comunicazione inter-piconet potrebbe essere migliorata. Anche i problemi degli handoffs non sono stati finora trattati. Nonostante l'architettura master-slave abbia aiutato a mantenere bassi i costi, il master diventa il collo di bottiglia per l'intera piconet in termini di prestazioni, tolleranza agli errori ed utilizzazione di banda. Cosa ancora più importante, la comunicazione Bluetooth avviene nella stessa banda di frequenza della WLAN e pertanto devono essere sviluppate soluzioni di coesistenza per evitare interferenze. La tecnologia è ancora in fase di sviluppo. Ci sono attualmente all'incirca 1800 compagnie che stanno contribuendo allo sviluppo della tecnologia.

4. Sicurezza

4.1 Sicurezza in Bluetooth

In generale possiamo dire che per garantire sicurezza in una trasmissione dati, devono essere assicurati alcuni servizi [security services]:

- **Confidenzialità**: i dati scambiati non devono essere letti da utenti non autorizzati, in altre parole, nessun membro esterno deve poter accedere al contenuto della trasmissione.

- **Autenticazione**: il messaggio che riceve B deve effettivamente provenire dall'unità che B ritiene essere il mittente (A): il messaggio deve essere autentico.

- **Integrità dei dati**: i dati ricevuti non devono essere danneggiati o corrotti.

- **Autorizzazione**: il dispositivo che invia i dati deve essere autorizzato.

Se questi servizi vengono garantiti, la connessione può essere ritenuta sicura. Tuttavia, durante la trasmissione di informazioni con questo metodo, è possibile essere esposti ad attacchi in linea. Di pari passo con la diffusione dei dispositivi Bluetooth, aumenta anche l'interesse dei cybercriminali. La specifica Bluetooth prevede tre livelli di sicurezza che devono essere implementati nei dispositivi.

Mode 1: nessuna sicurezza.

Mode 2: procedure di protezione a livello di servizio/applicazione.

Mode 3: procedure di protezione a livello di dispositivo; usando questa modalità è possibile far comunicare due o più dispositivi solo se questi sono già stati associati e quindi sono considerati "fidati" .

La sicurezza dei dispositivi è garantita da cinque elementi:

- **Indirizzo BT_ADDR**: L'indirizzo fisico del singolo dispositivo; ogni indirizzo è unico poichè assegnato dall' IEEE e collegabile al produttore che ha realizzato l'apparecchio. L'indirizzo è un campo di 48 bit (esempio: 00:0A:D9:62:12:36) di cui i primi 24 bit sono legati alla casa produttrice (in questo caso Sony Ericsson). L'idea di univocità della periferica è identica al caso delle schede di rete che hanno il cosidetto MAC Address; come ben sappiamo però questo indirizzo è facilmente falsificabile (tramite spoofing). Anche per la tecnologia

Bluetooth è possibile, su alcuni dispositivi, modificare questo valore cambiando "l'identità" del dispositivo.

- **Chiave di cifratura** (8-128 bit) utilizzando l'algoritmo SAFER+
- **Chiave di collegamento** (128 bit)
- **Numeri pseudocasuali** (128 bit)
- **Vari algoritmi** per la generazione delle chiavi (E0, E21, E22, ecc)

La sicurezza in Bluetooth viene gestita in tre fasi che vengono di seguito descritte.

4.1.1 Fase 1

In questa fase i dispositivi al momento della loro prima accensione creano la loro Unit Key, che verrà mantenuta in una loro memoria non volatile.

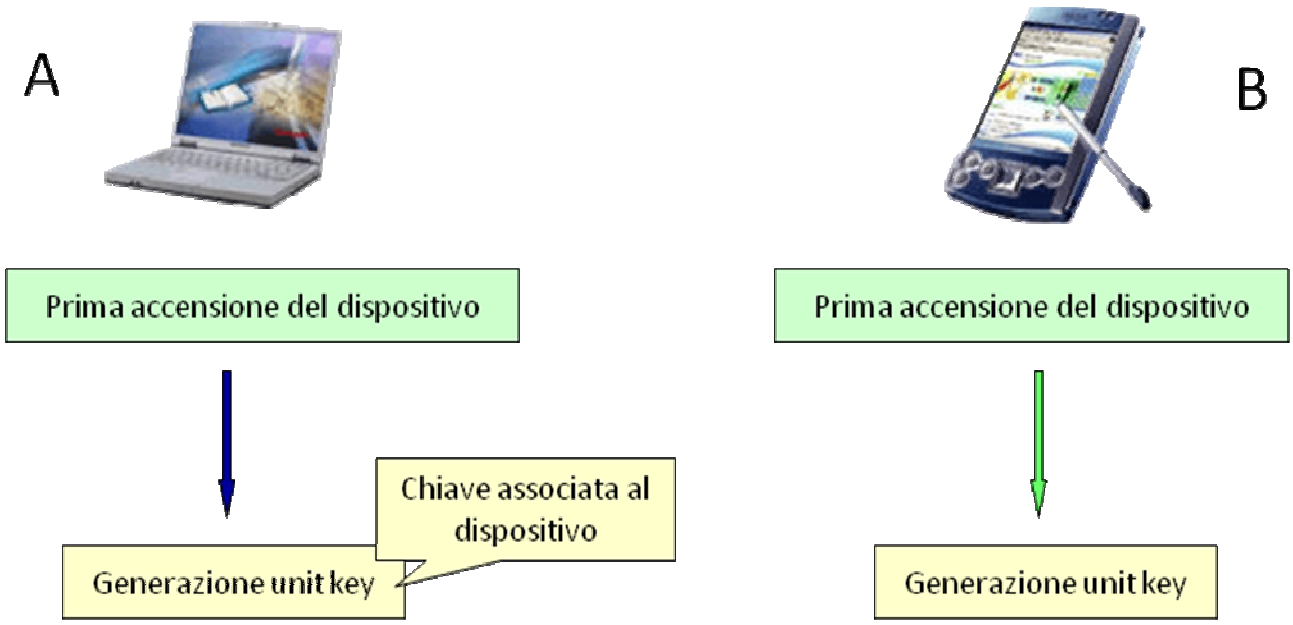

Fig 4.1 – Fase 1.

Questa chiave verrà successivamente utilizzata per la fase di autenticazione e può essere utilizzata come link key tra due dispositivi.

4.1.2 Fase 2

La seconda fase si verifica quando due dispositivi entrano in contatto per la prima volta, viene generata una Initialization Key, la quale viene utilizzata per l'autenticazione, successivamente viene scambiata la Link Key.

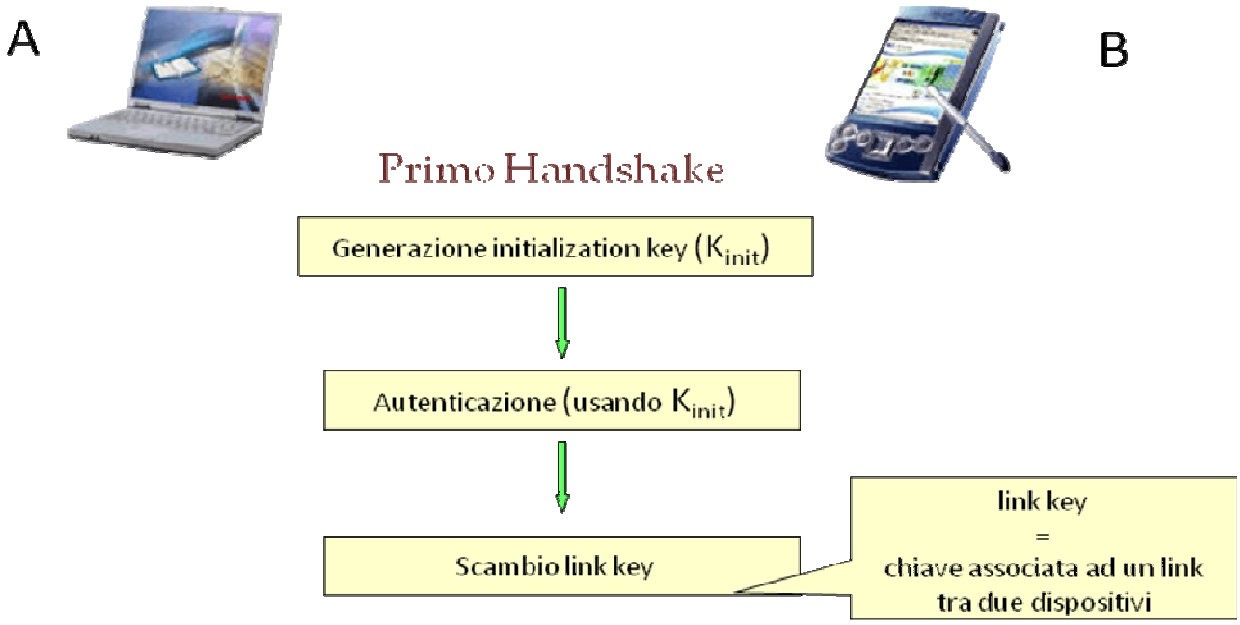

Fig 4.2 – Fase 2.

4.1.3 Fase 3

L' ultima fase si ha per gli handshake successivi quando due dispositivi sono entrati in contatto già in passato ed ora vogliono effettuare di nuovo una connessione e scambiarsi anche una Encryption Key.

Fig 4.3 – Fase 3.

4.1.4 Tipi di Key

Una link key è una chiave associata ad un link esistente tra due dispositivi, può essere:

- Una initialization key, creata in fase di inizializzazione
- Una link key semi-permanente
- Una link key temporanea

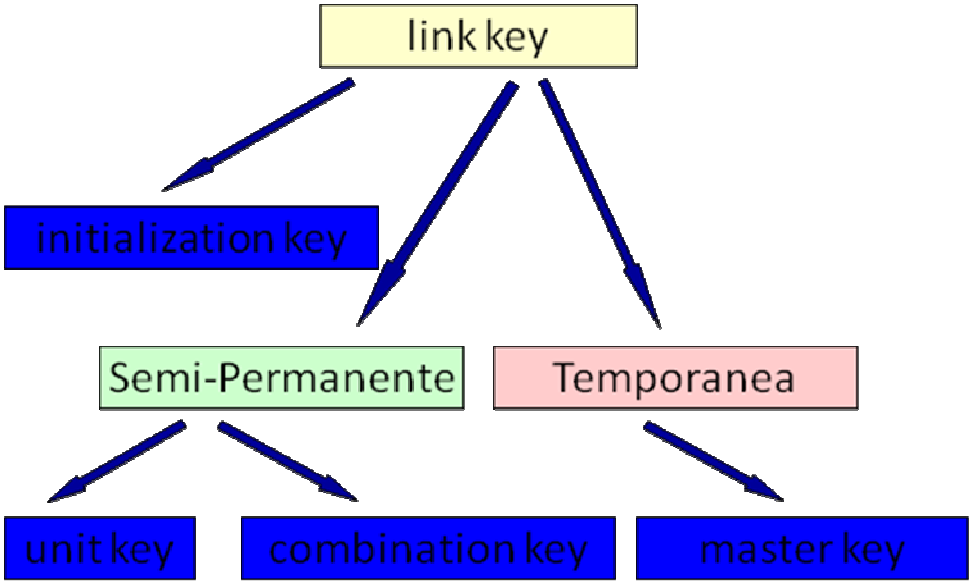

Fig 4.4 – Tipi di Link Key.

1. Una initialization key, K_{init}, è creata quando due dispositivi entrano in contatto per la prima volta.

2. Una unit key, K_A, è generata da un dispositivo nell'istante in cui è acceso per la prima volta.

3. Una combination key, K_{AB}, è ottenuta da informazioni prodotte da una coppia di dispositivi, A e B.

4. Una master key, K_{master}, è usata dal dispositivo master quando vuole trasmettere contemporaneamente a più dispositivi slave.

4.2 Generazione della Unit Key

Una Unit Key viene generata tramite l'algoritmo E21 quando un dispositivo si avvia per la prima volta. L' algoritmo E21 prende in input:

- BD_ADDR (indirizzo Bluetooth del dispositivo)
- RAND (numero casuale di 128 bit)

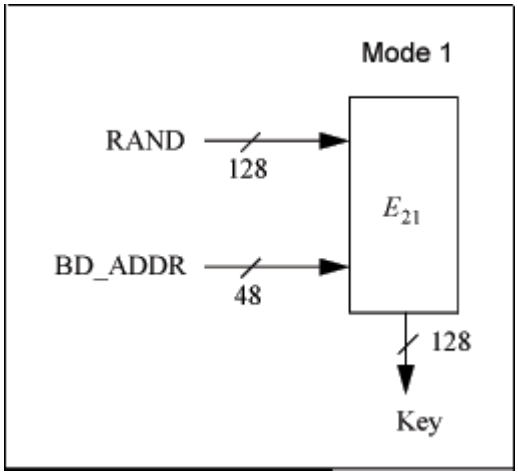

Fig 4.5 – Algoritmo per la generazione della Unit Key.

4.3 Generazione della Initialization Key

Una Initialization Key viene generata tramite l'algoritmo E22 quando due dispositivi entrano in contatto per la prima volta, uno dei due il Richiedente (B) prova a raggiungere il Verificatore (A). Il Richiedente deve dimostrare al Verificatore di essere un dispositivo autorizzato ovvero di condividere lo stesso PIN. Il PIN è un Personal Identification Number, nei dispositivi Bluetooth può variare tra 1 e 16 bytes. Le tipiche 4 cifre spesso usate per i codici PIN sono sufficienti per una situazione a basso rischio, mentre per avere un alto livello di sicurezza può deve essere usato un PIN più lungo.

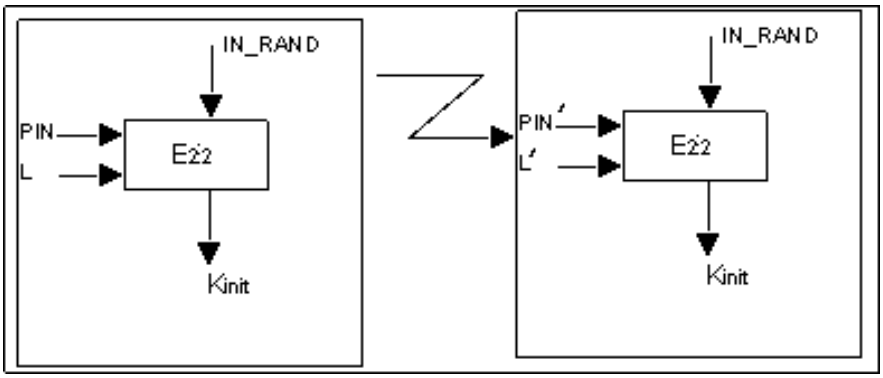

Fig 4.6 – Algoritmo per la generazione della Initialization Key.

Viene prima di tutto generata una initialization key in entrambi i dispositivi, sulla base di questo PIN, e poi è avviata la fase di autenticazione in cui A accerta che B condivide con se una stessa link key, che in questa fase coincide con l'initialization key appena generata.

4.4 Autenticazione

Per la fase di pairing viene usata la link key K_{AB} decisa dai due dispositivi. Il zrocesso di autenticazione si basa su uno schema di challenge-response in cui un dispositivo verificatore accerta che uno richiedente condivida con se una certa chiave segreta. Entrambi i dispositivi basano l'autenticazione sull'algoritmo E1, che ritorna come risultato i valori SRES e ACO (Authenticated Ciphering Offset) usato successivamente per la cifratura dei pacchetti. L'algoritmo E1 prende in input:

- un numero casuale AU_RAND_A prodotto dal dispositivo verificatore A;
- l'indirizzo Bluetooth del dispositivo B, che richiede di essere autenticato;
- l'attuale link key.

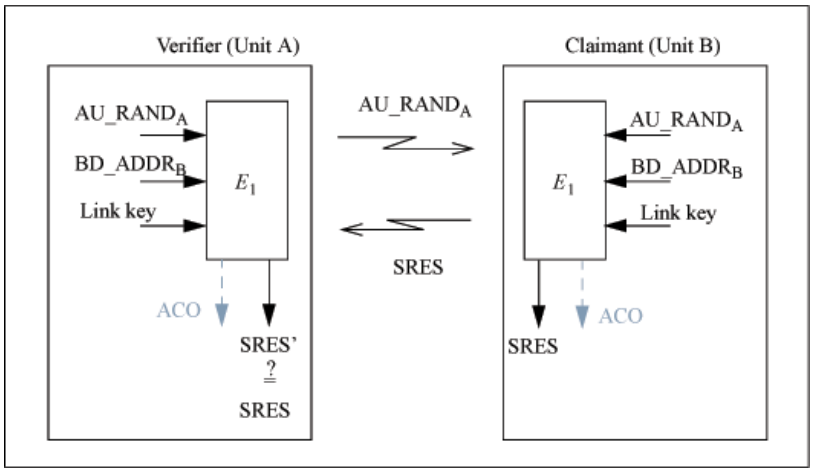

Fig 4.7 – Scambio di informazioni nella fase di autenticazione.

4.5 Scambio della Link Key

La chiave associata al link tra due dispositivi dipende dal grado di sicurezza richiesto e dalle capacità di memoria dei dispositivi, può essere:

- La Unit Key di uno di essi;
- Una Combination Key ottenuta da informazioni prodotte da entrambi i dispositivi

Quando viene utilizzata come Link Key la Unit Key di uno dei due dispositivi il grado di sicurezza che si è richiesto non è molto alto, vediamo un esempio di scambio della Unit Key:

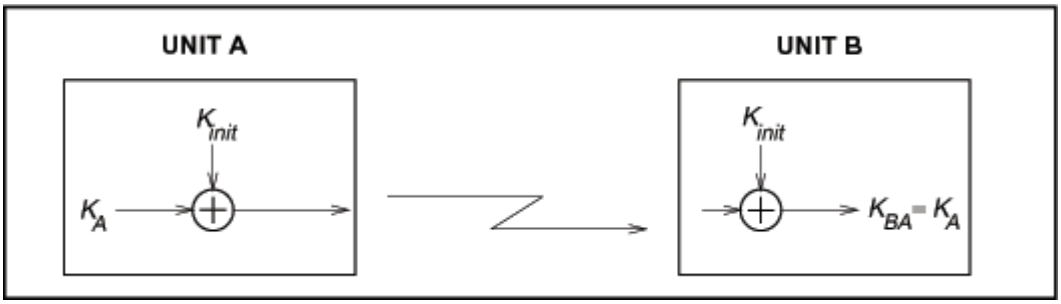

Fig 4.8 – Scambio della Unit Key.

Caso inverso lo si ha quando si richiede un livello di sicurezza alto, infatti viene utilizzata come Link Key una Combination Key, nella figura sottostante viene mostrato il meccanismo di costruzione e di scambio di questa chiave:

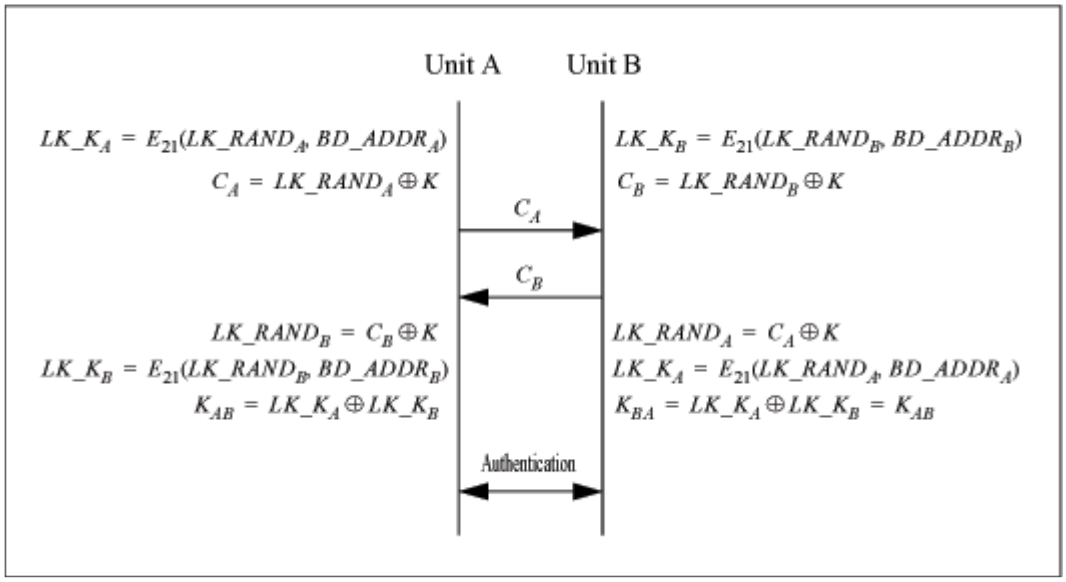

Fig 4.9 – Scambio della Combination Key.

Entrambi i dispositivi basano l'autenticazione sull'algoritmo E21.

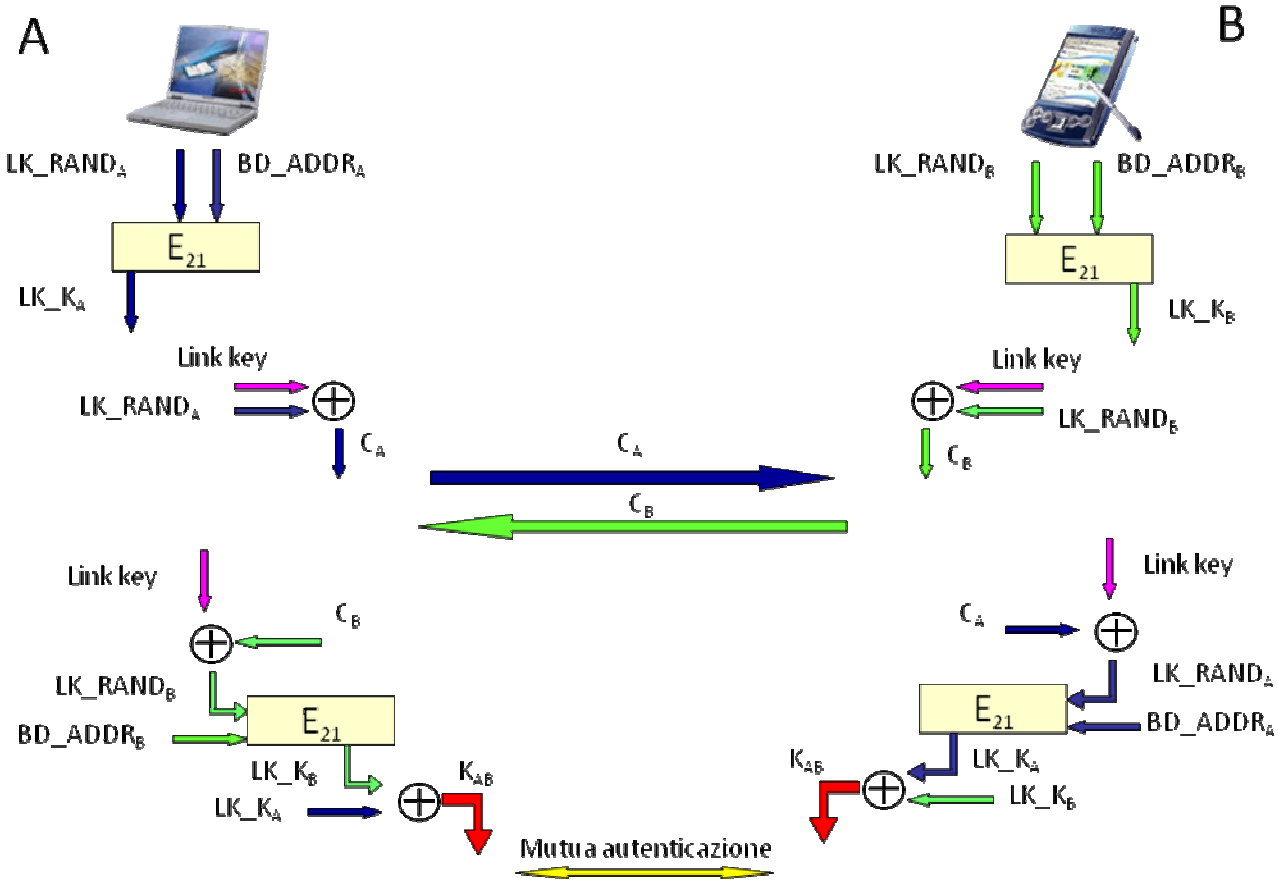

Fig 4.10 – Mutua Autenticazione tramite Combination Key.

4.6 Generazione della Master Key

Una master key è una chiave temporanea utilizzata per rimpiazzare la corrente link key, quando il master della piconet vuole trasmettere simultaneamente la stessa informazione a più dispositivi riceventi. Per prima cosa il master crea una link key da due numeri casuali, RAND1 e RAND2, a 128 bit applicando l'algoritmo E_{22}.

$$K_{master} = E_{22} (RAND1, RAND2, 16)$$

Creata la chiave K_{master}, il dispositivo master trasmette allo slave un terzo numero casuale, detto RAND. Utilizzando l'algoritmo E_{22} con la corrente link key e RAND come input, sia il master che lo slave calcolano un valore, detto overlay, OVL, a 128 bit. Il master invia lo XOR bit a bit dell'overlay e la nuova link key, K_{master}, allo slave. Lo slave conoscendo l'overlay, può così ricalcolare K_{master} effettuando lo XOR bit a bit del numero ricevuto e l'overlay calcolato. Per confermare il successo di questa transazione, le unità eseguono una mutua autenticazione usando la nuova link key. La procedura è ripetuta dal master per ogni slave che riceverà così la nuova link key.

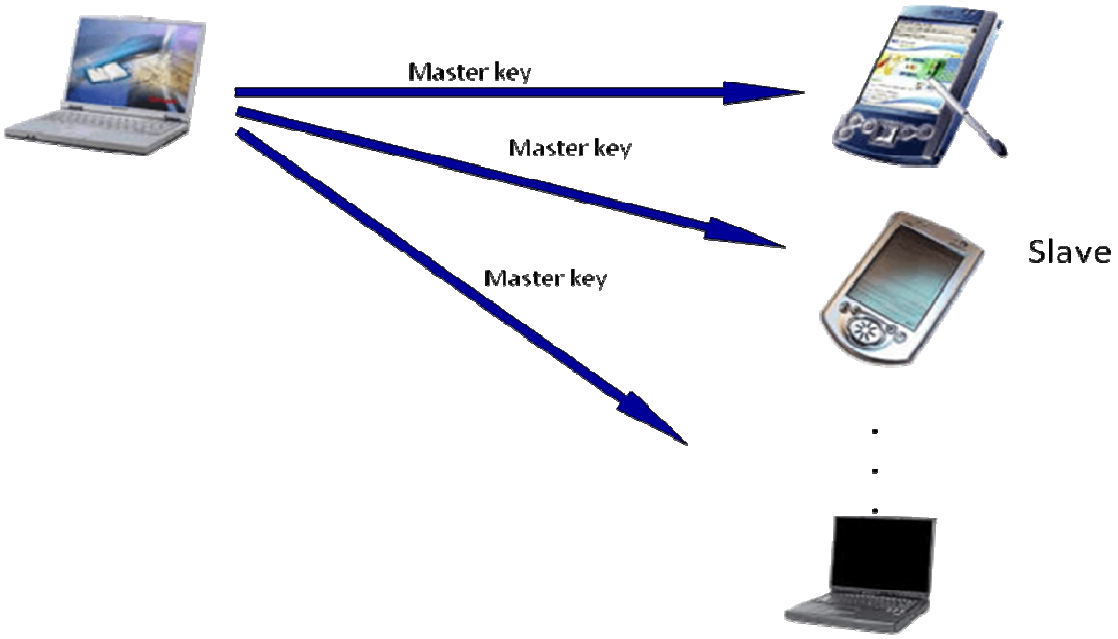

Fig 4.11 – Scambio della Master Key.

Esempio di generazione di una Master Key:

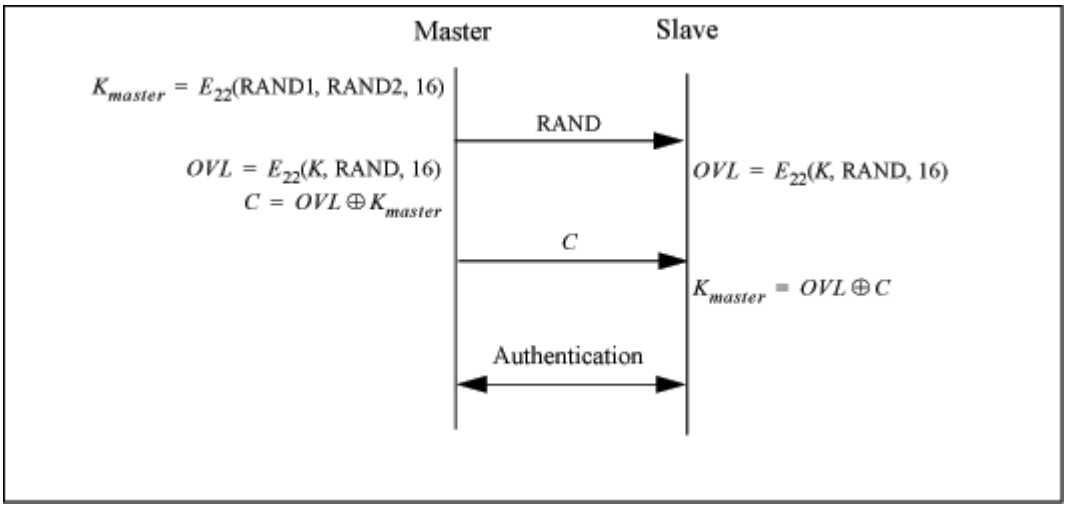

Fig 4.12 – Generazione della Master Key.

4.7 Encryption

In Bluetooth, le informazioni utente sono protette cifrando prima della trasmissione il campo payload dei pacchetti, ovvero il campo che contiene le vere e proprie informazioni da trasmettere. Come prima cosa viene creata una Encryption Key utilizzando l'algoritmo E3.

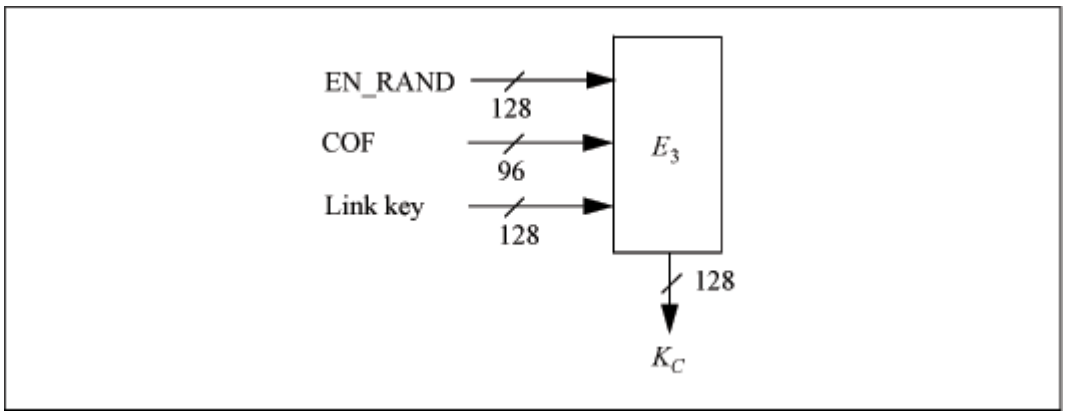

Fig 4.13 – Algoritmo per la generazione della Encryption Key.

Il COF può assumere due valori se la link key attuale è una master key allora si utilizza la concatenazione dell' indirizzo del master con se stesso (BD_ADDR concatenato a BD_ADDR), mentre se la link key non è una master key allora si usa il valore di ACO.

La cifratura è ottenuta per mezzo dello Stream Cipher E0, che viene nuovamente sincronizzato per ogni nuovo payload trasmesso.
Lo Stream Cipher consiste sostanzialmente di 3 parti:

- La prima esegue l'inizializzazione, ovvero genera la chiave per il payload (tramite l'algoritmo E0)
- La seconda genera i bit del key stream
- La terza esegue la cifratura o la decifratura delle informazioni trasmesse come XOR bit a bit tra ogni bit del testo in chiaro/cifrato e ogni bit generato dal key stream generator

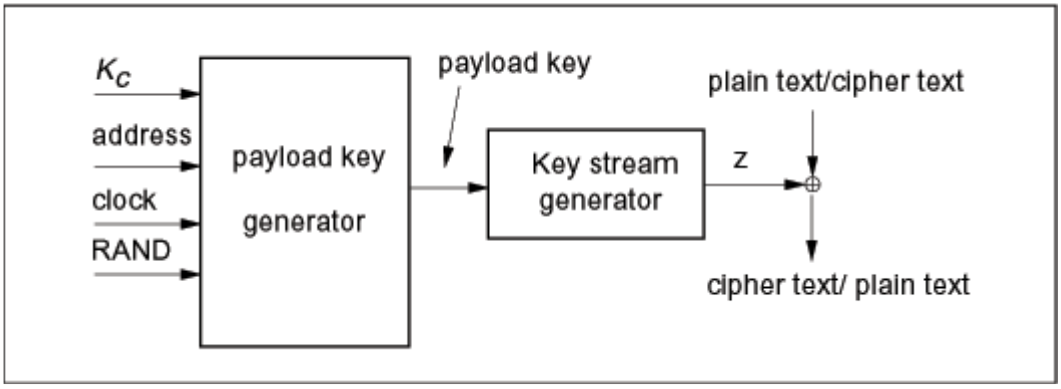

Fig 4.14 – Stream Cipher.

5. Vulnerabilità

5.1 Attacchi

5.1.1 Introduzione agli attacchi in Bluetooth

Durante l'associazione di un dispositivo ad un altro (**pairing**) avviene lo scambio di chiavi che dovrebbe assicurare la riservatezza della comunicazione; il segreto "condiviso" in questo caso è proprio il semplice "pin" che immettiamo nel nostro apparecchio. Se pensiamo al fatto che spesso questo pin è fissato a priori dal costruttore (per gli auricolari, kit vivavoce, navigatori satellitari) oppure che è limitato a 4 cifre, ci rendiamo conto di come possano nascere alcuni problemi. Gli esperti hanno mostrato come un attacco **brute-force** al meccanismo di pairing per alcuni dispositivi sia effettivamente possibile: nel caso di dispositivi con pin

preimpostato è addirittura attuabile online, ovvero direttamente contro il dispositivo vittima. Nel caso invece di "pin deboli" è possibile registrare il traffico, su tutti i 79 canali, attraverso un registratore frequenziale e poi testare le varie chiavi di collegamento (generate usando diversi pin) in maniera offline. Ovviamente quest'ultimo attacco, denominato attacco contro E22, non è alla portata di tutti per via dell'alto costo degli strumenti da utilizzare. E', comunque, sempre buona norma effettuare il pairing in luoghi considerati sicuri ed usare codici pin lunghi e difficili da indovinare. I problemi però esistono anche a livello applicativo: dal rilascio delle specifiche sino ad oggi, gli esperti di sicurezza hanno scoperto e segnalato numerose vulnerabilità dimostrando che, sebbene il protocollo è stato studiato approfonditamente, le aziende che hanno sviluppato i prodotti non hanno sempre tenuto in considerazione le possibili problematiche di sicurezza. Queste vulnerabilità dipendono unicamente dall'implementazione dei vari apparecchi e, nei casi più gravi, conducono al controllo completo del dispositivo vittima da parte dell'aggressore. La gravità di queste lacune non è giustificabile nemmeno con il ridotto range di utilizzo della tecnologia, in quanto gli esperti hanno dimostrato che modificando un semplice dongle Bluetooth, equipaggiandolo con un'antenna esterna (direzionale o omni-direzionale) è possibile aumentare considerevolmente la portata dei nostri dispositivi (si parla di centinaia di metri se il dongle è collegato con un'antenna tv). Per un aggressore questo significa poter effettuare degli attacchi Long-Distance o attacchi di tipo wardriving; il gruppo di ricerca Trifinite ha dimostrato come un semplice dongle modificato, è in grado di raggiungere la distanza di 1.78 Km.

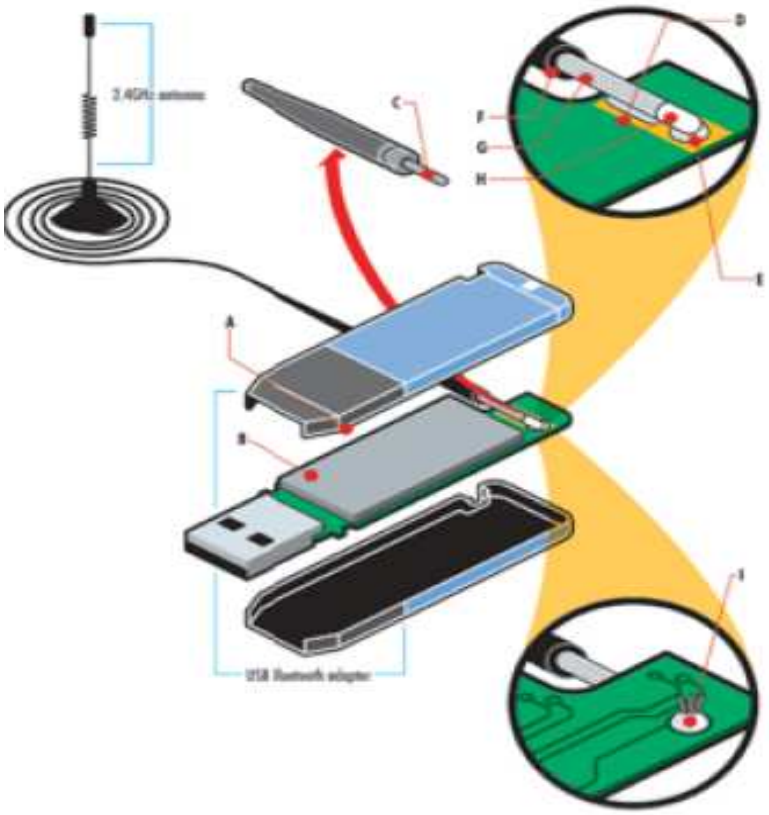

Fig 5.1 – Costruzione di un antenna per aumentare la portata del segnale Bluetooth.

Fig 5.2 –Dispositivo funzionante.

Come abbiamo visto nel primo capitolo dalla nascita dello standard Bluetooth ci

sono state varie versioni e anche un evoluzione degli attacchi a esso rivolti dallo schema sottostante possiamo avere una panoramica completa.

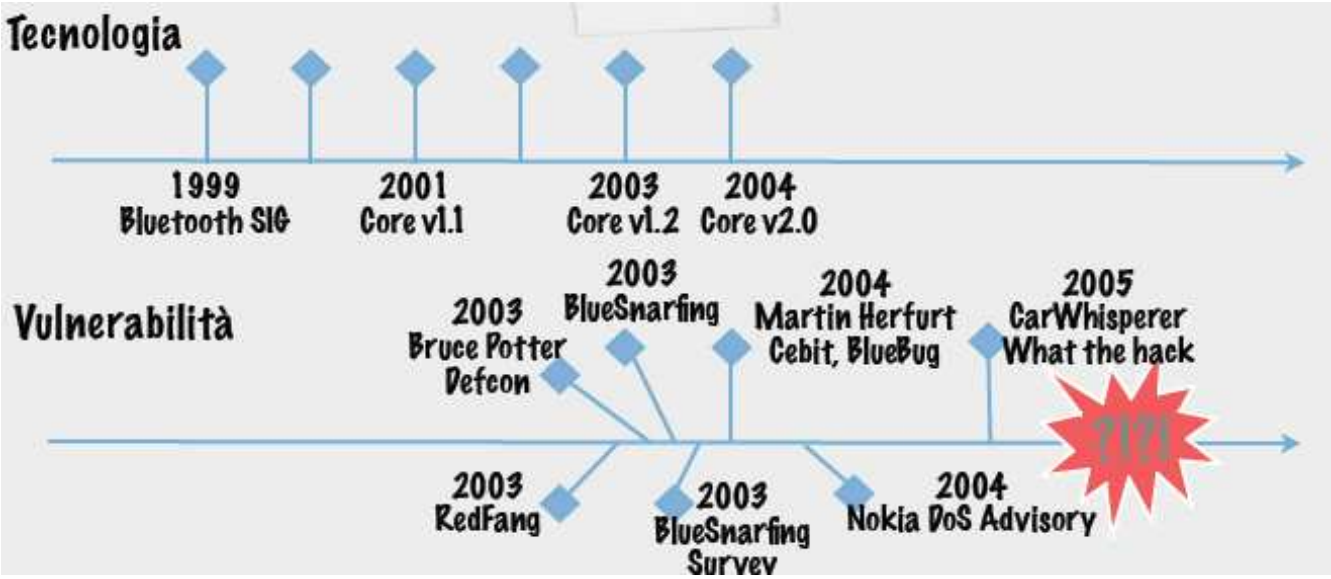

Fig 5.3 – Evoluzione dei malware in Bluetooth.

5.1.2 Attacchi a E22

Si cerca di scoprire il PIN del dispositivo vittima tramite un attacco BRUTE FORCE. Questo elemento è il "segreto condiviso" dell'algoritmo, ma spesso si tratta solamente di quattro cifre decimali . Un attacco "brute-force" può essere fattibile in due modalità:

- Attacco offline: serve un registratore frequenziale ($ costoso $)
- Attacco online: nel caso di dispositivi con pin assegnato

5.1.3 BlueJacking

Il Bluetooth cerca di rendere le interazioni tra dispositivi quanto più semplice per l'utente. Durante il discovery di altri apparecchi, per esempio quando vogliamo associare un telefono con il nostro computer, viene scambiato il nome identificativo dei device. Questa caratteristica, che semplifica l'utilizzo, è intrinsecamente pericolosa. Il nome del dispositivo è infatti un campo di testo che può contenere una stringa maggiore o uguale di 248 caratteri (ove possibile); utilizzando questa stringa è possibile scambiare messaggi tra dispositivi. Sebbene questa caratteristica è spesso usata per conoscere e socializzare con nuove persone dotate anch'essi di apparecchi Bluetooth (*toothing*), abbinata a tecniche di social engineering può

compromettere una delle fasi più delicate: quella del pairing tra dispositivi. L'utente inesperto ricevendo, ad esempio, un messaggio di questo tipo: "Problemi alla rete, digita 1234 per associare il telefono alla cella" o “Vodafone ti regala una suoneria..digita 1234 per proseguire” potrebbe essere tratto in inganno, facendo diventare trusted un dispositivo sconosciuto che quindi acquisirebbe tutti i privilegi necessari a compromettere i dati e le comunicazioni. Non dobbiamo stupirci se attacchi così semplici spesso sono i più efficaci: le tecniche di phishing in ambito web, sfruttano la stessa inesperienza degli utenti. La popolarità di questo tipo di abuso ha dato luogo allo sviluppo di una serie di software appositi (Freejack, Meeting, Bluejack, ecc.) usati spesso dai più giovani nei locali di divertimento.

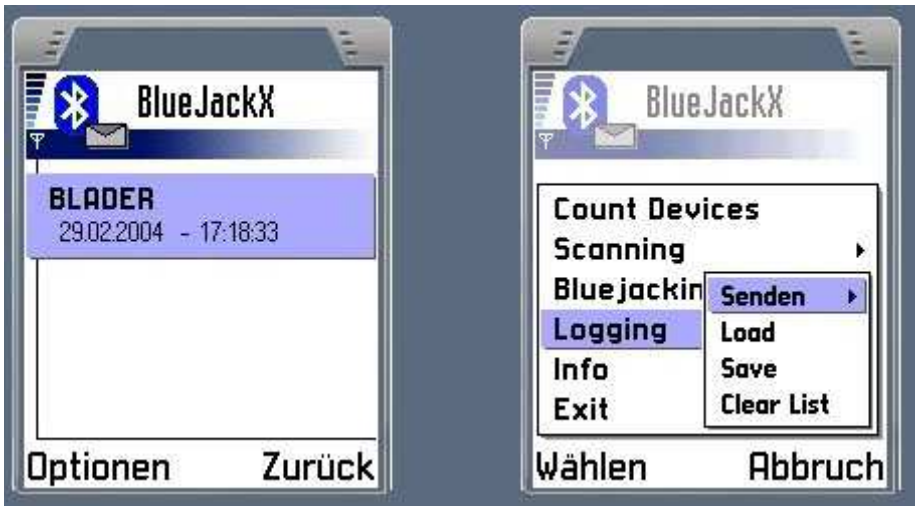

Fig 5.4 – Attacco tramite BlueJackX.

5.1.4 Discovery Mode Abuse

Su molti dei dispositivi in commercio è possibile selezionare la modalità di funzionamento Bluetooth:

- acceso (visibile)
- acceso (nascosto)
- spento

Selezionando l'opzione nascosto o invisibile, il dispositivo non fa nient'altro che scartare tutte le richieste di inquiry, inviate in "broadcast" da altri dispositivi che vogliono conoscere la presenza di soggetti con cui comunicare. A differenza di quanto si crede però non vengono in alcun modo disabilitati i servizi, ma solamente rifiutate le richieste che giungono all'SDP; in questo modo, per alcuni apparecchi, è comunque possibile interrogare direttamente il singolo dispositivo che risponderà alle richieste normalmente. Nascondere un dispositivo non deve quindi essere considerato come un meccanismo di protezione infallibile. @Stake, nota società di sicurezza informatica acquisita recentemente da Symantec, ha pubblicato uno strumento software in grado di scoprire eventuali dispositivi nascosti. Il funzionamento di questo software, chiamato RedFang, è basato su un meccanismo di brute-forcing. I primi 24 bit di un indirizzo Bluetooth sono fissi e dipendenti dal costruttore; i successivi 24 identificano invece univocamente il dispositivo. Scegliendo un particolare produttore, risulta computazionalmente possibile tentare di indovinare gli ultimi bit effettuando continue richieste (circa 1 ora, in versione multithread).

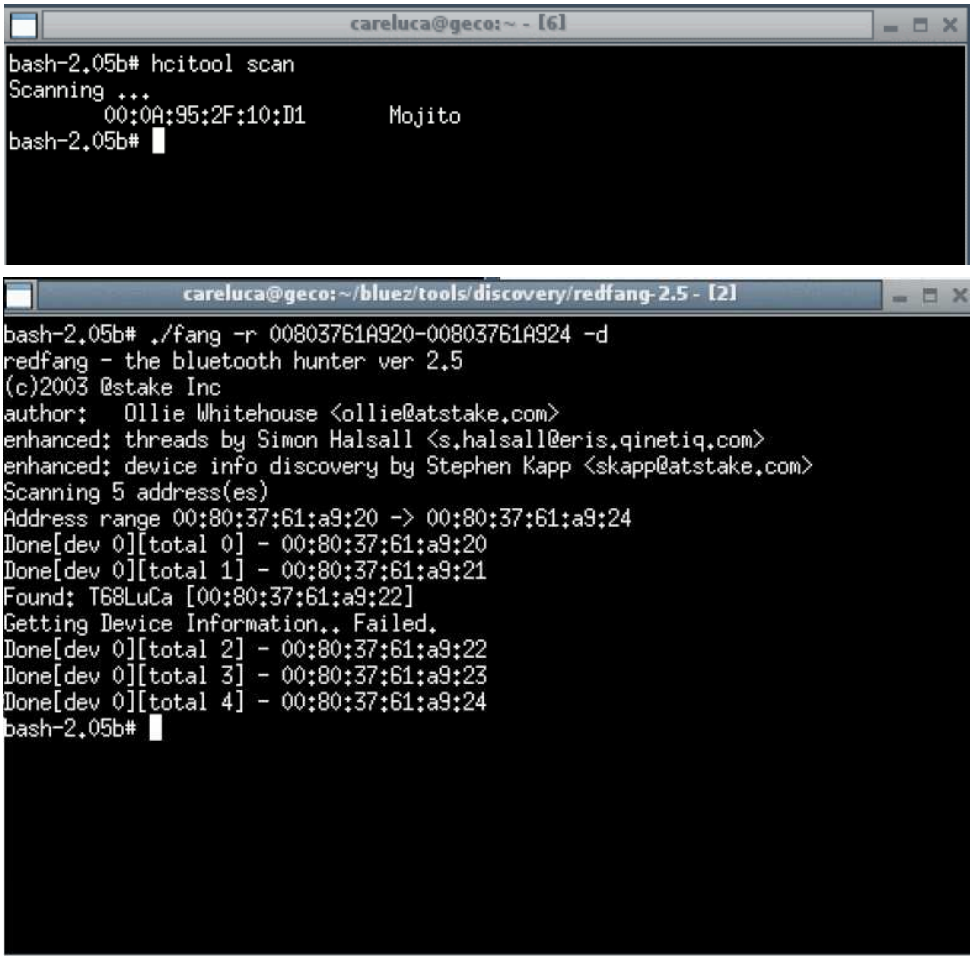

Fig 5.5 – Tecnica del Discovery Mode Abuse (Redfang).

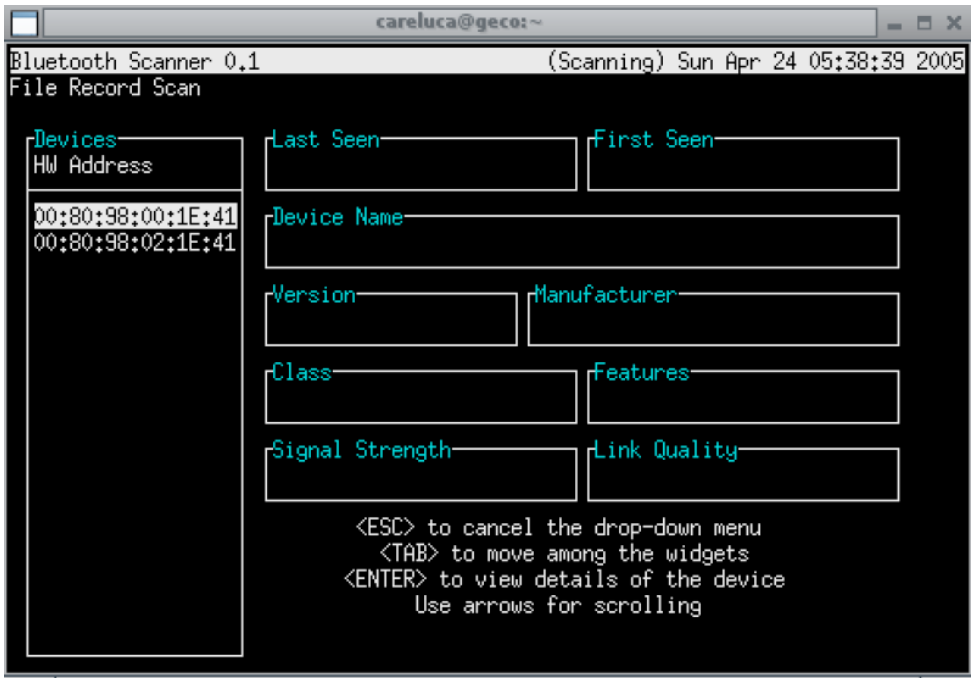

Fig 5.6 – Tecnica del Discovery Mode Abuse (BlueSnif).

5.1.5 BlueSnarf

Per realizzare questo tipo di attacco un aggressore non deve fare nient'altro che collegarsi al servizio OBEX Push usato spesso per scambiarsi biglietti da visita elettronici. In alcuni cellulari questa funzionalità è implementata in maniera errata e permette, oltre alla ricezione di file, anche l'OBEX Get ovvero la richiesta di file. In questo modo, conoscendo la presenza di qualche oggetto presente sul dispositivo è possibile scaricarlo senza autenticazione; l'assenza di autenticazione è una caratteristica intrinseca del servizio OBEX che, se è però implementato correttamente, non deve permettere il download di file. La necessità di conoscere il path di un oggetto sul dispositivo remoto non è comunque un problema: moltissimi apparecchi memorizzano le informazioni su file di testo la cui disposizione è nota e dipendente dal sistema. Per esempio, i prodotti Ericsson e SonyEricsson di prima generazione salvano la rubrica telefonica in telecom/pb.vcf oppure il calendario in telecom/calc.vcs, così come molti altri telefonini. Se uno di questi dispositivi è bacato, l'attacco è presto fatto utilizzando qualsiasi client OBEX (obexftp per Linux, obex-commander per Windows). I dispositivi afflitti da questa vulnerabilità non sono affatto pochi: Ericsson T68, Sony Ericsson T68m, T68i, T610, Z1010, Z600, R520m, Nokia 6310, 7650, 8910 e molti altri. In seguito i ricercatori hanno individuato una vulnerabilità molto simile che è stata chiamata **BlueSnarf++** in quanto oltre al download di file permette un accesso completo al filesystem dei

dispositivi vittima. Su questi apparecchi risulta possibile vedere i file presenti, ma anche eliminarli, senza dover effettuare alcun pairing tra dispositivi. Una variante è la famosa vulnerabilità riscontrata su alcuni dispositivi Motorola, chiamata **HeloMoto**. Essa permette la creazione di una nuova entry nell'elenco dei dispositivi paired. Sfrutta un bug del servizio OBEX e offre all'aggressore il pieno accesso ai comandi AT del dispositivo falsamente associato.

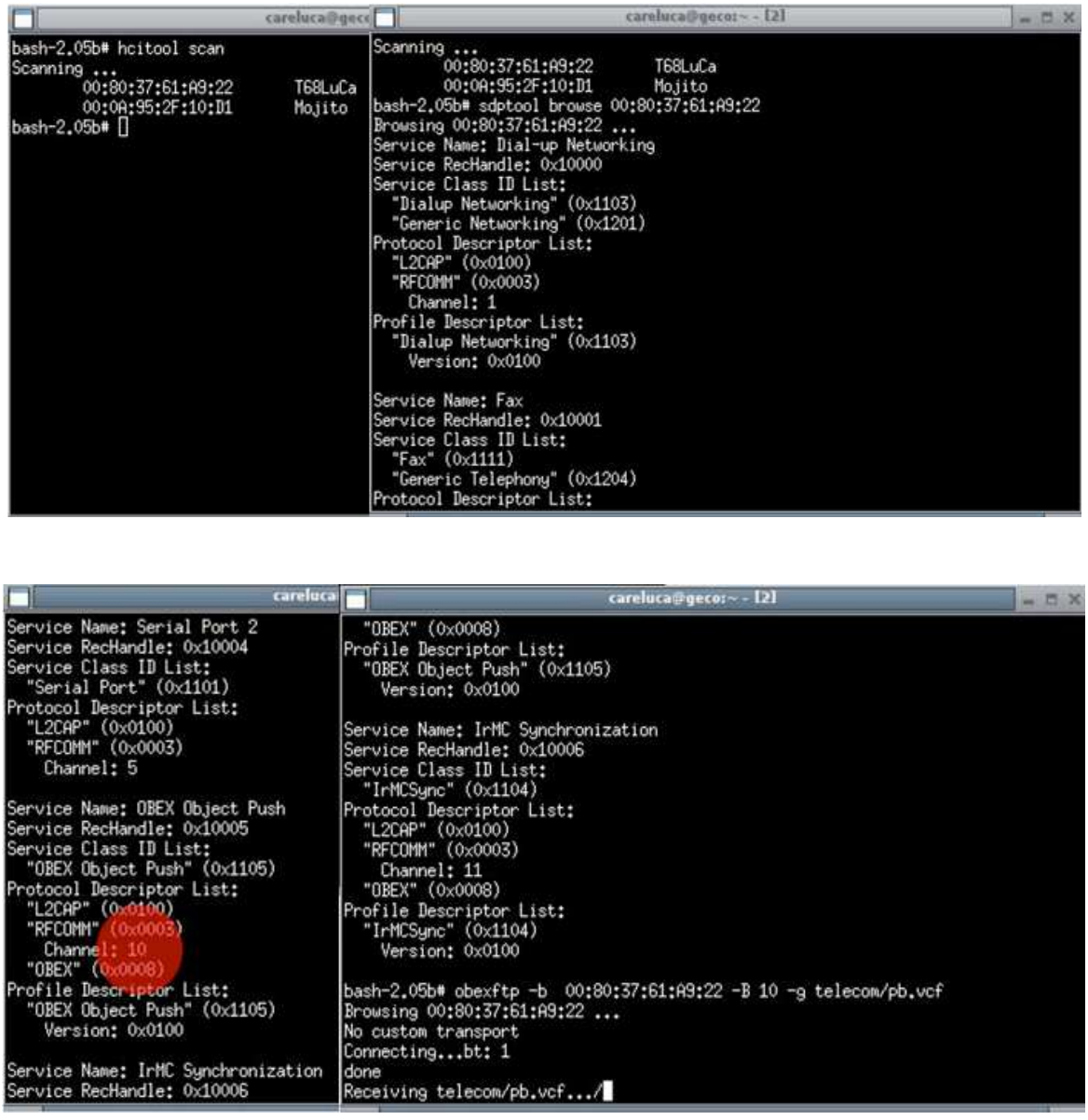

Fig 5.7 – Tecnica del BlueSnarf. Nel cerchio rosso si vede come viene individuato il comando OBEX su un dispositivo mobile.

5.1.6 BlueBug

Bluebug è il nome di una vulnerabilità presente in alcuni cellulari che permette un pieno accesso ai comandi AT del dispositivo. I comandi AT definiscono un set di comandi che permettono di ottenere il controllo completo del dispositivo: invio di chiamate, invio, lettura e cancellazione di SMS, modifica dei parametri di configurazione del telefono, ecc. Il Bluebug, obbliga il cellulare a telefonare a un numero indicato dal pirata. Ne può nascere quindi un business illegale: magari il numero è ad alto costo e, come con i dialer, chi lo gestisce è d'accordo con il pirata per spartirsi i ricavi. Ancora una volta il problema è legato ad un'implementazione errata dello stack Bluetooth in cui esistono dei servizi sul canale RFCOMM non pubblicati e non annunciati tramite SDP, ma che posso essere comunque usati. Spesso questi canali di comunicazione sono abilitati dal costruttore per eventuali test sui prototipi, ma poi non vengono rimossi in fase di produzione. Collegandosi ad uno di questi canali è quindi possibile impartire qualsiasi comando al dispositivo remoto, scaricare dati e compiere altre azioni illecite.

5.1.7 BlueSmack

BlueSmack è un tipico attacco DOS (Denial of Service) che permette di far diventare instabile un sistema operativo sino a fargli generare delle eccezioni critiche. Questo tipo di attacco è la rivisitazione del classico Ping of Death che affliggeva Windows 95, in ambiente Bluetooth. Come nel caso del Ping of Death, si incrementa oltre misura la dimensione di un pacchetto echo request (L2CAP ping) che verrà poi spedito verso il dispositivo vittima. Alcuni apparecchi, oltre ad un certa dimensione del pacchetto, ricevono il dato, ma generano degli errori che fanno bloccare completamente il sistema operativo; è il caso di alcuni modelli di Compaq IPAQ con sistema operativo Windows Mobile in cui se il numero dei byte del pacchetto ricevuto è superiore a 600 mostrano un messaggio di errore con un conseguente blocco del sistema.

5.1.8 Blooover

Esistono in circolazioni programmi che permettono di scovare queste vulnerabilità. Uno di questi è chiamato BloooverII e nella sua ultima release permette di provare molte delle vulnerabilità mostrate (BlueSnarf, BlueBug ma anche HELOMoto e Malformed Objects) in maniera veloce tramite una semplice interfaccia utente. Il software è disponibile per tutte le piattaforme che supportano Java Micro Edition

(MIDP 2.0) e le Bluetooth API (JSR-82); per questa caratteristica si presta perfettamente ad essere installato su dispositivi cellulari di ultima generazione.

5.1.9 BluePrinting

E' un metodo per avere informazioni tecniche sul dispositivo, in maniera remota. Attraverso la creazione di un database di "impronte" è possibile conoscere le caratteristiche tecniche dei dispositivi e le eventuali vulnerabilità. Per esempio il dispositivo con scheda Bluetooth numero :

Analizzato con BluePrinting ci dà le seguenti informazioni in output:

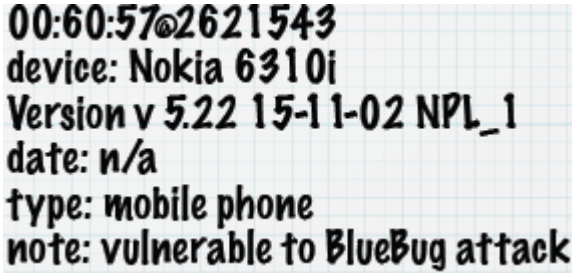

Fig 5.8 – Informazioni ricavate dal BluePrinting.

Queste informazioni possono essere usate da un eventuale aggressore per poter effettuare un attacco.

5.2 Il progetto BlueBag

Nato nel 2005 e terminato nel 2007 da un gruppo di ricercatori del Politecnico di Milano. Aveva come scopo comprendere la reale diffusione della tecnologia, determinare il reale rischio di fronte ad un possibile attacco, in particolare sono state studiate alcune caratteristiche dei dispositivi:

- Il loro numero all'interno di un area
- Il loro range di trasmissione
- Il tempo di esposizione
- Azioni e Reazioni di una potenziale vittima di attacco

- Limitazioni Ambientali

- Limitazioni Tecnologiche

Per rispondere a queste domande è stato realizzato il primo survey italiano dei dispositivi Bluetooth avvalendosi di un particolare strumento realizzato ad hoc: la *BlueBag*.

Fig 5.9 – BlueBug (1).

Fig 5.10 – BlueBug (2).

La BlueBag è basata su un sistema mini-ITX al fine di ridurre al minimo i consumi avendo comunque una discreta potenza computazionale, la BlueBag utilizza il dongle modificato con l'antenna per rilevare i dispositivi a distanza maggiore (intorno ai 150 metri). A livello software, la BlueBag utilizza un sistema GNU/Linux Gentoo con kernel 2.6 e lo stack protocollare BlueZ, l'implementazione più nota e diffusa per Linux. In maniera automatica il software della BlueBag cerca di associare anche la precisa tipologia e il particolare modello dell'apparecchio; (BluePrinting). La BlueBag ricerca tutti i dispositivi presenti nell'ambiente e cerca di effettuare il trasferimento di un file grazie al servizio OBEX PUSH, valutandone il tasso di successo, in questo modo è possibile determinare empiricamente il numero di potenziali vittime da Bluetooth malware. In meno di 24 ore totali di scansione nella città di Milano e in svariati luoghi, la BlueBag ha identificato 1405 dispositivi con una distribuzione del 93% per i telefoni cellulari, 3% notebook, 2% pda, 2% antenne GPS e altro. Dalle misurazioni effettuate tramite l'OBEX Pusher è stato inoltre possibile stimare il tasso medio di successo, valutando al 7.5% il numero delle persone che senza conoscere la sorgente ed il contenuto del trasferimento hanno accettato di buon grado un file potenzialmente dannoso. Il tempo medio di visibilità dei dispositivi che può essere letto come il tempo utile ad un eventuale aggressore per portare a termine un attacco:

- 12.3 secondi per il centro commerciale
- 10.1 sec. per il campus universitario
- 23.1 sec. per l'aeroporto
- 14.4 sec. per gli uffici generali di una banca

Tempi ridotti ma decisamente sufficienti per portare a compimento un attacco.

5.3 Conclusioni e suggerimenti per aumentare la sicurezza in Bluetooth

Bluetooth, inteso come standard è sicuro, i problemi sono a livello applicativo e di implementazione. Per rendere più sicuro il suo utilizzo possono essere espressi dei consigli a riguardo:

- Scegliere codici PIN non banali e lunghi (dove consentito dagli apparecchi). I codici composti da cinque o più cifre sono più difficili da indovinare.

- Evitare il pairing tra dispositivi Bluetooth in ambienti affollati o poco sicuri; un semplice errore nell'associare dispositivi untrusted potrebbe compromettere i propri dati

- Utilizzare il dispositivo in modalità nascosta o invisibile per allungare i tempi di un'eventuale aggressione, e cambiarla in "rilevabile" solo quando la si deve utilizzare.

- Scegliere dispositivi ritenuti sicuri dall'intera comunità di esperti che operano in questo settore. Attraverso le mailing list pubbliche o i forum è possibile trovare eventuali advisory relative a dispositivi bacati.

- Evitare di memorizzare dati riservati, come il codice fiscale, il numero di carta di credito e le password, sui dispositivi wireless.

- Installare un buon antivirus.

Soluzione tecniche auspicate:

- Fornire autenticazione ai livelli più alti del protocollo

- Maggior impegno e sinergia tra Bluetooth SIG e case produttrici

6. Bibliografia

1. Chatschik Bisdikian. "An Overview of the Bluetooth Wireless Technology", IEEE Communications Magazine, Vol. 39, No. 12, pp. 86-94, December 2001.

2. Creighton T. Hager, Scott F. Midkiff. "An Analysis of Bluetooth Security Vulnerabilities", Wireless Communications and Networking, Vol. 3, No. 16-20, pp. 1825 -1831, March 2003.

3. G.Lamm G. Falauto, J. Estrada, J. Gadiyaram, D. Cocherham. "Security Overview of Bluetooth", COSIC, Internal Report, June 2004.

   disponibile all'indirizzo:

   http://www.cosic.esat.kuleuven.be/publications/article-565.pdf

4. Marjaana Träskbäch. "Security of Bluetooth: An overview of Bluetooth Security", Department of Electrical and Communications Engineering, Internal Report, March 2004.

   disponibile all'indirizzo:

   http://www.cs.hut.fi/Opinnot/Tik-86.174/Bluetooth_Security.pdf

5. Karen Scarfone John Padgette. " Guide to Bluetooth Security (Draft)", National Institute of Standards and Technology – NIST, Special Publication 800-121, September 2008.

   disponibile all'indirizzo:

   http://csrc.nist.gov/publications/nistpubs/800-121/SP800-121.pdf

6. Luca Carrettoni. "Bluetooth Security", Smau e-Academy, 22 ottobre 2005.

   disponibile all'indirizzo:

   http://www.ikkisoft.com/stuff/bluesec.pdf

7. Luca Carrettoni. "Bluetooth Malware", Articolo pubblicato su www.hakin9.org, Nº 4 /2007.

   disponibile all'indirizzo:

   http://www.ikkisoft.com/stuff/bluemalware_it.pdf

8. “BLUETOOTH SPECIFICATION Version 3.0 + HS”, Specifica dello standard pubblicata sul sito www.bluetooth.com, April 2009.

   disponibile all'indirizzo:

   http://bluetooth.com/NR/rdonlyres/298BE70B-4353-4492-9A91-160549463612/10885/Core_V30__HS.zip