Bluetooth Device Discovery and Hop Synchronization by the Eavesdropper

[1]Ahmad Ali Tabassam, [2]Stefan Heiss, [3]Michael Höing
[1,2]Dept. of Electrical and Information Engineering
Fachhochschule Lippe ünd Höter D-32657-Lemgo, Germany
[3]Dept. of Pre-Development and Electronic Development
Weidmüller Interface GmbH & Co.KG, D–32758 Detmold, Germany
ahmadali@cc.fh-luh.de, stefan.heiss@fh-luh.de, michael.Hoeing@weidmueller.de

Abstract — **Bluetooth radio system uses the Frequency Hopping Spread-Spectrum (FHSS) and Time Division Duplexing (TDD) for transmitting and receiving a packet at 79 different channels at 1,600 hop per/sec. The Bluetooth devices must be properly synchronized so that they can hopped together from channel to channel; this can be done by using the same channel set as well as the same hopping sequence within that channel set along with the time synchronized within hopping sequence.**

The Inquiry procedure is used to locate the Bluetooth devices in neighborhood, Page procedure is used to establish the connection for Bluetooth communication. This paper describes how to eavesdrop packets which can determine the pseudo-random seed for the inquiry and paging hopping sequence by scanning the inquiry and page frequencies, which is done by eavesdropping on the identity/control (FHS) packets that are exchanged during the inquiry procedure and page procedure. We can determine the pseudo-random seed for the channel hopping sequence of the piconet from the master's device address and its clock from the ongoing communication.

Index Terms — **Frequency Hopping Spread-Spectrum (FHSS), Time Division Duplexing (TDD), Inquiry Procedure, Page Procedure.**

I. Introduction

The Bluetooth [1,2] radio system operates in the Industrial-Scientific-Medical (ISM) band ranges from 2.400 GHz to 2.4835 GHz. The Bluetooth wireless communication link is divided into 79 RF channels spaced 1.0 MHz apart in the 83.5 MHz bandwidth. It uses Time Division Duplex (TDD) modulation and Frequency Hopping Spread-Spectrum scheme, in which the channel is represented by a pseudo random hopping sequence through the entire 79 RF frequencies at a rate of 1600 hops/sec and channel spacing of 1 MHz. In order for the devices to communicate with each other in Bluetooth networks, they must be properly synchronized so that they can hop together from channel to channel and they must use the same:

- Channel Hopping Sequence.
- Phase in Hopping Sequence.
- Channel Access Code.

To eavesdrop on the communication, we need to know what seed is used for the pseudo-random hopping sequence. For devices in the inquiry substate; the seed is derived from the inquiring device's own clock and general inquiry access code (GIAC) see Table 1. For the devices in the page substate; the seed is derived from the clock and Bluetooth device address of the paged device (slave). In the connection substate and the channel hopping sequence of the piconet, the seed is derived from the Bluetooth device address and clock of the master device.

Table I
CHANNEL ACCESS CODES

Sr.	Access Code	LAP	Code length
1	Channel Access Code (CAC)	Master	72
2	Device Access Code (DAC)	Paged device	68/72
3	Inquiry Access Code (IAC)	Reserved	68/72

We can determine the pseudo-random [3] seed for the paging hopping sequence by scanning the inquiry frequencies and eavesdropping on the page response messages. We can derive the pseudo-random seed for the channel hopping sequence of the piconet as the master device reveals its identity and clock when sending its Frequency Hop Sequence (FHS) packet during the page response messages. During this process, the slave device calculates the clock offset between its clock and that of the master device. With this offset and the master's address, the slave device is able to synchronize its hop sequence with that of the master device.

II. Background

Data on the piconet channel is conveyed in packets (Fig. 1). Each general Bluetooth packet [2] includes 68/72-bit access code, 54-bit header, and a payload. The access code and header are of fixed size: 68/72-bits and 54-bits, respectively. The payload can range from zero to a maximum of 2745 bits. The Access Code includes a preamble, a sync word, and possibly a trailer. The preamble is a fixed zero-one pattern of 4 symbols either 1010 or 0101, depending on whether the LSB of the following sync word is 1 or 0, respectively. The sync word is a 64-bit code word derived from a *24-bit lower address part (LAP)* of the Master device and the trailer is either 1010 or 0101 depending on whether the MSB of the sync word is 0 or 1, respectively.

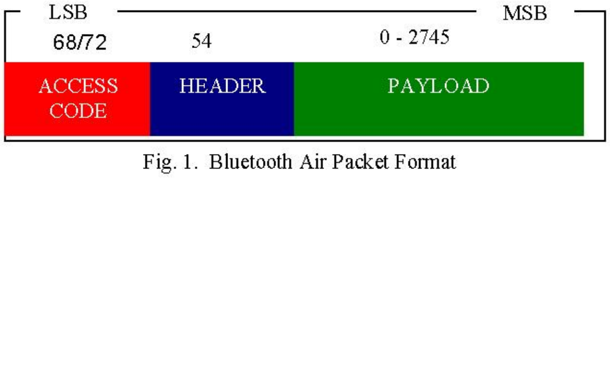
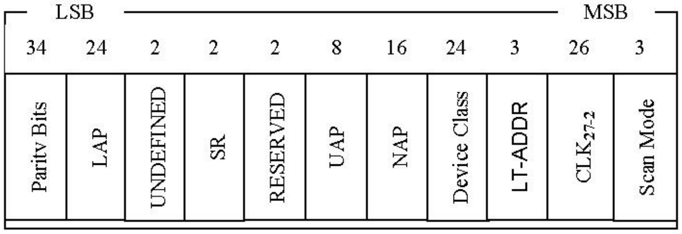


Fig. 1. Bluetooth Air Packet Format

1-4244-1494-6/07/$25.00 © 2007 IEEE



34	24	2	2	2	8	16	24	3	26	3
Parity Bits	LAP	UNDEFINED	SR	RESERVED	UAP	NAP	Device Class	LT-ADDR	CLK_{27-2}	Scan Mode

Fig. 2. Bluetooth FHS Packet Format

The Frequency Hop Synchronization (FHS) packet (Fig. 2) is a special control packet. It consists of the channel access code, packet header and payload; that contains, among the other things, the Bluetooth device address and the clock of the sender. The payload contains 144 information bits and 16-bit CRC. The FHS packet is used in inquiry response (by slave) and page response (by master). The FHS packet contains the real-time clock information. It is used for frequency hop synchronization before the piconet channel has been established.

The frequency hopping algorithm [4, 5] determines the channel hopping sequence from the Bluetooth Device Address ($BD\_ADDR_{(4bit\ UAP/24bit\ LAP)}$) of the Master device, the Phase in the hopping sequence is determined by the Clock ($CLK_{27-1bit}$) of the Master device and the Channel Access Code (CAC), is calculated from Lower Address Part ($LAP_{24\text{-bit}}$) of Master's Bluetooth Device Address (*BD_ADDR)*. The Bluetooth device address: BD_ADDR is a unique 48-bit device address. It is divided into a 24-bit LAP (Lower Address Part), a 16-bit NAP (Non-significant Address Part) and an 8-bit UAP (Upper Address Part). The Bluetooth clock: CLK is a 28-bit counter with 0.3125 ms resolution. This 28-bit clock overruns every 2^{28} × 0.3125 ms: approximately every 23 hours, 18 minutes, 6.08 seconds.

III. Inquiry porceduure

In Bluetooth networks the connections are managed by the link manager using the device discovery protocol at the baseband layer. The device (master) that initiates an inquiry procedure, runs the Inquiry protocol and a device willing to be discovered (slave) runs the Inquiry scan protocol. The inquiry procedure (Fig. 3) starts when the master device transmits an Identity *(ID1) packet* with an inquiry access code, which is a code common to all Bluetooth devices. Meanwhile, the slave device in the Standby state periodically enters the Inquiry scan state to search for IAC messages on the wake-up carriers, but only a time window of at least 11.25ms. When the slave device receives the second Identity (ID2) packet, it enters the Inquiry response state and exactly one slot later (625 μs) it returns a *Frequency Hop Synchronization (FHS) packet* which contains its device address and timing information for one-to-one correspondence to the current inquiry hopping sequence. FHS packet contains the Non-significant Address Part (NAP) of Bluetooth Device Address $BD\_ADDR_{47\text{-}32}$, Upper Address Part (UAP) $BD\_ADDR_{31\text{-}24}$, Lower Address Part (LAP) $BD\_ADDR_{23\text{-}0}$ and $CLK_{27\text{-}2}$. When the master device receives the FHS packet from the slave device, it uses the slave's address for Device Access Code (DAC) which is used during the page substates and it is derived from 24-bit Lower Address Part (LAP) of the slave device.

Table III
INQUIRY MESSAGES

Sr.	Packet	Direction	Hop Sequence	Access Code
1	ID (IAC)	Master to slave	Inquiry	Inquiry
2	FHS	Slave to master	Inquiry Resp.	Inquiry

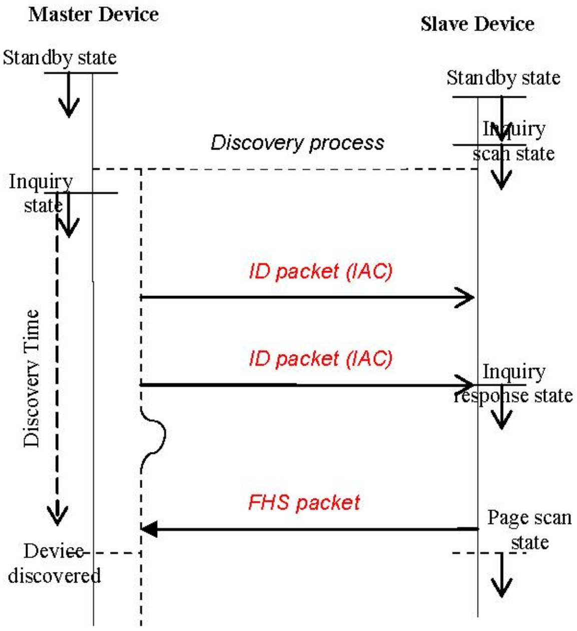


Fig. 3. Inquiry Procedure

The minimum time for the Inquiry Procedure is 2 slots (1.25ms) if the Master device sends the inquiry message: ID packet to first time slot and the Slave device responds with an FHS packet to the master's inquiry in the next time slot. So in total, 2 slots are needed.

IV. Page procedure

Paging procedure (Fig. 4) is used to establish a piconet connection. Only the Bluetooth device address is required to establish a connection. The device (master) that initiates a connection carries out a page procedure in the page substate. It may already know the BD_ADDR of the paged device or retrieve it from the inquiry procedure. It sends Identity (ID1) packet with Device Access Code (DAC) of the slave to be paged in the page substate. Meanwhile, the slave device in the Standby state periodically enters the Page scan state to search for DAC messages on the wake-up carriers, but only a time window of at least 11.25ms. When the slave device receives Identity (ID1) packet with its DAC from master device, the slave device enters the slave page response substate. It sends back a page response consisting of its Identity (ID2) packet which contains its DAC, at the frequency for the next time slot from the one in which page message was received. When the master device receives the ID (ID2) packet, it goes into the page master response substate and sends a *Frequency Hop Synchronization* (FHS) packet to the slave device, when the slave device receives the FHS packet it sends the ID (ID3) packet to the master device, the slave goes into the connection sub state and waits for the POLL packet to arrive from the master device, when the master device receives the ID (ID3) packet, it enters in the connection sub state and sends a POLL packet to the slave device. On reception of the POLL packet, the slave device sends a NULL packet as acknowledgment. When the master device receives the packet, a connection has been established.

Table III
PAGE MESSAGES

Sr.	Packet	Direction	Hop Sequence	Access Code
1	ID (ID1)	Master to slave	Page	Slave
2	ID (ID2)	Slave to master	Page response	Slave
3	FHS	Master to slave	Page	Slave
4	ID (ID3)	Slave to master	Page response	Slave
5	POLL	Master to slave	Channel	Master
6	NULL	Slave to master	Channel	Master

If the POLL packet is not received by the slave, or the acknowledgment NULL packet is not received by the master, the master and the slave shall return to the previous substate (page and page scan substates), respectively.

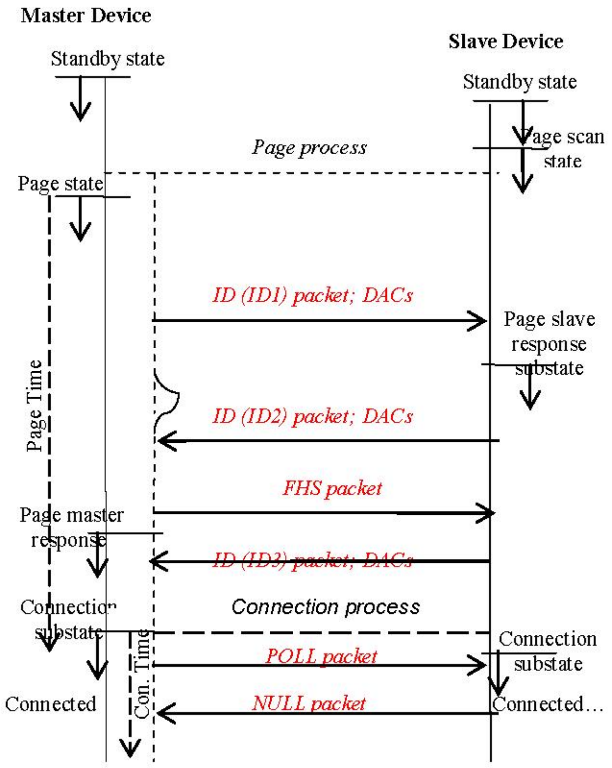


Fig. 4. Page Procedure

The minimum time for the Page Procedure is 4 slots (2.50ms), if Master device sends the page message: ID1 packet at first time slot, Slave device responses with ID2 packet for master's page response in the next slot and then master transmits FHS packet in next time slot to the slave. Finally in the next slot the slave responses with ID3 packet. So, in total 4 slots are needed.

V. Bluetooth Frequency Hopping

Bluetooth uses the Spread-Spectrum Frequency Hopping scheme in which channel is represented by a pseudo random hopping sequence through the entire 79 RF frequencies at the rate of 1600 hops per second and channel spacing is 1 MHz as shown. In order for the Bluetooth network devices to communicate with each other, they must be properly synchronized so that they can hop together from channel to channel. This means that the devices must use.

- Same channel set.
- Same hopping sequence within that channel set.
- Time synchronized within that hopping sequence.

These depend on four variables to calculate the next frequency BD_ADDR $_{\text{(4-bit UAP/24-bit LAP)}}$, CLK $_{\text{27-0bit}}$, Sequence Selection and the AFH_Channel Map. Hop Sequence Generator Block Diagram is detailed in Figure 5.

The selection scheme chooses a segment of 32 hop frequencies spanning about 64 MHz for the inquiry, inquiry response, page, page response hopping sequences, and visits these hops in a pseudo-random order. When the basic channel hopping sequence is selected, the output constitutes a pseudo-random sequence that slides through the 79 Hop channels. When the adapted channel hopping sequence is used, the pseudo-random sequence contains only frequencies that are in the RF channel set defined by the AFH_channel_map input.

VI. Implementation

Six types of hopping sequence (Table IV) are defined − five for the Basic frequency hop system and one for an Adapted frequency hop system. Basic hop sequences are: Inquiry hopping sequence, Inquiry response hopping sequence, Page hopping sequence, Page response hopping sequence, and Basic channel hopping sequence, while Adapted hop sequence is: Adapted channel hopping sequence.

Table IV
FREQUENCY HOPPING SEQUENCES

Nr.	Hopping Sequence	Wake-up Freq.
1	Inquiry hopping sequence	32
2	Inquiry response hopping sequence	32
3	Page hopping sequence	32
4	Page response hopping sequence	32
5	Basic channel hopping sequence	79
6	Adapted channel hopping sequence	≤ 79

In other words, four hopping sequences in total of six are used for inquiry and paging while the other two are used for piconet channel. Bluetooth device address of the master (BD_ADDR) is used when the basic or adapted channel hopping sequences are selected. BD_ADDR of the paged device (slave) is used when the page and page response hopping sequences are selected. For the inquiry and inquiry response hopping sequences, the 4-bitUAP/24-bitLAP corresponding to the GIAC shall be used. For the adapted channel hopping sequence, the *AFH_channel_map* is an additional input that indicates which channels shall be *used* and which shall be unused.

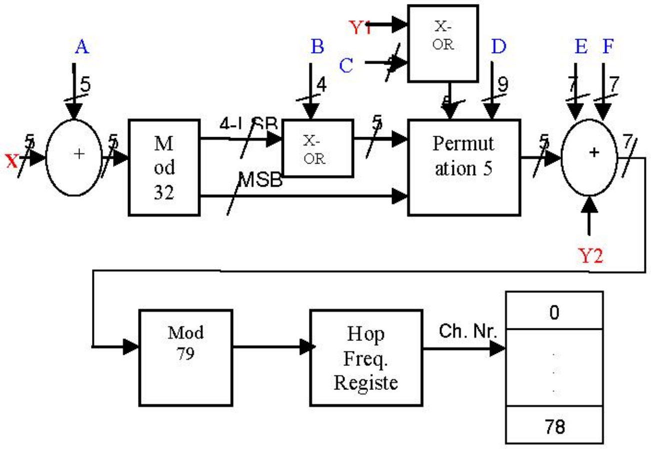


Fig. 5. Basic Hop Selection Kernel

The Hop Selection Kernel [5] addresses a register containing the RF channel indices. The inputs A to D determine the ordering within the segment, the inputs E and F determine the mapping onto the hop frequencies. The X input determines the phase in the 32-hop segment, whereas Y1 and Y2 select the time slots between master-to-slave and slave-to-master.

Table V
HOP SELECTION KERNEL INPUTS

Inputs	Values
A	$BD\_ADDR_{27-23} \oplus CLK_{25-21}$
B	BD_ADDR_{22-19}
C	$BD\_ADDR_{(8,6\ 4,2,0)} \oplus CLK_{20-16}$
D	$BD\_ADDR_{18-10} \oplus CLK_{15-7}$
E	$BD\_ADDR_{(13,11,9,7,5,3,1)}$
F	$(16 * CLK_{27-7})$ *Mod* 79
X	CLK_{6-2}
Y1	CLK_1
Y2	$32 * CLK_1$

VII. Laboratory experiments

For this research work, to find different types of security attacks against Bluetooth, Bluetooth research laboratory environment *"Labor für Technische Informatik"* is developed for demonstrating Bluetooth security attacks in practice. The laboratory environment consists of the following equipments:

A. Frontline – FTS4BT Bluetooth Protocol Analyzer & Packet Sniffer.

The Bluetooth Protocol Analyzer & Packet Sniffer [6] supports Bluetooth v2.0 + ERD. It finds the Bluetooth device address BD_ADDR of discoverable devices. It operates either in independent mode (sniffer) or Piconet (master/slave) modes. It has three sniffing modes: air, serial HCI, and virtual which allows the developer maximum testing and debugging flexibilities. It also extracts audio into WAV files for playback and analysis. It can only capture, decode and display the data export into CSV and other formats, but it can not insert any data in the piconet communication, however certain analyzer can do this such as LeCroy's CATC BTTracer/Trainer [7]. The FTS4BT Bluetooth Protocol Analyzer & Packet Sniffer must know the Bluetooth device addresses for the hopping sequence synchronization.

Frontline FTS4BT Bluetooth Protocol Analyzer & Packet Sniffer has only three synchronization modes however certain analyzers have more synchronization modes such as Mobiwave Bluetooth Protocol Analyzer BPA-12 [8] has Synchronizing via fake connection, Synchronizing via syncing to the slave (Inquiry Sync, Page Sync and Page & Hold Sync) and Synchronizing via syncing to the master.

1) Slave Inquiry: Inquire the slave device to learn its clock. Wait for the master to page the slave. It performs an inquiry of the slave and determines its clock. The slave must be discoverable.

2) Master Inquiry: Inquire the master device to learn its clock. It performs an inquiry of the master device to determine its clock. It is possible to synchronize to a master's clock before or after a baseband connection is made. The master must be discoverable. To listen the on going communication, eavesdropper must know the Bluetooth address of the master and its clock to find the channel hopping sequence.

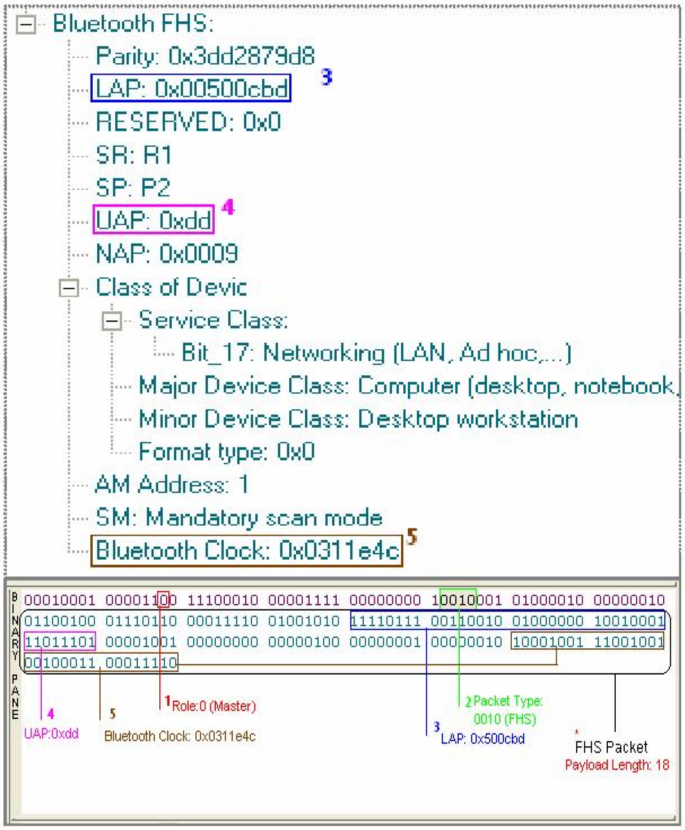
3) Slave Page: Page the slave device to learn its clock. Wait for the master to page the slave. The slave page option allows FTS4BT to learn the clock of a slave device that is not discoverable. If we know the slave device BD-ADDR, then by using slave page mode, we can learn the clock information during the paging process and will be synchronized to the undiscoverable slave's clock.

B. Bluetooth Environment

The Bluetooth system provides a point-to-point connection or a point-to-multipoint connection; two or more devices sharing the same physical channel form a *piconet.* One Bluetooth device acts as the master of the piconet, whereas the other device(s) act as slave(s), so, for real time Bluetooth communication we have the piconet environment, 4 PCs (two Linux Systems and two Microsoft Windows Xp Systems) and one laptop, PCs are connected with each other via Bluetooth. The Toshiba Satellite Pro laptop has the FTS4BT Bluetooth Protocol Analyzer & Packet Sniffer. It works as an eavesdropper on Bluetooth wireless communication network for HOP synchronization to listen the ongoing communication.

C. Bluetooth Softwares

KBTserailchat (client & server), BTChat (Bluetooth Chat client for Linux) and BTChatd (Bluetooth Chat server for Linux) are running on Linux Systems (Fedora Core 6). BTChatJava (Bluetooth chat client & server) and Hyper-terminal are running on Microsoft Windows XP Systems.



```
⊟ Bluetooth FHS:
    Parity: 0x3dd2879d8
    LAP: 0x00500cbd
    RESERVED: 0x0
    SR: R1
    SP: P2
    UAP: 0xdd
    NAP: 0x0009
  ⊟ Class of Devic
    ⊟ Service Class:
        Bit_17: Networking (LAN, Ad hoc,...)
      Major Device Class: Computer (desktop, notebook,
      Minor Device Class: Desktop workstation
      Format type: 0x0
    AM Address: 1
    SM: Mandatory scan mode
    Bluetooth Clock: 0x0311e4c
```

```
00010001 00001100 11100010 00001111 00000000 10010001 01000010 00000010
01100100 01110110 00011110 01001010 11110111 00110010 01000000 10010001
11011101 00001001 00000000 00000100 00000001 00000010 10001001 11001001
00100011 00011110
```

Fig. 6. FHS Packet Sniffed by FTS4BT

VIII. Conclusion

To follow the communication as the eavesdropper, we need to know the Bluetooth address (BD_ADDR_LAP) and clock of the master device (Bluetooth Clock). In the paging procedure from FHS packet, the pseudo-random seed is determined by the clock and Bluetooth device address of the master for channel hop sequence synchronization. So we have all the information from the FHS packet (Fig. 6) for channel synchronization; the detail is given below.

- *Device Role: 0x00;* Master device.
- *Packet Type: 0x0010;* FHS Packet.
- *Lower Address Part (LAP):* 0x500CBD.
- *Upper Address Part (UAP):* 0x0DD.
- *Bluetooth Clock:* 0x0311E4C.

The FHS packet contains all information to construct the channel access code without any mathematical derivation from the master's Bluetooth device address and its clock.

For ongoing communication, (without FHS packet) every packet access code has the 24-bit LAP while the 4-bit UAP is determined by Brute-Force BD ADDR Scanning [9] and Clock bits are recovered from Frequency Hop because the Bluetooth communication channel is divided into time slots. The time slots are numbered according to the Bluetooth clock of the piconet master. The slot numbering ranges from 0 to 228-1 with a cycle length approximately 23 hours, 18 minutes, 6.08 seconds. Each of the 79 Channels is equi-probable because FH sequence visits each carrier with equal probability. We want to find the probability that after k rounds, only one input is left and all of the others 228 −1 possible clock values are discarded at different rounds. Each of them have the probability (1/79)k. Assuming independence of Bluetooth master's device address BD_ADDR, and independence between the outcomes of different inputs, the probability is

$$(1+x)^n = (1-(\frac{1}{79})^k)^{2^{28}-1} \qquad (1)$$

As exponent is large and x is small with respect to 1, we can use the binomial expansion.

$$(1+x)^n = 1+\frac{nx}{1!}+\frac{n(n-1)x^2}{2!}+\ldots \qquad (2)$$

For $k = 6$, we have the approximation 1/79 to $1/2^6$.

Acknowledgment

The authors would like to thank the reviewers for their helpful comments. We would also like to thank Prof. Stefan Witte for providing the hardware.

References

[1]. DZUNG et al: "Security for Industrial Communications Systems," in *Proc IEEE vol. 93, no. 6*, June 2005.

[2]. Bluetooth Special Interest Group, "The Bluetooth Specification, Core 2.1+ ERD vol. 0," July 26, 2007.

[3]. M. Jakobsson and S. Wetzel, "Security Weaknesses in Bluetooth," *Proc. Cryptographer's Track at RSA Conf. (CT-RSA 2001)*, Laboratory Notes in Computer Science 2020. Berlin, Germany: Springer, 2001.

[4]. Robert Morrow, *Bluetooth Operation and Use.* McGraw-Hill Press, New York, 2002.

[5]. Jason Ballagh, "Bluetooth Frequency Selection Kernel Impact on Inter-Piconet Interference," *Proc. of Virginia Polytechnic Institute and State University*, Blacksburg Virginia US, April 2003.

[6]. Frontline – "FTS4BT Bluetooth Protocol Analyzer & Packet Sniffer," http://www.fte.com/products/FTS4BT-08.asp

[7]. LeCroy – "BTTracer/Trainer *Bluetooth Protocol Analyzers,"* CATC Protocol Solutions Group, LeCroy.

[8]. Mobiwave – "BPA-D12 Bluetooth Protocol Analyzer," http://www.mobiwave.com/bpa-d12.html

[9]. K.M.J. Haataja, "Two Practical Attacks against Bluetooth Security using New Enhanced Implementations of Security Analysis Tools," *Proc. of IASTED, Communication Network and Information Security,* UC Berkeley California 2007.