

Forbedret autentisering i Bluetooth

3/27/07

Ståle Andreas Kleppe
<stale.kleppe@student.uib.no>

Institutt for Informatikk
Universitetet i Bergen

- Hva er Bluetooth?
- Sikkerheten i Bluetooth.
- Svakheter i sikkerheten.
- Løsning
 - Teori
 - Praksis
 - Utfordringer
 - Resultat
- Oppsummering og konklusjon.

Hva er Bluetooth?

- Trådløs teknologi med lavt strømforbruk, kort rekkevidde og lave produksjonskostnader.
- Kan erstatte kabler mellom de fleste typer enheter.
- Over 1 milliard mennesker eier minst én Bluetooth-enhet.
 - Nærmest samtlige mobiltelefoner og bærbare datamaskiner leveres med Bluetooth.

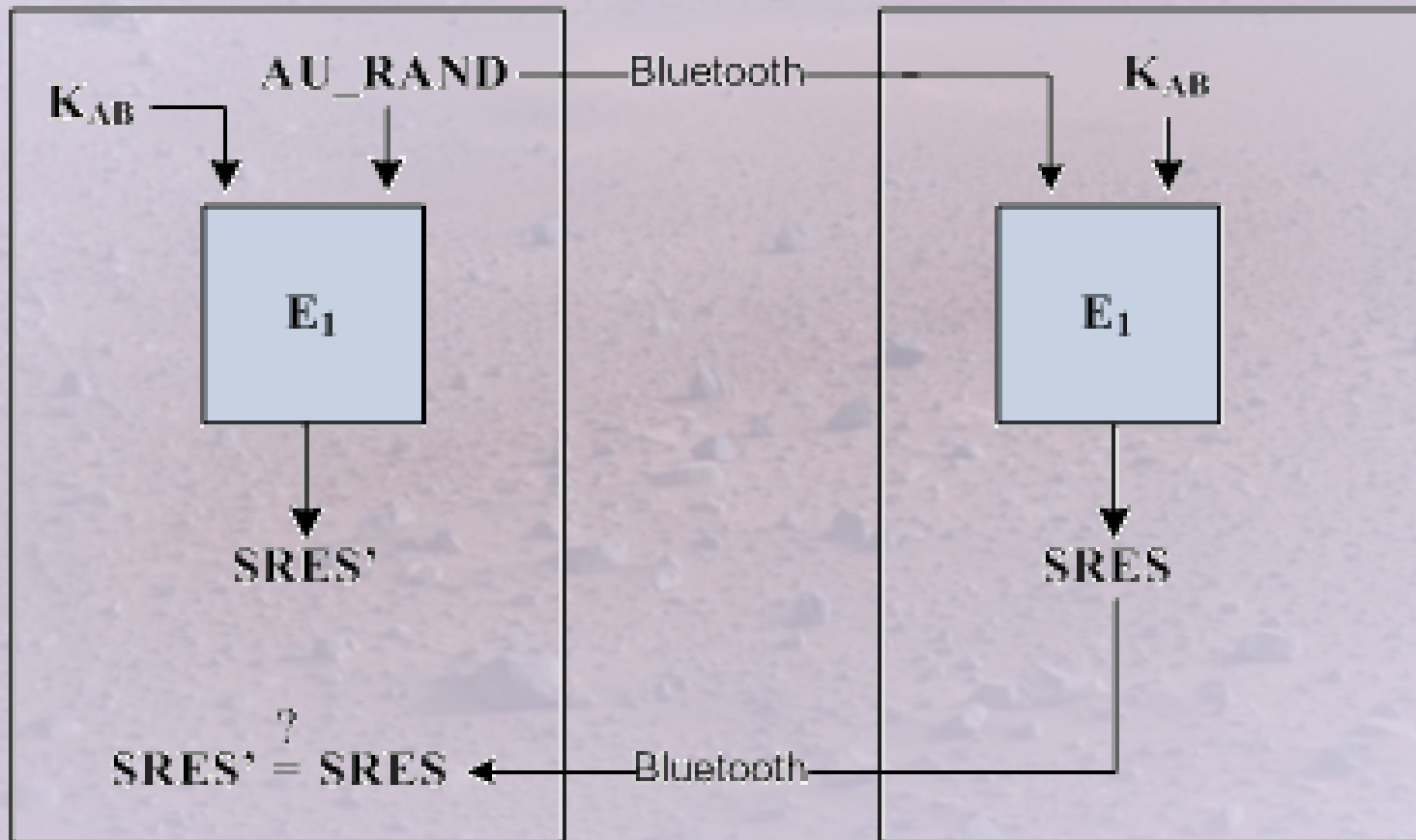
Sikkerheten i Bluetooth

- Nøkler
- “Pairing”
 - Nøkkelgenerering
 - PIN-kode
- Autentisering
- Kryptering

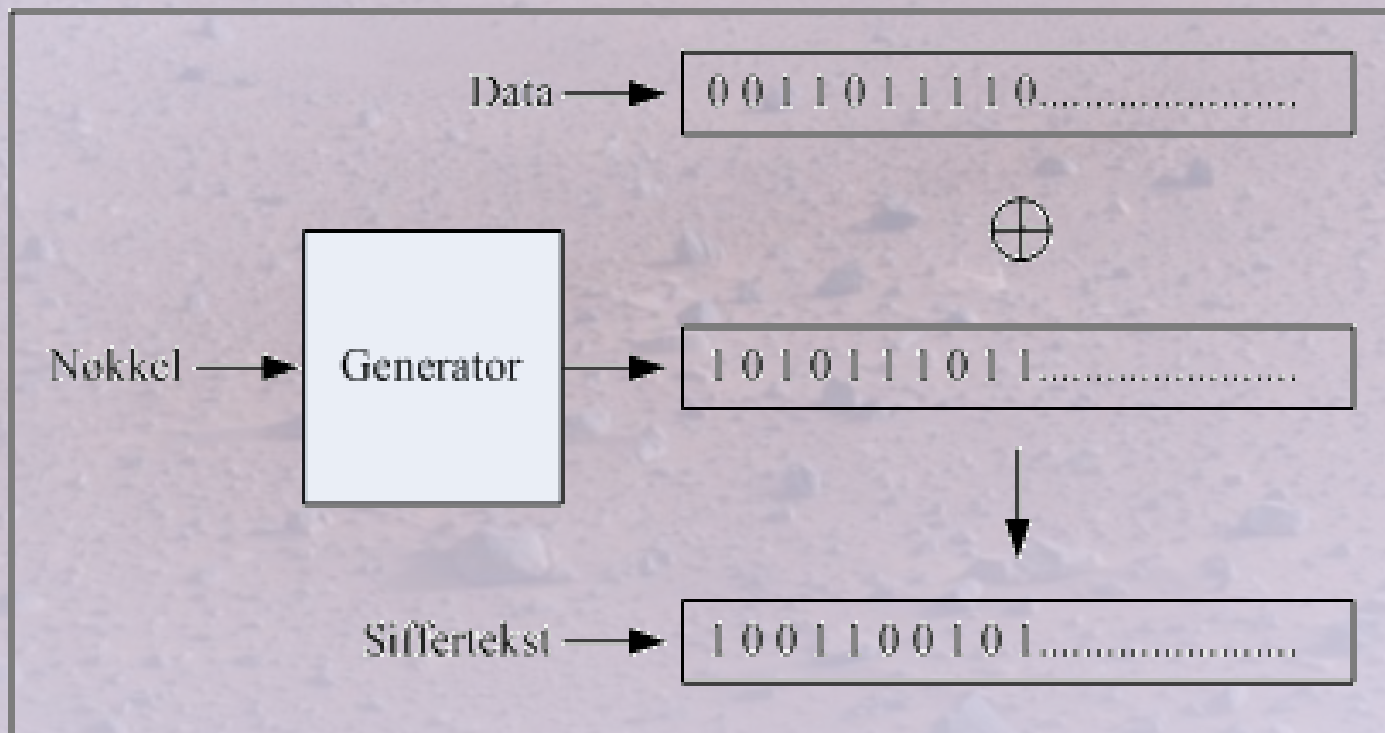
- Initialiseringsnøkkel K_{init}
 - Brukt under utveksling av autentiseringsnøkkel.
 - Midlertidig; slettes etter “pairing”-prosessen er over.
- Autentiseringsnøkkel K_{AB}
 - Brukes for å autentisere to enheter mot hverandre.
 - Semi-permanent; varer til enhetene ber om å generere en ny.
- Krypteringsnøkkel K_C
 - Brukes til å kryptere dataoverføringen.
 - Varer så lenge tilkoblingen eksisterer.

- Nøkkelgenereringen i Bluetooth.
 - Samme PIN-kode tastes in på begge enheter.
 - Utfra PIN-kode og noe annen data genereres initialiseringsnøkkel K_{init} .
 - Informasjon til autentiseringsnøkkel utveksles, beskyttet av K_{init} .
- Etter “pairing” kan to enheter opprette en sikker forbindelse.

Challenge/Response



- BT benytter en strømsiffer
 - Nøkkelstrømmen initialiseres av krypteringsnøkkel K_C



- PIN-koden eneste hemmelige i beregning av K_{init} .
 - All annen informasjon “offentlig” kjent.
 - “Offline”-angrep
- Svake PIN-koder, f.eks. 4-sifrede, er vanlig.
 - 10000 forskjellige PIN-koder kan gjennomløpes og testes raskt på en moderne datamaskin.
 - '0000' blir ofte brukt og er standard på de fleste Bluetooth hands-free-løsninger.

Angrep på PIN-koden

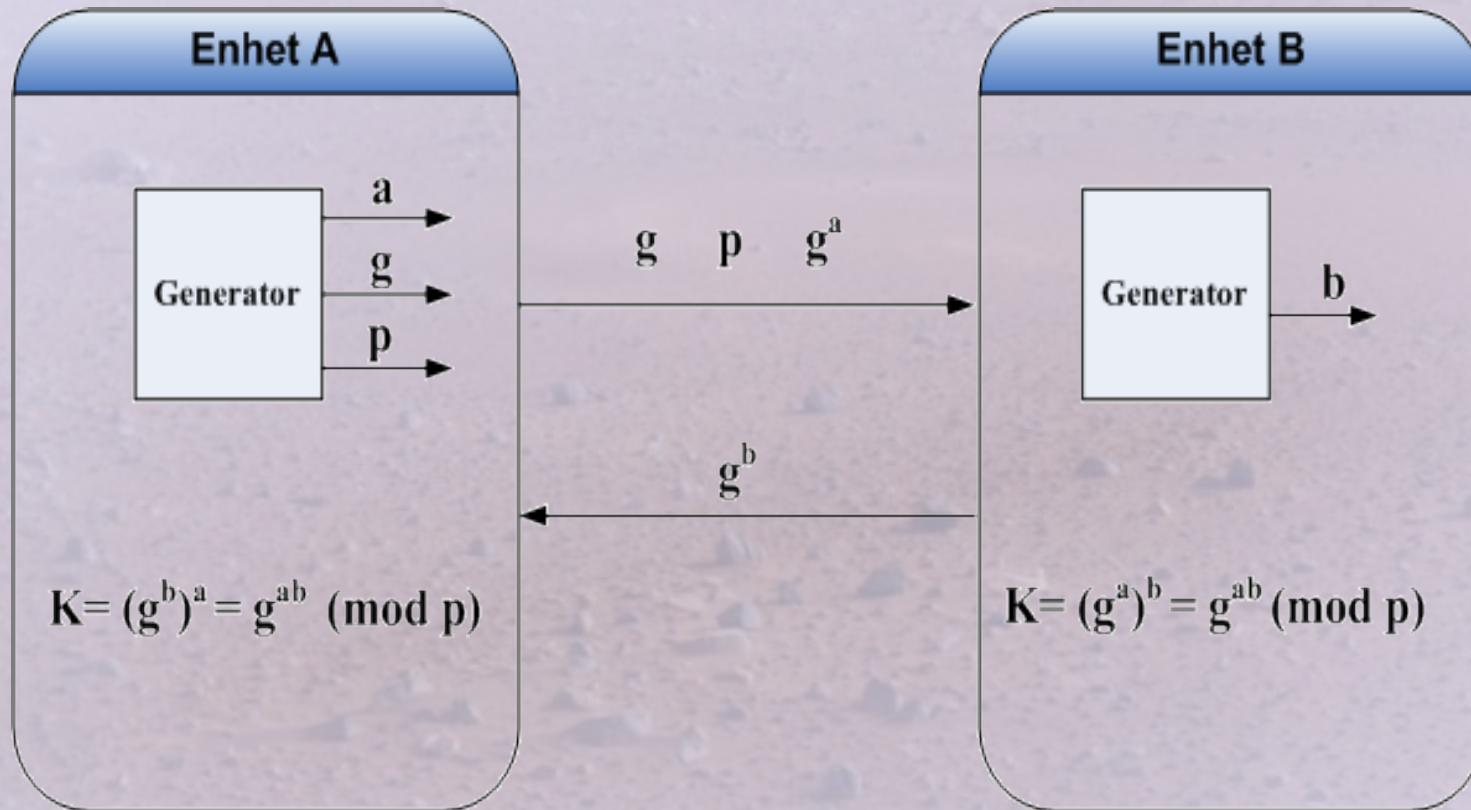
- Lytt på linken etter nødvendig informasjon.
- For hver mulige PIN-kode:
 - Generer K_{init}' og deretter K_{AB}' .
 - Sjekk om $K_{AB}' = K_{AB}$
 - Bruk informasjon fra autentiseringsprosessen.
 - Beregn en egen SRES' og sammenlign med SRES.
 - Er de lik, er PIN og K_{AB} funnet.
- Ved å først forsøke de mest sannsynlige PIN-koder, f.eks. 0000, 1111, osv. blir angrepet mer effektivt.

- “Sniffing” av Bluetooth-pakker.
 - Frontline Bluetooth Protocol Analyzer & Packet Sniffer.
 - <http://www.fte.com/products/FTS4BT-01.asp>
- Programmvare for angrepet.
 - http://www.nruns.com/_downloads/BTCrack.zip

- Med kjennskap til PIN-koden og dermed også K_{AB} , kan en angriper-enhet gi seg ut for å være en av de to som deler K_{AB} .
 - Oppskrift finnes her:
<http://www.digitalmunition.com/TheftOfLinkKey.txt>
- Alle tjenestene offeret tilbyr motparten, blir tilgjengelig for angriper:
 - Avtalebok
 - E-post
 - GSM-tilgang
 - Annet

- Diskret logaritme problemet.
 - Gitt $g^a \equiv (\text{mod } p)$, finn a .
 - Et antatt praktisk umulig problem.
- Diffie-Hellman nøkkelutveksling.
 - A velger hemmelig eksponent a , sender $g^a \pmod{p}$.
 - B velger hemmelig eksponent b , sender $g^b \pmod{p}$.
 - Partene beregner nøkkel $K \equiv (g^b)^a \equiv (g^a)^b \pmod{p}$.

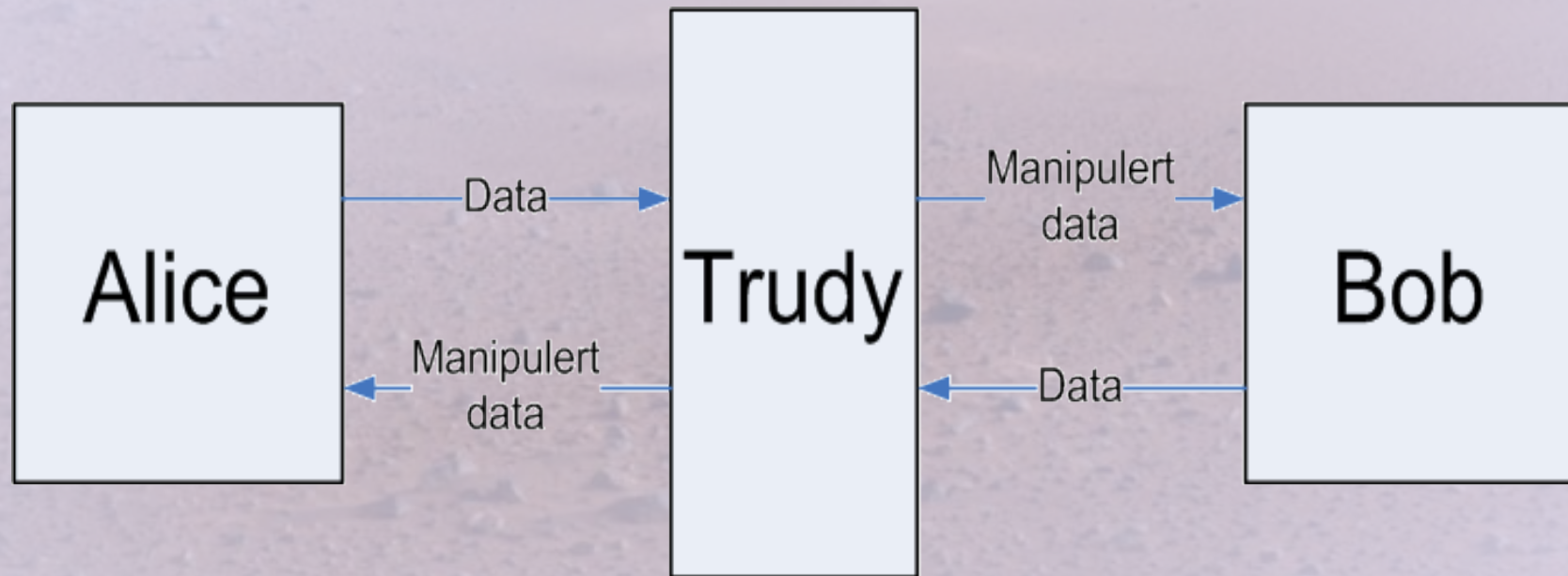
Diffie-Hellman



- Et “offline”-angrep kan i praksis IKKE utføres mot Diffie-Hellman.

Man in the Middle

- DH sårbar mot “Man in the middle”-angrep.



Diffie-Hellman og integritetssjekk

- Verifiser at g^a kommer fra riktig avsender.
 - Beregn $mac = MAC(K_{MAC}, g^a)$.
 - Vis mac og K_{MAC} på enhet A.
 - Bruker skriver inn mac og K_{MAC} på enhet B.
 - B sjekker om mac er lik egen beregning.

• I hardware

- Mest effektivt.
- Lite fleksibelt.

• I software

- Mindre effektivt.
- Fleksibelt; mulig å implementeres av 3. parts utviklere.

Implementasjon i software

- BlueZ – Linux' implementasjon av høyerelags BT-protokoller
 - Gir full kontroll over operasjoner i alle lag, f.eks. lagring av autentiseringsnøkkel.
- Implementert i C
 - Tilgang til alle protokoller og operasjoner

- Store heltall
 - FreeLIP:
<http://www.netsw.org/system/libs/math/freelip-1.1.tar.gz>
 - FreeLIP har det meste av funksjonalitet.
- Tilfeldige tall
 - Linux' tilfeldige kilde: /dev/random
- Primtall
 - Primtallstest

- Prototype fungerer som forventet.
 - Genererer og lagrer K_{AB} .
- Like brukervennlig som eksisterende løsning.
- Ytelse
 - Primtall på 256 bit: 2,5 sek.
 - Primtall på 512 bit: 7,8 sek.
- Kan bli for tungt for små mobile enheter.
 - Maskinvare-implementasjon vil sannsynligvis bedre ytelsen.
 - Bruk av elliptisk-krurve-kryptografi.

Oppsummering og konklusjon

- Sikkerheten i Bluetooth har en alvorlig svakhet.
 - Angrep på PIN-koden fullt mulig.
- En forbedret nøkkelgenerering er nødvendig.
- Diffie-Hellman fungerer tilfredsstillende, og kan med litt optimalisering erstatte eksisterende “pairing”-mekanisme.
- Ny versjon av Bluetooth blir snart offentliggjort.
 - En DH-variant implementert over elliptiske kurver er inkludert.

Spørsmål og kommentarer

???