Säkerhet i WPAN

Kajsa Goffrich Karl Vestgöte
Linköpings universitetet, Sweden
Email: {kajgo449,karve058}@student.liu.se

Sammanfattning

Trots att säkerhetstänkandet har varit prioriterat vid utveckling av standarden bluetooth har man inte kunnat undvika vissa luckor. En stor del av dessa beror på att användare hanterar sina enheter felaktigt, ofta beroende på lathet och slarv men även på grund av bristande kunskaper och dålig information från tillverkaren. De tillverkare som implementerar bluetooth i sina enheter orsakar också en del säkerhetsproblem genom att använda protokollen felaktigt.

Även om de flesta luckorna ligger tämligen nära slutanvändarna innehåller protokollet också vissa brister. Längden på krypteringsnyckeln och PIN-kod hör till de säkerhetsbrister som påpekas oftast, men även andra exempel på brister i tekniken finns.

1. Inledning

Detta projekt undersöker WPAN ur ett säkerhetsperspektiv. Den avgränsning som gjorts är bluetooth och de tillämpningar som finns i vardagslivet. Hit hör främst mobiltelefoner, bärbara datorer och andra praktiska produkter riktade till allmänheten snarare än professionella tekniker.

1.1 Bakgrund

Olika artefakter med WPAN börjar bli väldigt vanliga och användandet av bluetooth ökar hela tiden i telefoner och datorer. Mot bakgrund av detta undersöks bluetooth ur ett säkerhetsperspektiv med fokus på de eventuella säkerhetsrisker som finns i den här relativt nya tekniken.

Det primära syftet med denna rapport är att få bättre kunskap om bluetooth och dess användningsområden, samt att visa på vilka risker som uppkommer när man ger handdatorer och mobiltelefoner möjlighet att kommunicerar med omvärlden. En annan central del är också att ta reda på om utvecklarna av de olika versionerna har tagit lärdom av tidigare problem, exempelvis med trådlösa nätverk. Den stora frågan här är om de säkerhetshål som upptäcks i tidiga versioner verkligen täpps till innan nästa variant släpps.

Slutligen undersöks också om tillverkare av mobiltelefoner och liknande följer standarder och har en genomtänkt design för att förhindra onödiga säkerhetsrisker. Hur väl kan den vanliga användaren hantera bluetooth, är det lätt lura honom/henne att acceptera anslutningar från okända. Vet han/hon om att mobiltelefonen i fickan kommunicerar med andra.

1.2 Förväntade resultat

De övergripande resultat vi förväntar oss av detta arbete är följande främst att bluetooth är inte bara möjligt att på olika sätt knäcka, utan också relativt enkelt. Dock tror vi att förbättringar i säkerheten görs i takt med att nyare versioner och nya standarder når marknaden.

En annan stor anledning till att bluetooth kan vara relativt enkelt att knäcka torde också vara att tillverkare inte följer de säkerhetsföreskrifter som faktiskt finns. Hela Internet är förmodligen också fullt av olika former av metoder och program för att komma åt artefakter med inbyggd bluetooth.

2. Protokoll

Detta stycke innehåller en översiktlig beskrivning av protokollet bluetooth. För en djupare inblick se [1].

2.1 Bluetooth radio

Frekvenserna som bluetooth använder spänner mellan 2.402GHz och 2.480GHz, området är uppdelat i 79 stycken 1MHz steg som används vid frekvenshopp.

Bluetoothapparater är uppdelade i tre klasser baserade på vilkens signalstyrka de använder. Klass 1 är konstruerad för lång räckvidd, 100 m, och sändaren har en maxeffekt på 100 mW. Klass 2 är den vanligaste och har räckvidden, 10 m, och en maxeffekt på 2,5 mW. Till sist finns klass 3 som är för kort räckvidd ungefär 10 cm och den har en sändareffekt på maximalt 1 mW [1].

2.2 Bluetooth baseband

Baseband synkroniserar frekvenshoppen och klockorna hos de kommunicerande bluetoothartefakterna och tillhandahåller två olika typer av fysiska radiolänkar, Synchronized Connection Oriented (SCO) och Asynchronous Connectionless (ACL) som kan sändas samtidigt på samma kanal. Vanlig data skickas över ACL länken medan SCO länken tar hand om ljuddata och ibland också vanlig data. Datatrafiken kan förses med olika nivåer av felkorrigering [2].

2.3 Link Manager Protocol (LMP)

Link Manager Protocol (LMP) ansvarar för att skapa och konfigurera länkar mellan kommunicerande bluetooth artefakter. I det ingår att kontrollera baseband paketen och också att bestämma vilken storlek de ska ha. Även säkerheten i bluetooth ansvarar LMP för då den utför verifiering, kryptering, generering och kontroller av länk och krypteringsnycklar samt byte av dessa. LMP styr också bland annat signalstyrkan hos radiosändaren [2].

2.4 Service Discovery Protocol (SDP)

Att upptäcka tillgängliga tjänster är en mycket viktig del i bluetooths design eftersom de är grunden i all form av användning. Genom att använda Service Discovery Protocol (SDP) kan man få information om bluetoothartefakter i närheten och tjänsterna de erbjuder. När användaren väl har fått information om tillgängliga tjänster kan hon/han välja fritt bland de hon är intresserad av och därefter kan man upprätta en länk mellan två eller flera bluetoothapparater [2].

3. Användande

Bluetooth är en standard för trådlös kommunikation över korta avstånd. Tekniken är robust, flexibel, billig och förbrukar lite energi vilket gör den till en av de flitigast använda inom området (www.bluetooth.com). En bluetooth-enhet består av en sändare och mottagare för radiofrekvenser (RF tranciever), basband och en protokollstack.

Det fysiska lagret (RF) använder det icke licensierade bandet vid 2,4 Ghz. För att undvika interferens och dämpning används en frekvensmodulator och överföringshastigheten är i orginalutförandet Basic Rate 1 Mbps eller 2-3 Mbps i ett mer avancerat utförande kallat Enhanced Data Rate. Vid överföring bildas ett piconät av ett antal enheter där en agerar master, vilket innebär att den är referens vid synkroniseringen av överföringen, och övriga enheter är slavar. Enheterna synkroniseras mot en masterns klocka och i ett givet frekvensmönster [4].

3.1 Tänkta användningsområden

Tillverkarna av bluetooth ser tekniken som det självklara valet vid behov av en tillförlitlig och smidig trådlös kommunikation över korta avstånd. Det är en global standard som har fördelen att alla enheter där stöd för bluetooth implementerats kan kommunicera med varandra utan att drivrutiner behöver installeras. Tekniken är utvecklad för tillämpningar såsom bland annat mobiltelefoner, bärbara datorer, bilar, headsets, mp3-spelare. I sitt senaste och fjärde utförande har många onödiga förluster arbetats bort och detta har resulterat i att energikonsumtionen sjunkit samt att både storleken på och priset för chipet minskat.

En annan anledning till att bluetooth blivit så vida spritt är att tekniken är väldigt enkelt att hantera för användaren. Kryptering och autentisering med PIN-kod finns inbyggt från början, och det enda som krävs av användaren för säker överföring är att han eller hon slår sin PIN-kod [4].

3.2 Faktiska användningsområden

De av tillverkaren föreslagna användningsområdena för bluetooth stämmer väl överens med de faktiska bluetooth-enheterna som finns på marknaden. Mobiltelefoner, tillbehör till dessa och headsets är de områden där tekniken är mest använd. Andra vanliga områden är olika kontorstillämpningar såsom skrivare, persondatorer och liknande. Tillverkaren lyfter även fram lite mer oväntade tillämpningar av tekniken inom vården, exempelvis trådlösa pulsmätare och handscanners [4].

Som med de flesta flexibla och vitt spridda teknikerna finns också olika exempel på riktigt obskyra tillämpningsområden. Några exempel på detta följer här:

En cykelhjälm med bluetoothmottagare som marknadsförs som en säkerhetsfördel då cyklisten slipper prata i mobiltelefon eller lyssna på musik via en extra öronsnäcka eller hålla artefakten i handen [5].

En teddybjörn avsedd att bäras offentligt och som programmeras via bluetooth. Tillverkaren motiverar björnen med att inget attraherar vackra kvinnor mer än män som bär runt på teddybjörnar [6]

En golfvagn som har en inbyggd bluetoothmottagare. Golfaren själv bär på en sändare vilket möjliggör att vagnen följer efter sin ägare på golfbanan. Marknadsförs som den perfekta presenten till en lat, icke-paranoid och givetvis pryltokig golfare [7].

4. Säkerhet

Grundtanken med bluetooth är att användaren ska kunna få en trådlös överföring på korta avstånd som upplevs vara säker. Utvecklarna av standarden hade under hela processen säkerheten i bakhuvudet och satsade stora resurser på att få till en säker överföring. Säkerheten för åtkomst av enheter är indelad i tre nivåer där nivå ett, osäker, är den lägsta, nivå två, säkerhet på servicenivå, är mellan och nivå tre, säkerhet på länknivå, är den säkraste. Vilken säkerhetsnivå som implementeras avgörs av tillverkaren av de produkter som innehåller bluetooth-utrustning. En närmare beskrivning av de olika nivåerna återfinns nedan.

I fallet bluetooth har utrustning och tjänster olika säkerhetsnivåer. När det gäller utrustning finns två nivåer, betrodd enhet och icke betrodd enhet. För tjänser finns tre nivåer, en hög som kräver både godkännande och verifiering, en mellanhög som endast kräver verifiering samt en låg nivå där tjänsten är öppen för alla enheter.

Företaget bluetooth framhåller noga att de säkerhetsbrister som finns och de fall där data blivit påverkad helt och hållet beror på implementeringsfel hos tillverkaren av enheten. Detta motiveras med att krypteringsalgoritmen som finns i standarden bluetooth är säker och således kan inte eventuella säkerhetshål bero på standarden [4].

4.1 Inbyggd säkerhet

Då räckvidden är starkt begränsad med bluetooth betyder det att en eventuell attackerare måste befinnas sig relativt nära den enhet som ska attackeras. Detta kan bidra till ökad säkerheten i tex större företagsbyggnader. Denna begränsning av räckvidd kan dock ge en falsk trygghet, man bör vara medveten om att det finns ett flertal sätt att plocka upp signalerna på längre avstånd. Till exempel kan man med riktade antenner plocka upp en signal på upp till 1,5 km avstånd, förutsatt att man har fri siktlinje.

En annan fördel med bluetooth är att signalen hoppar mellan 79 olika frekvenser 1600 per sekund. Tack vare detta försvåras avlyssning då det blir krångligare att hitta en specifik signal och följa den.

4.2 Parning och verifiering

Bluetooth initieringsprocess består av tre steg:

1. Skapa en initieringsnyckel (K-init)
2. Skapa en länknyckel (K-ab)
3. Verifiering

Efter de här tre stegen kan bluetooth artefakterna derivera fram en krypteringsnyckel. Innan parningen kan börja så måste användaren mata in en PIN-kod båda apparaterna. Om en av artefakterna har en PIN-kod som ej går att ändra, tex bluetooth hörlurar, så måste dess PIN-kod slås in i den andra artefakten.

Att skapa K-init sker genom att en algoritm som heter E-22 används för att skapa initieringsnyckeln K-init som är ett 128 bitars tal och har tre indata:

1. en BD_ADDR (bluetooth adress).
2. PIN-koden och dess längd.
3. Ett 128-bitars slumptal, IN_RAND.

Om en av bluetooth artefakterna som skall paras har en fast PIN nyckel så används BD_ADDRen från den överliggande artefakten, om bägge har variabel PIN så används slavens BD_ADDR.

Mastern (A) vet om slavens (B) BD_ADDR då den innan parningsprocessen påbörjas har skickat en förfrågan till slaven om dess BD_ADDR, på samma sätt vet också slaven om masterns BD_ADDR.

PIN-koden matas in av användaren i bägge artefakterna. Mastern slumpar fram ett 128-bitars tal, IN_RAND, och sänder det i klartext till slaven. Därefter använder mastern E22 med BD_ADDR-B(slavens bluetoothadress), IN_RAND och PIN som indata för att räkna ut initieringsnyckeln K-init. Slaven tar emot IN_RAND och gör på samma sätt för att få fram K-init.

K-init används endast under parningsprocessen och kastas därefter bort.

Länknyckeln (K-ab) skapas genom att bluetooth-artefakterna använder initieringsnyckeln K-init för att utbyta två nya 128-bitars slumptal, LK_RAND-A och LK_RAND-B. Bägge apparaterna väljer ett 128-bitars slumptal och och sänder det till den andre efter att bitvis ha XORat det med K-init. Efter att ha mottagit den andra partens sända meddelande och XORat det bitvis med K-init så har bägge artefakterna LK_RAND-A och LK_RAND-B. Efter att ha kört algoritm E21 två gånger, en gång med LK_RAND-A och BD_ADDR-A som indata och den andra gången med LK_RAND-B och BD_ADDR-B som indata, fås LK_K-A och LK_K-B. Därefter XORas LK_K-A och LK_K-B ihop och tillsammans bildar de länknyckeln K-ab. Det här görs på bägge bluetooth-artefakterna så de båda har länknyckeln.

Det sista steget är verifiering. Efter att länknyckeln K-ab skapats hos bägge artefakterna så genomförs en ömsesidig verifiering av varandra. Det bygger på ett utmaning-svars koncept. En av bluetooth-artefakterna, verifieraren, slumpar fram ett 128-bitars tal, AU_RAND-A, och sänder det i klartext till den andra parten, pretendenten. Pretendenten beräknar ett 32-bitars svar, SRES, genom att använda algoritm E1 med AU_RAND-A, K-ab och sin bluetoothadress (BD_ADDR-B) som indata. Han skickar därefter SRES till verifieraren som också beräknar SRES, med den andra partens BD_ADDR, och jämför det resultatet med den mottagna SRESen. Stämmer dom med varandra så byter artefakterna roller och upprepar processen. En ytterligare effekt av denna verifiering är att parallellt med beräknandet av SRES beräknas också en 96-bitars offset som används vid framtagningen av krypteringsnyckeln. [10] [12]

4.3 Kryptering

När parning och verifiering genomförts kan kryptering ske utan vidare insats av användaren. Krypteringsnyckeln, som är mellan åtta och 128 bitar beroende på säkerhetsnivån, beräknas utifrån länknyckeln för vidare överföring.

4.4 Krypteringsnyckeln

Efter parnings- och verifieringsprocessen så har båda bluetooth-artefakterna en 128-bitars länknyckel K-ab, den används nu för att generera kryperingsnyckeln K-c. Genom att köra algoritm E3 med K-ab, EN_RAND och den 96-bitars krypteringsoffset som räknades fram i verifieringsprocessen som indata får man fram krypteringsnyckeln K-c. K-c modifieras sedan för att längden ska stämma med den som bluetooth artefakterna tidigare har kommit överens om. Nyckeln används sedan tillsammans med masterns BD_ADDR och en klocka som är olika för varje paket för att bilda det initiala värdet för E0. E0 genererar sedan en bit-ström som XORas ihop med klartexten för att kryptera den. [12]

4.5 Krypteringsalgoritmen

Den algoritm som är implementerad i protokollet heter E0 Stream Cipher. Kryptering innebär att en sekvens med pseudoslumpmässiga siffror genereras och kombineras med klartexten genom en XOR operation. Nyckellängden kan variera, men den är oftast 128 bitar. Vid varje iteration genererar E0 en bit genom att använda fyra skiftregister av olika längd (25, 31, 33, 39 bitar) och två inre tillstånd som vardera är 2 bitar långa. Vid varje

klockning skiftas registrerna och de inre tillstånden uppdateras med det nuvarande tillståndet, det föregående tillståndet och värdena i skiftregistrerna. Fyra bitar från skiftregistrerna adderas sedan ihop, algoritmen XORar den summan med värdet i 2-bitars registret. Första biten av resultatet är utsignal från krypteringen.

E0 är indelad i tre steg:

1. Datanyckel genereras
2. Överföringsnyckeln genereras
3. Kryptering

Det initiala tillståndet i bluetooth använder samma struktur som generatorn av slumptalsbitströmmarna. Detta resulterar i två kombinerade E0 algoritmer. Först tas det initiala 132-bitars tillståndet fram av fyra komponenter (128 bitars nyckeln, bluetooh adressen på 48 bitar, och den 26 bitars ”masterns” klocka.) Utdatan processas sen av en polynomisk funktion och resultatet blir en nyckel. Denna går genom ett andra steg vilket genererar den ström som används för kodning. Nyckeln varierar mellan 8 och 128 bitar men är alltid en multipel av två. Vanligast är 128 bitars nyckel. Den sparas i det andra stegets skiftesregister. 200 pseudoslumpmässiga bitar genereras sen under 200 klockcykler och dom sista 128 bitarna sparas i skiftregistrerna. Det är nyckelgeneratorns starttillstånd [9] [13].

4.6 Kryptoanalys

Det har gjorts flera olika försök att attackera E0 och ett flertal svagheter har hittats. 1999 visade Miia Hermelin och Kaisa Nyberg att E0 kunde knäckas med 2^{64} operationer istället för 2^{128}, förutsatt att 2^{64} bitar av den krypterade texten var känd. Den här attacken förbättrades sedan av Kishan Chand Gupta och Palash Sarkar. Scott Fluhrer hittade en teoretisk attack med 2^{64} förberäknade operationer och en sökning efter nyckeln som omfattade 2^{65} operationer. Han kom fram till att den maximala säkerheten för E0 är likvärdig den som en 65 bitars nyckel tillhandahåller, samt att längre nycklar inte ökar säkerheten.

2004 publicerade Yi Liu och Serge Vaudenay en attack baserad på statistik som förutsatte att man visste dom 24 första av 2^{35} paket. Det krävdes ungefär 2^{40} operationer för att få fram nyckeln. Attacken förbättrades senare till att omfatta 2^{37} operationer av data uträknat i förväg och 2^{39} operationer för att söka efter nyckeln. 2005 kom Lu, Meier och Vaudenay med en attack baserad på känd klartext. De lyckades få fram nyckeln genom att veta om dom första 24 bitarna av $2^{23,8}$ paket och göra 2^{38} beräkningar för att få fram nyckeln.

4.7 Säkerhetsnivåer

Säkerhetsnivå ett innebär, som tidigare sagts, total frånvaro av ytterligare säkerhetsfinesser. Denna nivå är främst tänkt att användas när enheten varken lagrar, sänder eller tar emot data av känslig natur.

Vid säkerhetsnivå två klassas anslutningarna in i två olika klasser, betrodda och icke betrodda. Betrodda anslutningar är sådana med apparater som är parade tillsammans med den här apparaten, det vill säga enheter som godkänts av användaren som betrodda. Dessa parade enheter har full tillgång till alla tjänster. En icke betrodd bluetoothapparat är en sådan som inte har någon permanent relation, en parning, till vår apparat. Den har därför begränsad tillgång till tjänster och måste godkännas vid varje kontakt.

Säkerhetsnivå tre är den mest täta. En enhet med denna nivå tillåter ingen överföring till eller från enheter som inte är parade med den [2].

5. Säkerhetsbrister

Kända och mindre kända. Här redovisas både faktiska attacker och teorier om säkerhetsbrister.

5.1 Kända brister

Nedan följer exempel på de vanligaste problemen med säkerheten vid överföring via bluetooth. I huvudsak är det mobiltelefoner som är den attackerade enheten.

5.1.1 Bluejacking

Bluejacking är en typ av phishing-attack där den attackerande parten sänder ett visitkort eller liknande till enheter som upptäckts i närheten. Får personen svar sänds ett mer personligt meddelande tillsammans med en förfrågan om att bli tilllagd i listan med betrodda enheter. Är svaret jakande har attacken lyckats [4].

5.1.2 Bluesnarfing

Bluesnarfing är en typ av attack som syftar till att komma över information som finns lagrad på den angripna enheten. Attacken går till så att angriparen ansluter till offrets enhet via bluetooth vilket ger honom eller henne möjlighet att ladda ner filer såsom telefonbok, kalender och liknande. En skicklig angripare kan även ändra eller byta ut filerna i den angripna enheten.

För att genomföra en bluesnarfing-attack krävs mjukvara utvecklad för ändamålet. Numera kan man köra denna typ av mjukvara skriven i Java på en telefon med stöd för J2ME till skillnad mot tidigare när telefonerna på marknaden inte var lika avancerade. Då behövde man en dator för att kunna genomföra en attack, men i och med att dagens mobiltelefoner klarar av bluesnarfing har möjligheterna att genomföra attacken obemärkt ökat markant. Den i bluetooth inbyggda möjligheten att via protokollet OBEX utväxla visitkort och liknande är anledningen till att bluesnarfing går att genomföra. OBEX har en push-funktion som används för att skicka visitkort till andra enheter, men via denna funktion kan man även sicka en förfrågan, en get-request, där man efterfrågar filer med kända namn. Filer med kända namn kan i detta fall vara exempelvis telefonbok (telecom/pb.vcf) eller kalender (telecom/cal.vcs). De telefoner som inte kräver

autentiering när en annan enhet försöker genomföra en push-funktion är ett enklare mål, men även andra enheter kan attackeras.

För att en bluesnarfing-attack ska kunna genomföras bör enheten vara i synligt läge, men i teorin är denna typ av angrepp även möjlig att göra på dolda telefoner. För att komma åt en av dessa dolda telefoner måste mjukvaran komma åt telefonen via det 48-bitarsnamn som alla bluetooth-enheter har. Dessa namn är möjliga att knäcka genom brute-force, men det finns snabbare sätt. Kvalificerade gissningar är oftast bättre än brute-force och i detta fall är bra gissningar telefonens modellnummer eller modellnamn. Dessa enhetsbeteckningar utgör ofta hela eller delar av enhetens bluetoothnamn, speciellt när det gäller äldre modeller av mobiltelefoner. [11]

5.1.3 Bluebugging

Till skillnad från bluejacking och bluesnarfing kan angriparen helt ta över de attackerade enheten med bluebugging. Offret märker inte att enheten, ofta en telefon, har blivit attackerad och angriparen kan således ostört ringa samtal, skicka sms, läsa och ändra telefonbok, samtalslistor och andra filer samt mycket mer. Telefonen som tagits över kan också användas som en mikrofon för att ta upp och skicka vidare alla ljud i närheten. Möjlighet finns också att lyssna av eller till och med vidarekoppla inkommande samtal, samt mycket mer.

De flesta telefoner, speciellt nyare modeller, har skydd mot bluebugging men de enheter som saknar ett sådant skydd kan tas över helt utan fysisk kontakt. Proceduren är ganska lik den som beskrivits vid bluejacking då telefonen nås via en push-förfrågan. Exempel på enheter med dåligt slydd är några tidiga telefoner av märket Motorola, där varken autentiering eller PIN-kod krävs för push-funktionen. Om inkräktaren bryter kontakten med en sådan enhet mitt i autentieringsprocessen sparas kontaken som känd (trusted) och angriparen kan då ansluta till offrets headset. Via den kontakten skickas sedan alla kommandon som krävs för att kontrollera enheten. [11]

5.1.4 Knäcka PIN-koden

Yaniv Shaked och Avishai Wool har visat att det på mycket kort tid går att återskapa PIN-koden som används i en parningsprocess om man är tillräckligt nära för att avlyssna och spara hela parningsprocessen. Med en 4-siffrig PIN-kod tar det 0,3 sekunder att knäcka den om man använder en gammal Pentium III på 450 Mhz, används en Pentium 4 på 3 Ghz så tar det 0,06 sekunder.

För att kunna göra det här krävs det att man kan tjuvlyssna på parningsprocessen och verifieringsprocessen och spara alla meddelande som skickas mellan mastern och slaven. Efter att man har sparat alla meddelanden kan man använda sig av en brute-force algoritm för att söka reda på PIN-koden.

Attackeraren tar alla möjliga PIN-koder och eftersom han vet om IN_RAND och BD_ADDR, de sänds i klartext, så kan han köra algoritm E-22 med IN_RAND, BD_ADDR och de gissade PIN-koderna som indata. Han får då fram möjliga kandidater till K-init, han kan nu använda dessa kandidater för att dekryptera LK_RAND-A och LK_RAND-B eftersom de är krypterade genom XORing med K-init. LK_RAND-A och LK_RAND-B innehåller information som gör att attackeraren kan beräkna kandidater till länknyckeln K-ab. För att kontrollera om han har fått den korrekta K-ab nyckeln så använder han sin kandidat till K-ab och AU_RAND-A, som sänts i klartext, för att beräkna SRES som också sänt i klartext. Stämmer hans framräknade SRES med den skickade vet han att han har rätt PIN. Stämmer inte hans SRES med den skickade så valde han fel kandidat till PIN-koden och får då prova med en annan.

Den här attacken fungerar endast mot PIN-koder som är mindre än 64-bitar, färre än 8-siffror på basen 10, men vanligtvis är PIN-koderna 4-siffriga. Det dock en väldig skillnad i tidsåtgång för att räkna fram PIN-koden beroende på om den är 4-siffrig, det tar 0,063 sekunder, eller om den är 7-siffrig vilket tar ca 76 sekunder. Uträkningarna sker på en 3 Ghz P4:a. Efter att PIN-koden knäckts kan attackeraren dekryptera all trafik mellan de två bluetoothartefakterna.

Den här attacken kräver att man kan avlyssna hela parningsprocessen men om de två bluetoothartefakterna som ska kommunicera tidigare har parats så har dom sparat länknyckeln K-ab och då kan de skippa direkt till verifieringsfasen. För att då komma över datan som behövs för att kunna få fram PIN-koden så blir man tvungen att få de kommunicerande apparaterna att göra om parningsprocessen. Det finns tre metoder för att uppnå detta:

1. I verifieringsprocessen skickar mastern ett AU_RAND meddelande till slaven och förväntar sig ett SRES meddelande som svar, i bluetooth specifikationen finns det möjlighet för en bluetoothartefakt att glömma bort en länknyckel och då istället skicka LMP_not_accepted message och på så sätt få mastern att kasta länknyckeln och börja parningsprocessen på nytt. För att genomföra attacken krävs det att man skickar ett LMP_not_accepted till mastern vid rätt tidpunkt för att få den att kasta länknyckeln.
2. I början av verifieringsfasen så ska mastern skicka AU_RAND meddelande till slaven. Innan det meddelandet skickas så skickar attackeraren IN_RAND meddelande till slaven vilket får den att tro att mastern har tappat bort sin nyckel och parningsprocessen startas om.
3. Under verifieringsprocessen så skickar mastern AU_RAND till slaven och förväntar sig korrekt SRES meddelande som svar. Innan slaven hinner svara skickar attackeraren ett felaktigt SRES meddelande till mastern vilket får verifieringsprocessen att börja om, efter ett par sådana här misslyckade verifieringar pga av attackerarens meddelanden så kommer mastern

att deklarera att verifieringsprocessen är avbruten och parningsprocessen måste köras igen.

Om någon av de här attackerna lyckas så kommer den attackerade användaren att vara tvungen att slå in PIN-koden igen och en misstänksam användare kan då förstå att hans bluetoothartefakter är under attack. [10]

5.1.5 Generella säkerhetsbrister

Trots att tillverkaren är så säker på att protokollet inte innehåller några brister har ett flertal svaga punkter uppdagats på olika håll. Några av de vanligaste klagomålen redovisas nedan [8].

Det är vanligt att PIN-koden, alltså det som är säkerhetskoden för enheten, kortas ner till bara fyra siffror. Detta är tillverkarens val och en eftergift för användarens bekvämlighet, men inte desto mindre ett exempel på bristande säkerhet.

Nyckelgenereringen vid parningsprocessen är klen. Schemat tillåter användning av ett statiskt tal vilket sänker effektiviteten av verifieringen. Ett annat problem med verifieringen är att den bara kräver challenge från den ena parten, vilket inbjuder till attacker av typen man-in-the-middle.

Krypteringsalgoritmen och dess tillämpningar får sig också många kängor av kritiker. Den anses vara både för simpel och inte användas fullt ut. Trappe och Washington [3] ger en redogörelse för algoritmer och attacker.

6. Slutsatser

Resultatet av litteraturstudien redovisas nedan, tillsammans med våra egna reflektioner och åsikter.

6.1 Resultat

Ett generellt konstaterande är att väldigt många av de säkerhetsproblem som upptäckts vid överföring via bluetooth beror på felaktigt handahavande. En väldigt vanlig typ av attack är phishing i olika former, vilket innebär att användaren på ett eller annat sätt luras att dela med sig av koder, lägga till enheter och motsvarande. Det kan diskuteras om dessa problem är enbart användarens problem eller om tillverkaren av enheten borde skydda eller informera mer, men det går ändå inte att komma ifrån att dessa säkerhetshål är de oftast utnyttjade.

En annan variant är att hackers använder sig av bluetooth för överföring av maskar och elak kod. Detta bygger på att den attackerande parten använder sig av svagheter i implementationen av protokollet hos mottagarenheten. Denna typ av attacker rapporteras inte lika ofta som de phishing-baserade.

Trots att tillverkaren anser att protokollet bluetooth är totalt vattentätt och har en säkerhetsnivå som inte går att knäcka kan man ändå konstatera att så inte är fallet. En 128 bitars PIN-kod är kanske jobbig men absolut inte omöjlig att knäcka, och kryptering med åtta till 128 bitars nyckel kan garanterat brytas med lite god vilja och mycket tid.

6.2 Diskussion

Den uppfattning vi har efter detta projekt är att bluetooth har ett gediget säkerhetstänkande. Säkerheten har satts i fokus vid implementationen och man har kommit en bra bit på väg. Lite farligt är dock att anse att man är färdig och protokollet är säkert, det är trots allt en dynamisk process och säkerhetstänkandet måste få vara en stor del av allt utvecklingsarbete, inte bara det inledande.

Vidare anser vi att det faktum att så många attacker beror av felaktigt handahavande är oacceptabelt. Många användare vet förmodligen inte alls vilken risk de tar när de exempelvis traskar runt med sin felaktigt inställda mobiltelefon i fickan. Här tycker vi att tillverkarna av enheten har ett väldigt stort ansvar, både när det gäller att implementera säkerhet som standard och information om riskerna till användare som ändå vill sänka den förinställda säkerheten.

En reflektion är också om bluetooth verkligen var tänkt att användas vid så kritiska tjänster som bank- och kreditkortsaffärer. Faktum är att i de flesta av dagens mobiltelefoner kan man både läsa potentiellt hemlig mail och göra bankaffärer, och om användama dessutom inte vet att de har en vidöppen bluetooth-kanal i samma artefakt är det snararast förvånande att inte fler blivit utsatta för attacker.

Källor

[1] Bray, J och Sturman, CF *Bluetooth Connect without Cabels* (2001)
[2] Muller, N J *Bluetooth Demystified* (2000)
[3] Trappe, W och Washington, L *Introduction to Cryptography with Coding Theory* (2006)
[4] http://www.bluetooth.com (9-27 april)
[5] http://www.realtechnews.com/posts/4270 (13 april)
[6] http://gizmodo.com/gadgets/robots/echo-robot-teddy-bear-meets-women-for-you-232163.php (13 april)
[7] http://www.halfbakery.com/idea/Bluetooth_20Golf_20Trolley (13 april)
[8] http://www.tech-faq.com/bluetooth.shtml (26 april)
[9] http://www.wikipedia.org/wiki/E0_(cipher) (27 april)
[10] Shaked, Y och Wool, A *Cracking the bluetooth PIN*
[11] http://www.wirelessnetdesignline.com/showArticle.jhtml?articleID=192200279 (2 maj)
[12] www.bsi.de/english/publications/brosh/B05_bluetooth pdf (2 maj)
[13] Shaked, Y och Wool, A *Cryptanalysis of the Bluetooth E0 Cipher using OBDD's* (2006)