BLUETOOTH SECURITY

Antnan Giousouf
antnan.giousouf@rub.de

Instructor: Kerstin Lemke

Communication Security Department
Ruhr University, Bochum

CONTENTS

WHAT IS BLUETOOTH?

Bluetooth is a wireless radio specification designed to replace cables as the medium for data and voice signals between electronic devices. The specification is defined by the Bluetooth Special Interest Group (SIG) which is made up of over 1000 manufacturers. Intended primarily for mobile devices, Bluetooth's design sets a high priority on small size, low power consumption, and low costs. The Bluetooth specification seeks to simplify communication between electronic devices by automating the connection process.

Gartner research estimates that 161 million Bluetooth devices will be shipped in 2003 and this number will rise to 362 million in 2004. Bluetooth can currently be found in devices such as laptop computers, cellular phones, PDA's, headsets, printers, computer keyboards and mice, as well as digital cameras, and other consumer electronic devices. [1]

Brief History

The original architect for Bluetooth, named after the 10th century Danish king Harald Bluetooth, was Ericsson Mobile Communication. In 1998, IBM, Intel, Nokia, and Toshiba formed the Bluetooth SIG, which serves as the governing body of the specification.

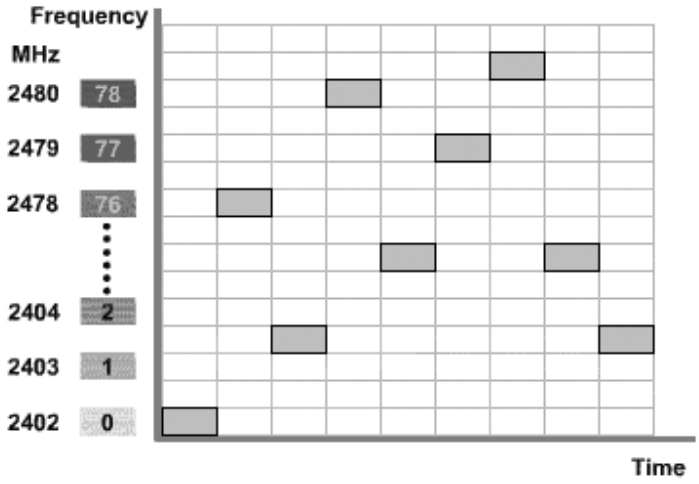
The SIG began as a means to monitor the development of the radio technology and the creation of a global and open standard. Today more than 1000 organizations are part of the Bluetooth SIG, comprising leaders in the telecommunications and computing industries that drive development and promotion of Bluetooth technology.

Bluetooth was originally designed primarily as a cable replacement protocol for wireless communications. However, SIG members plan to develop a broad range of Bluetooth-enabled consumer devices to enhance wireless connectivity.

Bluetooth is now standardized within the IEEE 802.15 Personal Area Network (PAN) Working Group that formed in early 1999. [2]

HOW DOES BLUETOOTH WORK?

Bluetooth is designed to operate in the unlicensed 2.4GHz Industrial, Scientific, and Medical application (ISM) frequency band. This frequency is already used by some other devices such as microwave ovens, baby monitors, cordless telephones, and 802.11b/g wireless networking devices In order to avoid interference from these devices; Bluetooth uses a technology called frequency hopping spread spectrum (FHSS). Spread spectrum frequency hopping changes the transmission frequency up to 1600 times per second across 79 different frequencies with channel spacing of 1 MHz [1].



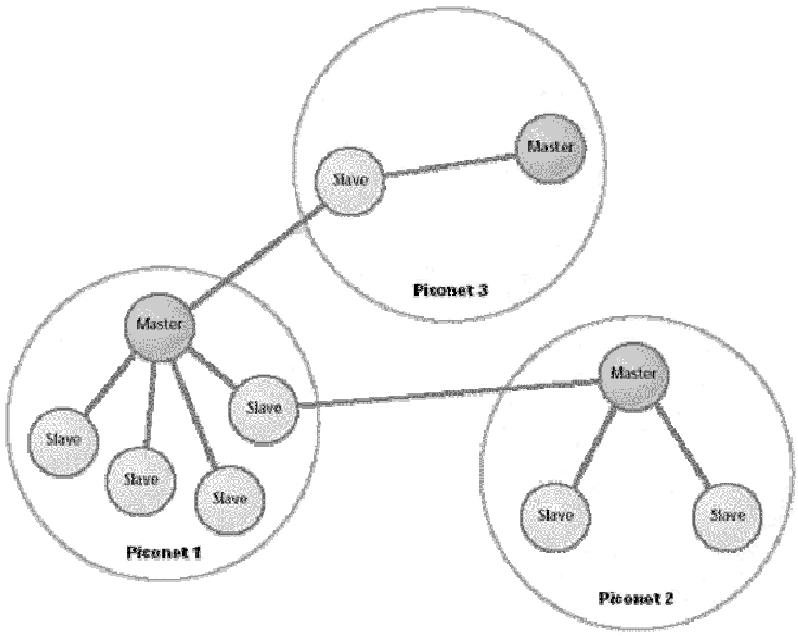
A sample FHSS distribution

One channel is used in 625 microseconds followed by a hop in a pseudo-random order to another channel for another 625 microsecond transmission; this process is repeated continuously. Bluetooth technology permits transmission speeds of up to 1 Mbps and achieves a throughput of approximately 720 kbps. Although the data rates are low compared to those of 802.11 wireless LANs, it is still three to eight times the average speed of parallel and serial ports, respectively. [2]

There are three classes of Bluetooth devices, according to the power they use and the range they have.

Type	Power	Power Level	Operating Range
Class 1 Devices	High	100 mW (20 dBm)	Up to 100 meters
Class 2 Devices	Medium	2.5 mW (4 dBm)	Up to 10 meters
Class 3 Devices	Low	1 mW (0 dBm)	0.1–10 meters

Bluetooth devices form ad hoc networks, called piconets. In these piconets, one of the Bluetooth devices acts as a master and the others are slaves. The master sets the frequency-hopping behavior of the piconet. It is also possible to connect up to 10 piconets to each other to form so-called scatternets. [3]



Sample piconets in a scatternet

Bluetooth devices automatically attempt to communicate whenever one device comes within range of another. Bluetooth devices discover each other and initiate communication via inquiry and page transmissions. Bluetooth devices have the ability to form ad hoc networks. The topology of these networks is both temporary and random. An ad hoc network of two or more Bluetooth devices is called a piconet.

When two Bluetooth devices initiate a connection, they automatically determine if one device needs to control the other. Generally, the device that initiates the communication assumes the role of master and exercises certain controls over the other members of the piconet which are known as slaves. Upon establishing a piconet, the slave devices synchronize their frequency hopping sequence and system clock with that of the master in order to maintain their connection. A master device can have up to seven slaves. A slave in one piconet can also be the master in another, thus allowing piconets to overlap and interact forming what is known as a scatternet.

Security Objectives

Bluetooth claims the following security objectives.

Access Control

Authentication (Link Layer)

Ensuring that a device possesses authentication secrets.

Authorization (Application Layer)

Ensuing hat only allowed services are granted.

Confidentiality

Ensuring that information exchanged cannot be disclosed.

Integrity

Ensuring that modified data is detected.

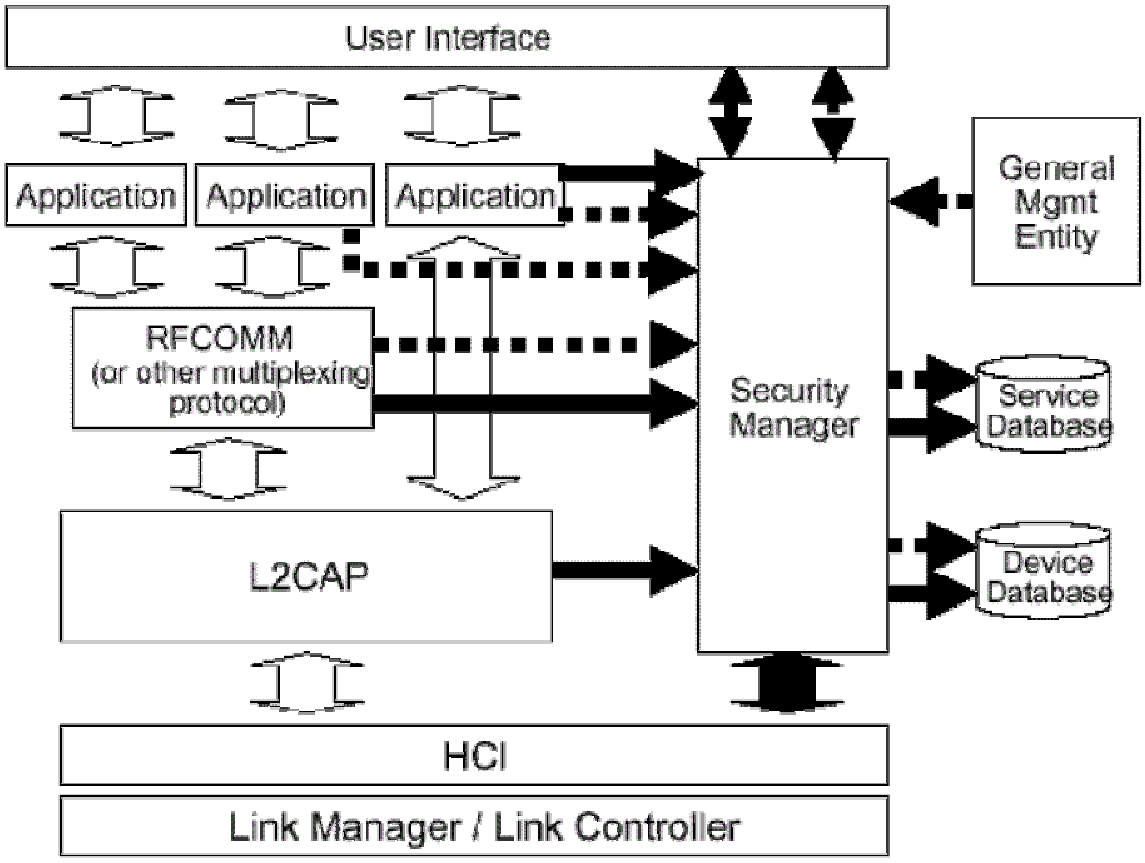
SECURITY OVERVIEW

Bluetooth uses a symmetric encryption scheme; therefore decryption is performed in exactly the same way using the same key as used for encryption and the secret keys must be exchanged in advance.

Four different entities are used for maintaining security at the link layer.

- **A Bluetooth device address, BD_ADDR (48 bits)**
- **A secret authentication key, the link key (128 bits)**
- **A secret encryption key (8-128 bits):** The encryption key is derived from the authentication key. Each time encryption is activated; a new encryption key shall be generated.
- **A pseudo-random number, RAND (128 bits):** It will be regenerated for each new transaction. Each device has a pseudo-random number generator to generate pseudo-random numbers.

The link layer is transparent to the security controls imposed by the application layers. Thus it is possible to enforce user-based authentication and fine-grained access control within the Bluetooth security framework.

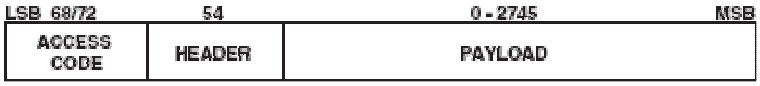


Bluetooth security architecture

INTEGRITY CHECK

The packets are checked for errors or wrong delivery using the channel access code, the HEC in the header, and the CRC in the payload.

A packet may consist of:
- the shortened access code only
- the access code and the packet header
- the access code, the packet header and the payload.



General Packet Format

Access Code
Every packet starts with an access code. The Access code is used for synchronization, DC offset compensation and identification. The access code identifies all packets exchanged on a physical channel.

Header Error Check (HEC)
Each header has a header-error-check to check the header integrity. The HEC is an 8-bit word code.

CRC Code
There is a 16-bit CRC in the payload.

KEY MANAGEMENT

The Link keys

A link key is used in the authentication procedure and as one of the parameters to calculate the encryption key.

The link keys are either semi-permanent or temporary. A semi-permanent link key may be stored in non-volatile memory and may be used after the current session is terminated. The lifetime of a temporary link key is limited by the lifetime of the current session.

Four types of link keys have been defined:

The unit key K_A: If a device A has little memory, it can use the unit key for all of the connections. This key is changed very rarely.

The initialization key K_{init}: The initialization key is used as the link key during the initialization process. The initialization parameters are encrypted using the initialization key and transferred. The key is derived from a random number, an L-byte PIN code, and a BD_ADDR.

The combination key K_{AB}: The combination key is specific to one pair of devices.

The temporary key K_{master}: If a master device wants to send a broadcast message to more than two devices simultaneously, it replaces the original link key temporarily with the master key and uses the master key as the link key.

The combination key K_{AB} and the unit key K_A are functionally indistinguishable. The difference is in the way they are generated. The combination key is derived from information in both devices A and B. The combination key is derived for each new combination of the pair (A, B).

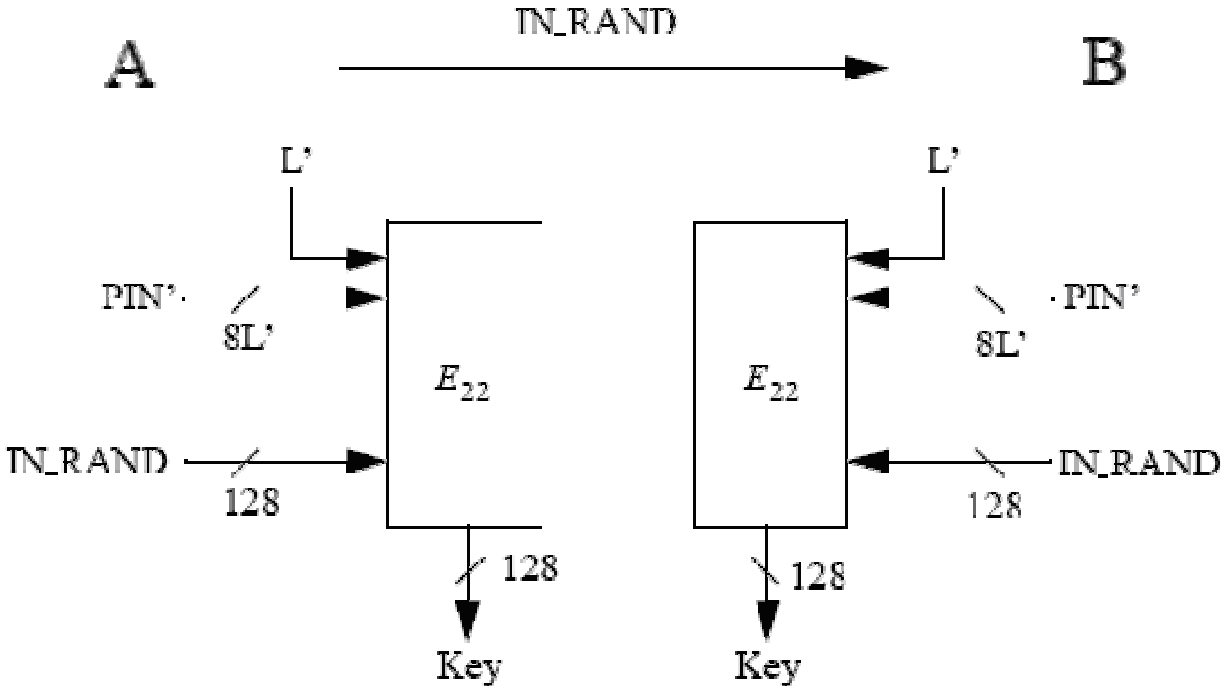
The PIN

The PIN may be a fixed number provided with the device or it can be chosen by the user. If no PIN is available, a default value of zero (0x00) is used. The PIN code may be chosen to be any length from 1 to 16 bytes.

The Encryption key, K_c

The encryption key is derived from the current link key. Each time encryption is activated, the encryption key shall be changed automatically. To be able to use a shorter encryption key without weakening the strength of the authentication, the encryption key length can be separately configured.

Generation of the initialization key, K_{init}



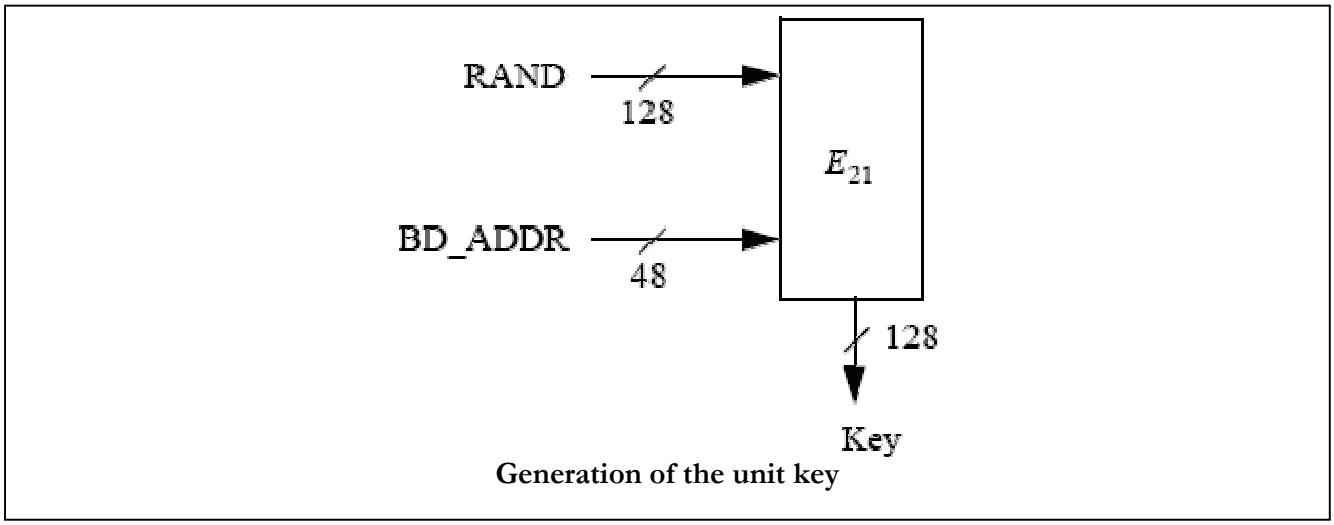
Generation of the initialization key

A link key that is used temporarily during initialization, is called the initialization key K_{init}. The E_{22} (see page 21) algorithm produces K_{init} from **a BD_ADDR**, **a PIN code**, **the length of the PIN** (in bytes, **L'**), and **a random number IN_RAND**.

The PIN is entered by a human user into both devices and it serves as the original secret used for key generation. The PIN is augmented with the BD_ADDR. If one device has a fixed PIN the BD_ADDR of the other device shall be used. If both devices have a variable PIN the BD_ADDR of the device that received IN_RAND shall be used. If both devices have a fixed PIN they cannot be paired. Since the maximum length of the PIN used in the algorithm cannot exceed 16 bytes, it is possible that not all bytes of BD_ADDR will be used.

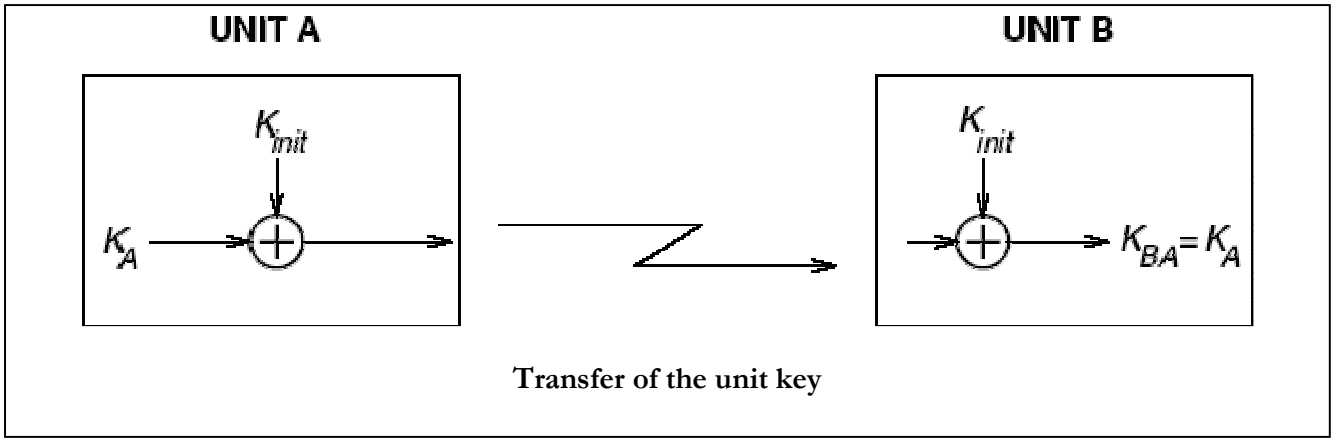
Generation of the unit key, K_A

The unit key shall be generated by the E_{21} algorithm. (see page 21)



Generation of the unit key

The unit key of device A, K_A, is being used as the link key for the connection A-B; device A sends the unit key K_A to device B; device B will store K_A as the link key K_{BA}. For another initialization, for example with device C, device A will reuse its unit key K_A, whereas device C stores it as K_{CA}.



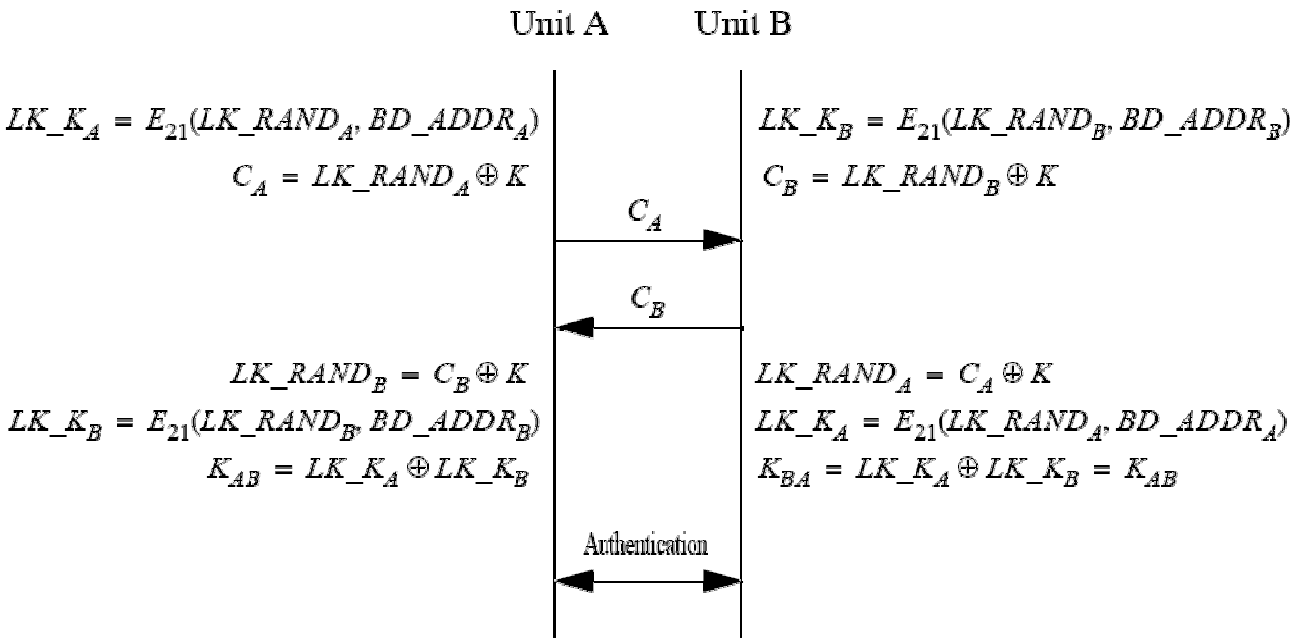
Transfer of the unit key

Generation of the combination key, K_{AB}

The combination key is the combination of two numbers generated in device A and B, respectively. Each device generates a random number, LK_RAND_A and LK_RAND_B. Then, utilizing E_{21} they generate LK_K_A and LK_K_B respectively.

$LK\_K=E_{21}(LK\_RAND, BD\_ADDR)$

LK_K_A and LK_K_B are XORed with the current link key which is already shared (during the initialization the link key is K_{init}) and exchanged. Device A calculates LK_RAND_A and Device B calculates LK_RAND_B. K_{AB} is calculated simply by XORing LK_K_A and LK_K_B.



Generation of the combination key

When both devices have derived the new combination key, a mutual authentication procedure is initiated to confirm the success of the transaction. The old link key is discarded after a successful exchange of a new combination key.

Generation of the master key, K_{master}

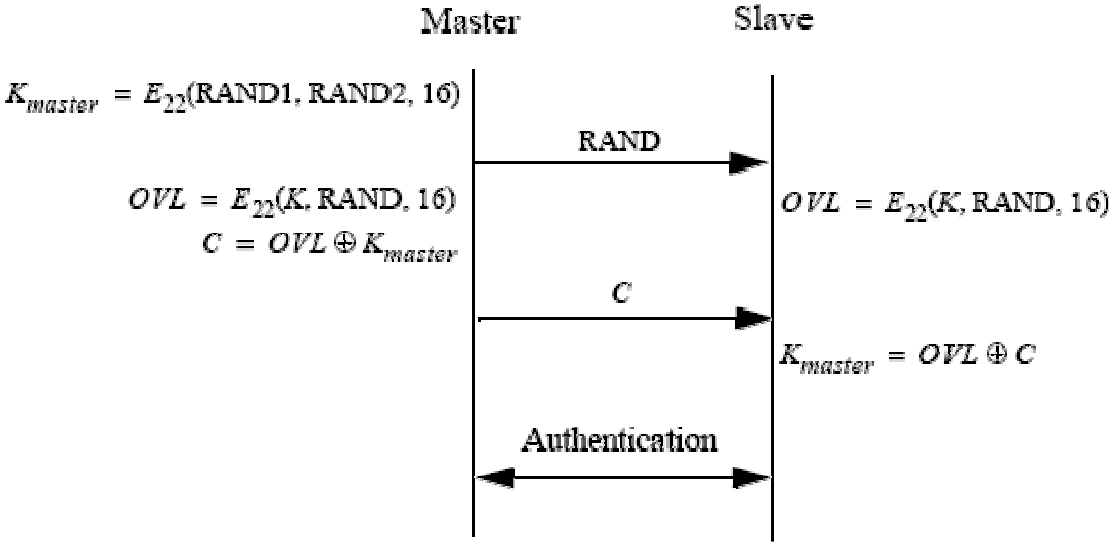
The master creates a new link key from two 128-bit random numbers, RAND1 and RAND2.

$K_{master} = E_{22}(RAND_1, RAND_2, 16)$

Another RAND is send to the slave. On both sides an overlay (OVL) is calculated using E_{22} with the current link key and the RAND as inputs.

$OVL = E_{22}(K, RAND, 16)$

The master sends the bitwise XOR of the OVL and the new link key to the slave. The slave recalculates K_{master}. To confirm the success of this transaction, the devices perform an authentication procedure using the new link key. This procedure is repeated for each slave that receives the new link key.

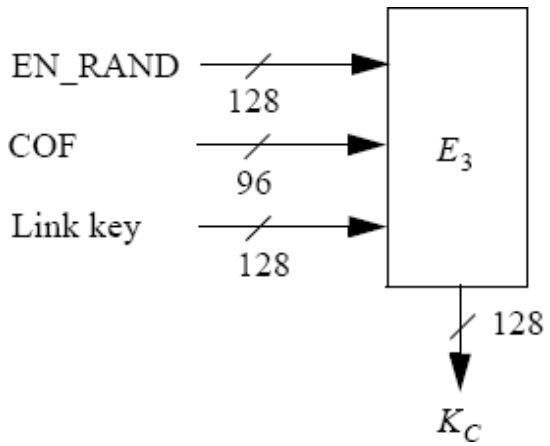


Distribution of the master key

Generation of the encryption key, K_C

The encryption key, K_C, is derived by algorithm E_3 (see page 22)from the current link key, a 96-bit **C**iphering **OF**fset number (COF), and a 128-bit random number. The COF is determined in one of two ways. If the current link key is a master key, then COF is derived concatinating BD_ADDR with itself . Otherwise it is ACO (see page 19) as computed during the authentication procedure.

$$COF = \begin{cases} BD\_ADDR \cup BD\_ADDR, & \text{if link key is a master key} \\ ACO, & \text{otherwise.} \end{cases}$$



Generation of encryption key

AUTHORIZATION

Authorization is the process by which a Bluetooth device determines whether or not another device is allowed access to a particular service. Authorization incorporates two important Bluetooth security concepts: trust relationships and service security levels. Authorization is dependent on authentication as the authentication process establishes the device identity which is used to determine access. [1]

The Bluetooth specification allows for three different levels of trust between devices:

- **Trusted:** Device is authenticated, and its access to services on device is allowed.
- **Untrusted:** Device is authenticated, but its access to services on device is restricted.
- **Unknown:** Device has not been authenticated and it is considered untrusted..

Service security levels control access to a device's services on a per service basis. There are three service security levels:

- **Service security level 1:** Authorization and authentication are required. The identity of the requesting device has to be confirmed and the requesting device has to be granted specific permission to access the service.

- **Service security level 2:** Only authentication is required. The identity of the requesting device need only be judged genuine in order to be granted access to the service.

- **Service security level 3:** Open to all devices. Access to the service will be granted to any device that is encrypting its communications.

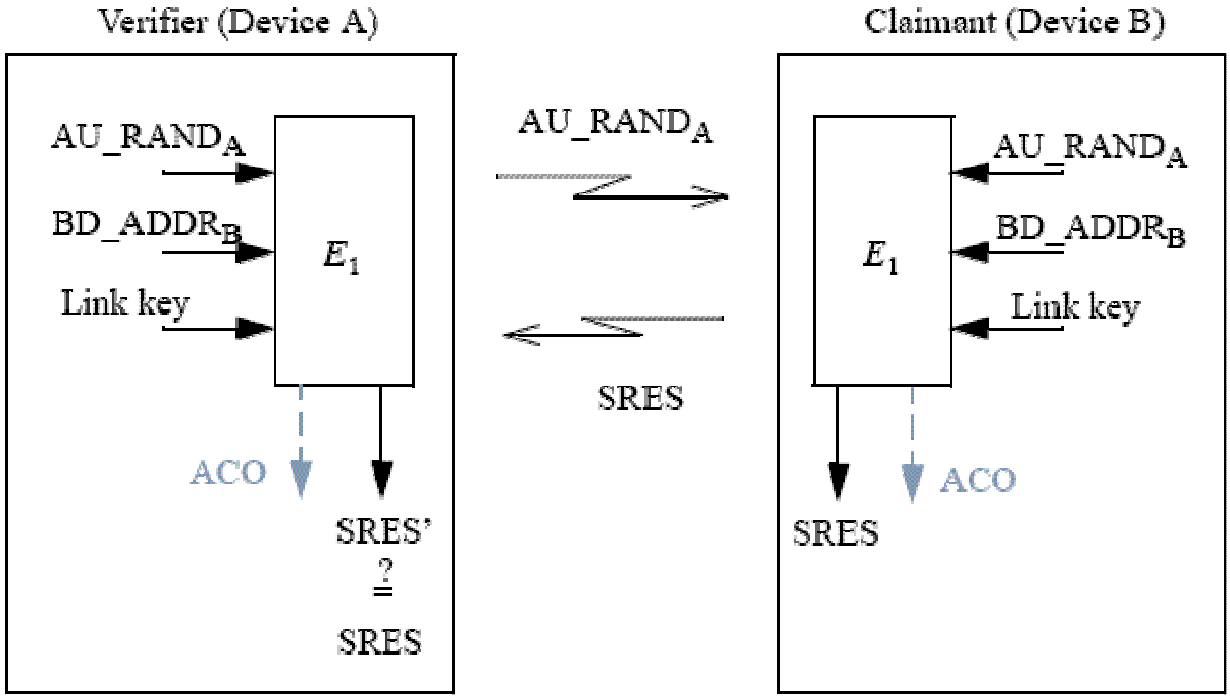
Associated with these levels are the following security controls to restrict access to services:

- Authorization required (this always includes authentication),
- Authentication required,
- Encryption required (link must be encrypted before the application can be accessed).

AUTHENTICATION

For each new connection between A and B, they use the common link key for authentication. They don't need to generate a new K_{init}. During each authentication, a new AU_RAND_A is issued.

Authentication uses a challenge-response scheme in which a claimant's knowledge of a secret key is checked through a 2-step protocol using symmetric secret keys. The latter implies that a correct claimant/verifier pair shares the same secret key, for example K. In the challenge-response scheme the verifier challenges the claimant to authenticate a random input (the challenge), denoted by AU_RAND_A, with an authentication code, denoted by E_1 (see page 19), and return the result SRES to the verifier. This figure also shows that the input to E_1 consists of the tuple AU_RAND_A and the Bluetooth device address (BD_ADDR) of the claimant. The use of this address prevents a simple reflection attack. The secret K shared by devices A and B is the current link key.



Challenge-response for the Bluetooth

When the authentication attempt fails, for each subsequent authentication failure with the same Bluetooth device address, the waiting interval is increased exponentially.

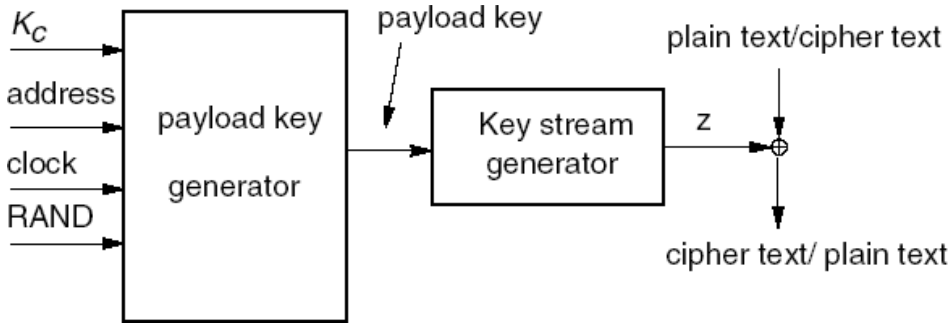
When no new failed attempts are made during a certain time period, the waiting time is exponentially decreased. This procedure prevents denial-of-service attacks, an intruder from repeating the authentication procedure with a large number of different keys.

To make the system less vulnerable to denial-of-service attacks, the devices should keep a list of individual waiting intervals for each device it has established contact with. For a mutual authentication, this 2-step protocol is repeated with exchanged roles. [4]

ENCRYPTION

Only the packet payload is encrypted; the access code and the packet header is never encrypted. The encryption is carried out with a stream cipher E_0. E_0 is re-synchronized for every payload.

The stream cipher system E_0



The stream cipher system E_0

The stream cipher system E_0 shall consist of three parts: [4]

- the first part performs initialization (generation of the payload key). The payload key generator combines the input bits in an appropriate order and shifts them into the four LFSRs used in the key stream generator.

- the second part generates the key stream bits by using a method derived from the summation stream cipher generator attributable to Massey and Rueppel. The second part is the main part of the cipher system, as it will also be used for initialization.

- the third part performs encryption and decryption.

Encryption of broadcast messages

There are three different encryption modes:

Mode 1: No encryption (Default).
Mode 2: Point-to-point only encryption. Broadcast messages are not encrypted.
Mode 3: All messages are encrypted.

Encryption Procedure

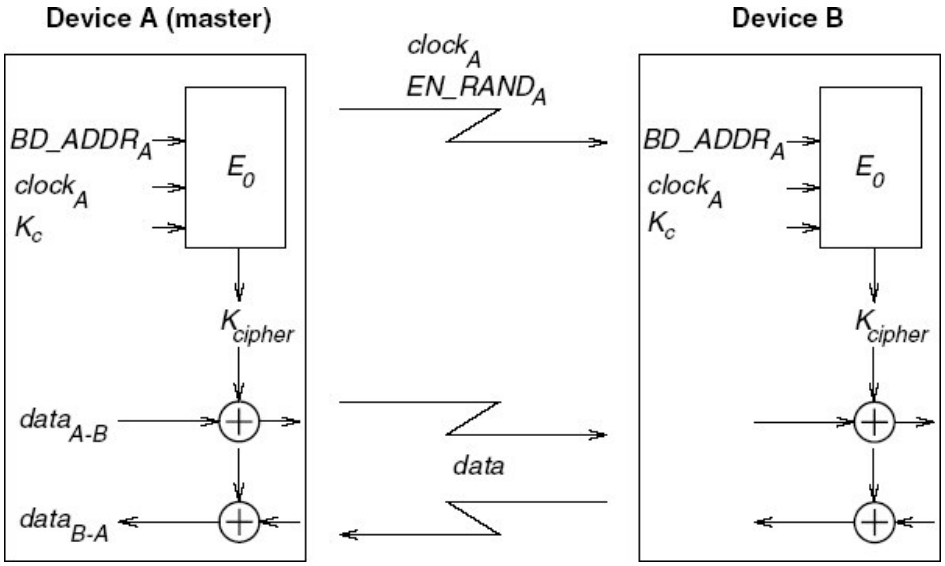
Each device has a maximal and minimal allowed key length. Before generating the encryption key, the devices involved shall negotiate to decide the key size to use.

Each packet payload is encrypted separately. The cipher algorithm E_0 uses the master Bluetooth device address, 26 bits of the master real-time clock and the encryption key as input.

The encryption algorithm E_0 generates a binary keystream, K_{cipher}, which is XORed to the data to be encrypted. The encryption key K_C is derived from the current link key, COF, and a random number, EN_RAND_A, as described above.

$$E_0(BD\_ADDR, CLK_{26\text{-}1}, K_C)$$

The real-time clock is incremented for each slot. The E_0 algorithm is reinitialized at the start of each new packet. By using $CLK_{26\text{-}1}$ at least one bit is changed between two transmissions. Thus, a new keystream is generated after each reinitialization. For packets covering more than a single slot, the Bluetooth clock as found in the first slot is used for the entire packet.



Encryption procedure

The encryption algorithm E_0 generates a binary keystream, K_C, which is XORed with the data to be encrypted. The cipher is symmetric; therefore decryption is performed in exactly the same way using the same key as used for encryption.

In addition to reducing interference, Bluetooth's limited range and spread spectrum frequency hopping help to ensure confidentiality by reducing the possibility of eavesdropping. The use of fast frequency hopping, at 1600 hops per second, represents an important barrier to interception. Since the transmitter only dwells on a specific frequency for 625 microseconds, it is difficult to even detect the presence of a Bluetooth device unless it is in the process of actively paging another device. [7]

Most Bluetooth devices are equipped with radios which have a range of 10 meters. Potential eavesdroppers would have to be within this range to intercept a Bluetooth device's transmissions.

BASIC PROBLEMS OF BLUETOOTH SECURITY

PIN

The PIN is the only secret used for the key generation that is not transferred by wireless communication. For many applications, the PIN will be a relatively short string of numbers. Typically, it may consist of only four decimal digits. If the PIN is small or, even worse, has the value zero, then an exhaustive search can derive the initialization of security keys. [9]

Character set used	Min. recommended length	Minimum PIN length
0-9 (10 characters)	19 characters (= 63 bits)	12 characters (= 40 bits)
0-9 A-Z (36 characters)	12 characters (= 62 bits)	8 characters (= 41 bits)
0-9 A-Z, a-z (62 characters)	11 characters (= 65 bits)	7 characters (= 42 bits)
(Printable) ASCII (95 characters)	10 characters (= 66 bits)	6 characters (= 39 bits)

Encryption key length is negotiable

A more robust initialization key generation procedure must be developed.. Especially it should be assured that the key length used is not too small. [2]

E0 stream cipher algorithm is weak

The Bluetooth SIG needs to develop a more robust encryption procedure. [2]

Unit key sharing can lead to eavesdropping

A malicious device that once gained knowledge of a unit key of another device is able to intercept any further communication that is carried out using this compromised unit key. [2]

Device authentication is simple shared-key challenge-response

One-way-only challenge-response authentication is subject to man-in-the-middle attacks. Mutual authentication is required to provide verification that users and the network are legitimate. [2]

End-to-end security is not performed

Only individual links are encrypted and authenticated. Data is decrypted at intermediate points. Application software with additional security services above the Bluetooth software can be developed. [2]

Security services are limited

Audit, nonrepudiation, and other services do not exist. If needed, these can be developed at particular points in a Bluetooth network. [2]

Strength of the challenge-response pseudorandom generator is not known

It is widely known that the Random Number Generator (RNG) of low cost devices might be implemented in an insecure way. RNG may produce static number or periodic numbers that may reduce the effectiveness of the authentication scheme. [2]

Device Address Validation

Addresses are not validated. Therefore addresses can be spoofed. This is similar to IP address spoofing. It was determined that the spoofed device was capable of connecting to the authentic device to create a piconet. Text messages were transferred between the two devices. Moreover, Master-Slave switches were made between the two identical addresses. [9]

CONCLUSION

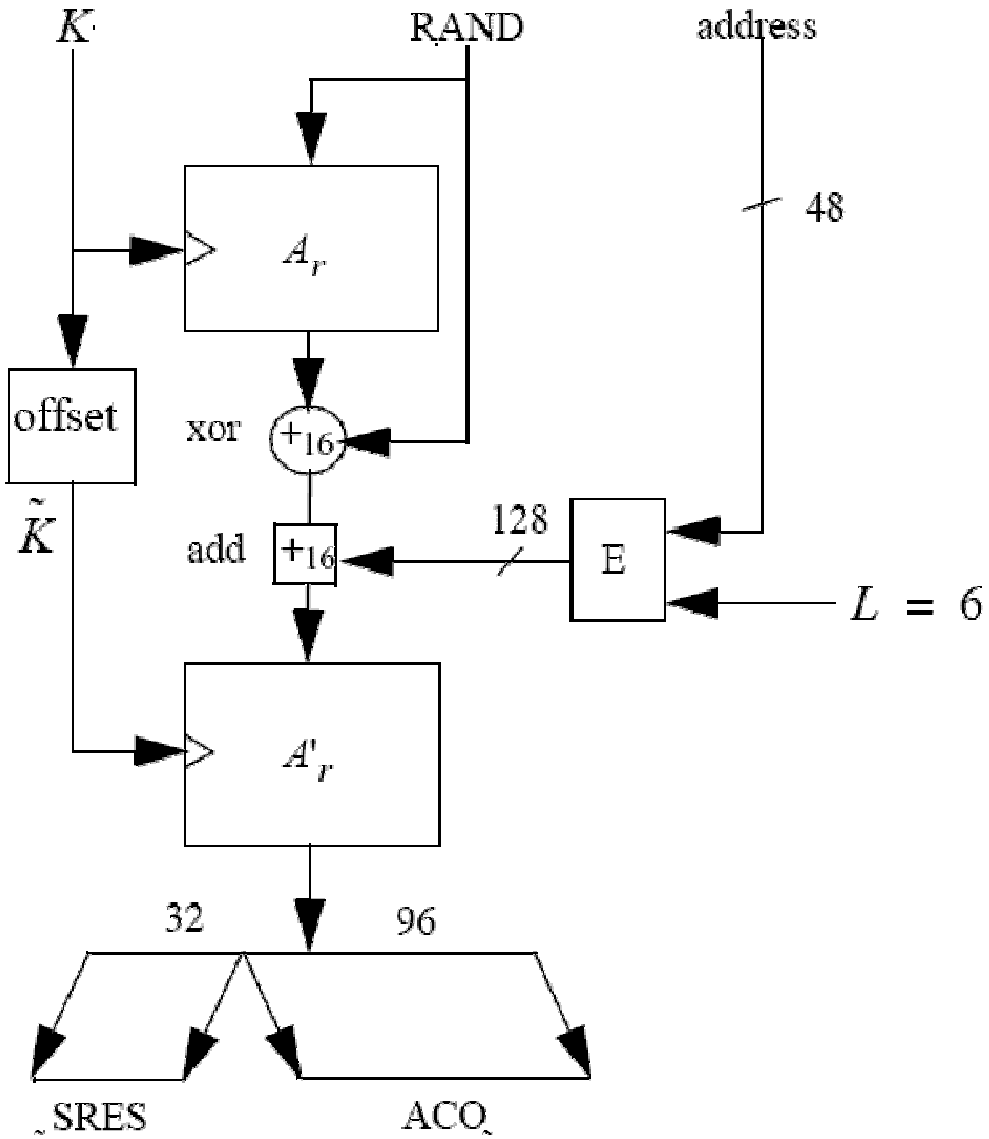
Bluetooth enables virtually effortless high security for cable replacement applications such as printers and headsets. This security includes 128-bit encryption keys that change for each session, but require no user action once initialized.

Bluetooth is a WPAN standard that is moderately secure but still has weaknesses in its security architecture, making it vulnerable to attacks by malicious intruders. With its ever-growing popularity as a standard technology in wireless personal networks, Bluetooth security has become an increasingly important aspect.

It is important that consumers understand the technology and the risks involved in the use thereof. Most of these risks can be easily mitigated by following device configuration guidelines and security policies when it comes to the use of a Bluetooth device.

THE AUTHENTICATION AND KEY-GENERATING FUNCTIONS

The Authentication Function E_1



$\tilde{K}[0] = (K[0] + 233) \mod 256$ $\quad \tilde{K}[1] = K[1] \oplus 229$

$\tilde{K}[2] = (K[2] + 223) \mod 256$ $\quad \tilde{K}[3] = K[3] \oplus 193$

$\tilde{K}[4] = (K[4] + 179) \mod 256$ $\quad \tilde{K}[5] = K[5] \oplus 167$

$\tilde{K}[6] = (K[6] + 149) \mod 256$ $\quad \tilde{K}[7] = K[7] \oplus 131$

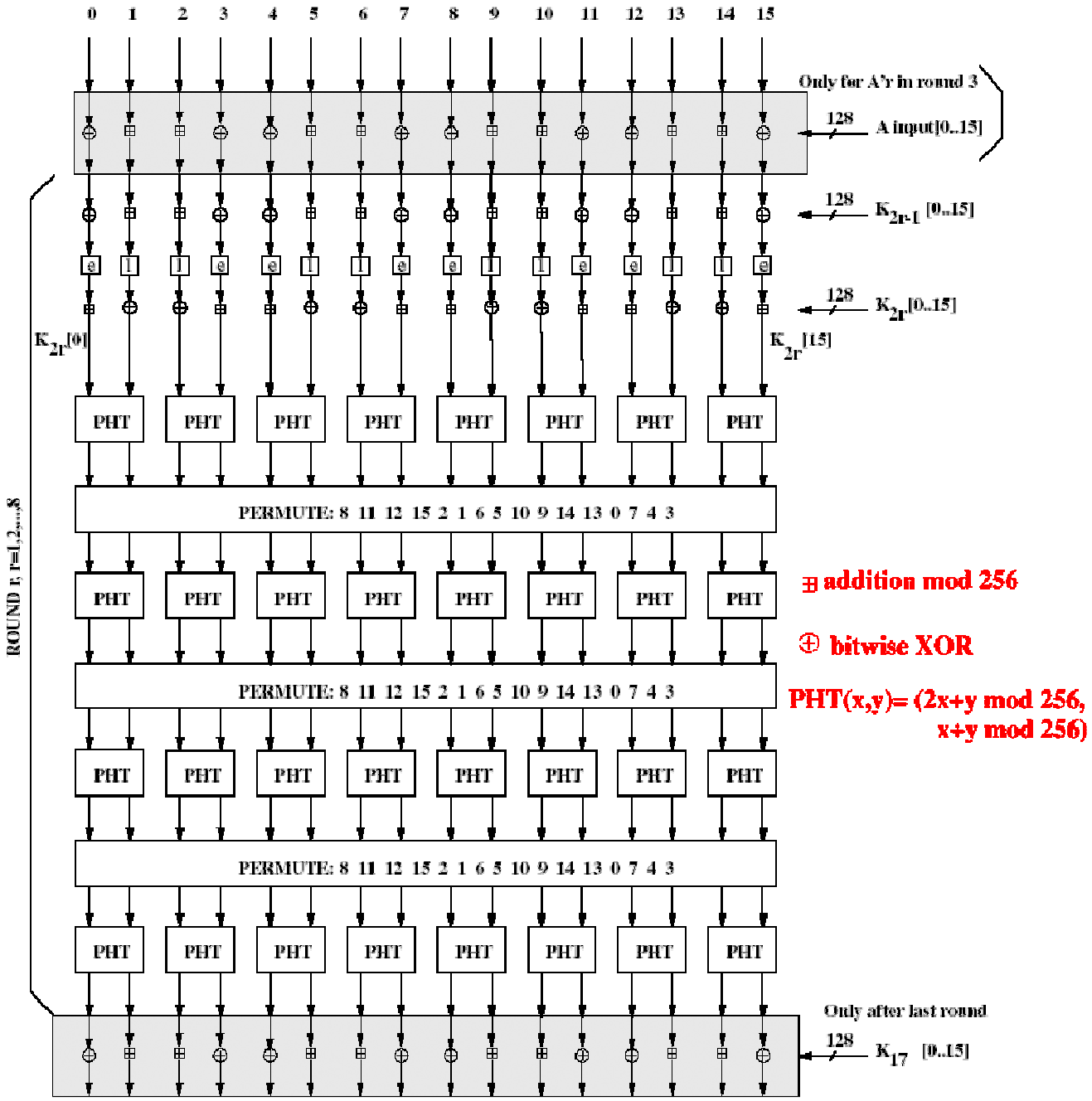
$\tilde{K}\{8\} = K[8] \oplus 233$ $\quad \tilde{K}[9] = (K[9] + 229) \mod 256$

$\tilde{K}[10] = K[10] \oplus 223$ $\quad \tilde{K}[11] = (K[11] + 193) \mod 256$

$\tilde{K}[12] = K[12] \oplus 179$ $\quad \tilde{K}[13] = (K[13] + 167) \mod 256$

$\tilde{K}[14] = K[14] \oplus 149$ $\quad \tilde{K}[15] = (K[15] + 131) \mod 256$

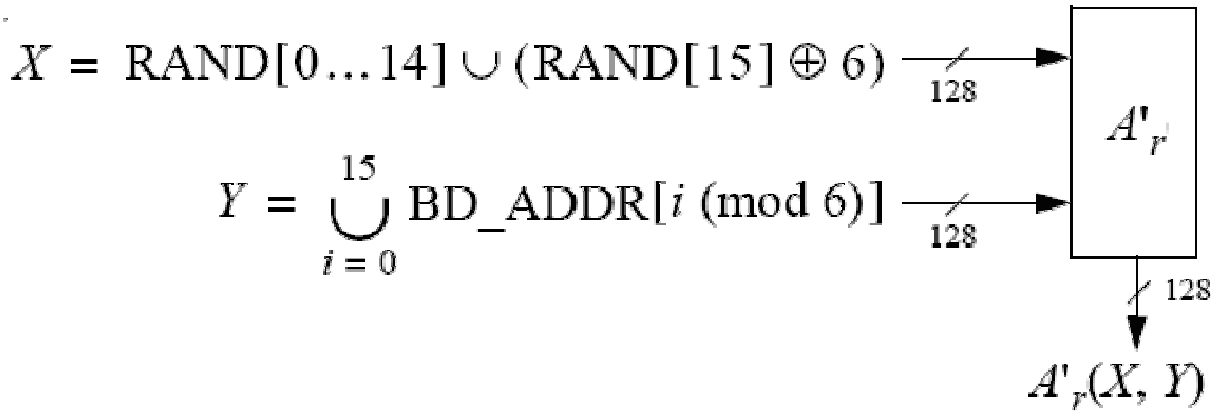
A_r and A'_r (SAFER+)



E_{21} Key Generation Function for Authentication

$$X = \text{RAND}[0 \ldots 14] \cup (\text{RAND}[15] \oplus 6)$$

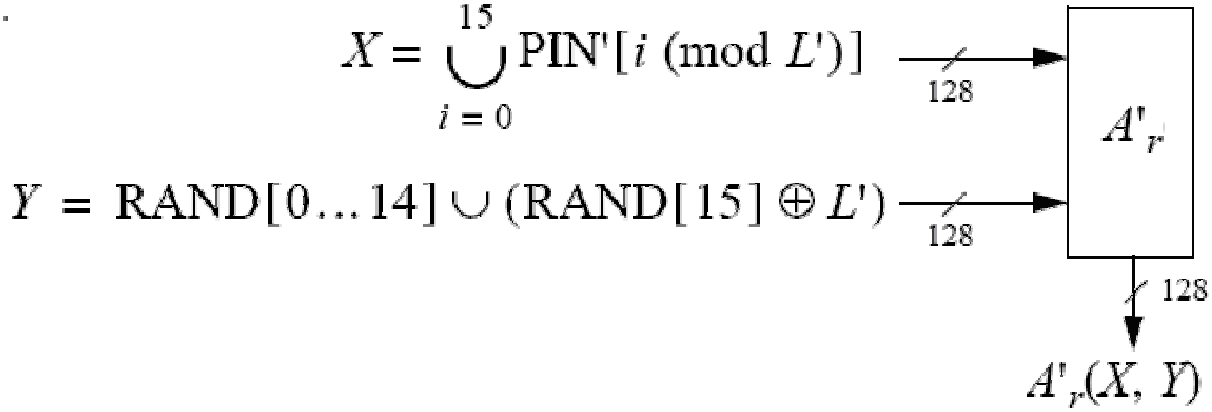
$$Y = \bigcup_{i=0}^{15} \text{BD\_ADDR}[i \pmod 6]$$



E_{22} Key Generation Function for Authentication

$$X = \bigcup_{i=0}^{15} \text{PIN}'[i \pmod{L'}]$$

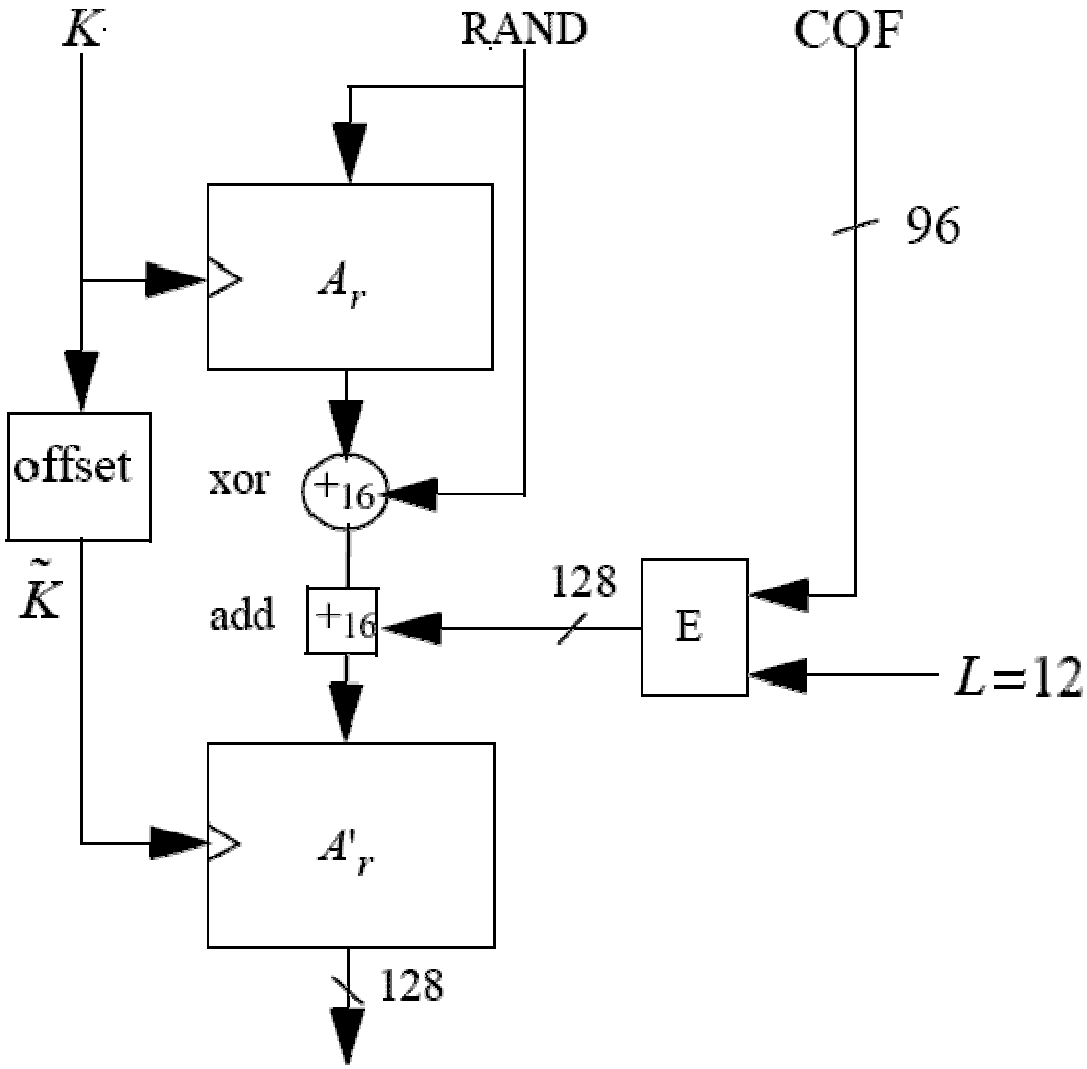
$$Y = \text{RAND}[0 \ldots 14] \cup (\text{RAND}[15] \oplus L')$$



$$L' = \min\{16, L + 6\}$$

$$\text{PIN}' = \begin{cases} \text{PIN}[0 \ldots L-1] \cup \text{BD\_ADDR}[0 \ldots \min\{5, 15-L\}], & L < 16, \\ \text{PIN}[0 \ldots L-1], & L = 16, \end{cases}$$

E_3 Key Generation Function for Encryption



$$COF = \begin{cases} BD\_ADDR \cup BD\_ADDR, & \text{if link key is a master key} \\ ACO, & \text{otherwise.} \end{cases}$$

BIBLIOGRAPHY

[1] Jeffrey B. Hall, Brush up on Bluetooth, 2003
http://www.giac.org/practical/GSEC/Jeffrey_Hall_GSEC.pdf

[2] Tom Karygiannis and Les Owens, Wireless Network Security: 802.11, Bluetooth and Handheld Devices, 2002
http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf

[3] Juha T. Vainio, Bluetooth Security, 2000
http://www.niksula.cs.hut.fi/~jiitv/bluesec.html

[4] Specification of the Bluetooth System, Security Specification
http://www.bluetooth.org/foundry/adopters/document/Bluetooth_Core_Specification_v1.2

[5] Adam Laurie, Serious flaws in Bluetooth security lead to disclosure of personal data, 2003
http://www.thebunker.net/security/bluetooth.htm

[6] Jun-Zhao Sun, Douglas Howie, Antti Koivisto, and Jaakko Sauvola, Design, Implementation and Evaluation of Bluetooth Security
http://www.mediateam.oulu.fi/publications/pdf/87.pdf

[7] Widcomm, Inc. "Bluetooth Security Solutions". 2001
http://www.widcomm.com/bluetooth/pdfs/BluetoothSecurityWP.pdf

[8] Oddbjørn Heimdal, Bluetooth Security, Protocol, Attacks and Applications, 2004
http://www.item.ntnu.no/fag/ttm4705/kollokvie_presentasjoner_2004/bluetooth_security.ppt

[9] Creighton T. Hager and Scott F. Midkiff, Demonstrating Vulnerabilities in Bluetooth Security

[10] Subhansu Bandyopadhyay, Anirban Majumdar, Om Ghosh, Samidh Charterjee, Santanu Chattopadhyay, "A Proposal for Improvement in Service-Level Security Architecture of Bluetooth"

[11] BSI, Bluetooth Threats and Security Measures, 2003
http://www.bsi.bund.de/english/brosch/B05_bluetooth.pdf

[12] Marjaana Träskbäck , Security Of Bluetooth: An Overview Of Bluetooth Security
http://www.cs.hut.fi/Opinnot/Tik-86.174/Bluetooth_Security.pdf

[13] Catharina Candolin, Security Issues For Wearable Computing And Bluetooth Technology
http://www.cs.hut.fi/Opinnot/Tik-86.174/btwearable.pdf

[14] Markus Jacobsson And Susanne Wetzel, Security Weaknesses In Bluetooth
http://www.informatics.indiana.edu/markus/papers/bluetooth.pdf

[15] Thomas Müller, Bluetooth Security Architecture Version 1.0, 1999
http://cnscenter.future.co.kr/resource/hot-topic/wpan/1c11600.pdf

[16] Nikhil Anand, An Overview of Bluetooth Security, 2001
http://www.giac.org/practical/gsec/Nikhil_Anand_GSEC.pdf

[17] Tu C. Niem, Bluetooth And Its Inherent Security Issues, 2002
http://www.giac.org/practical/GSEC/Tu_Niem_GSEC.pdf

[18] Christian Gehrmann, Bluetooth Security White Paper, Bluetooth SIG, 2002
http://www.bluetooth.com/upload/24Security_Paper.PDF

[19] Martin Hinz, Wireless LAN Sicherheit, 2002
http://www.computec.ch/dokumente/wireless/wireless_lan_sicherheit/wireless_lan_sicherheit.pdf

[20] Dirk Fox, Sicherheitsmechanismen des Bluetooth Standards (Version 1.1), 2002
http://www.secorvo.de/whitepapers/secorvo-wp05.pdf

[21] Keijo Haataja, Bluetooth security threats and possible countermeasures, 2004
http://www.cs.karelia.ru/fdpw/2004/haataja.pdf

[22] Dave Singelée, Overview of the security weaknesses in Bluetooth, 2003
http://www.esat.kuleuven.ac.be/cosic/seminars/slides/Security_Bluetooth.ppt

[23] Ollie Whitehouse, War Nibbling: Bluetooth Insecurity, 2003
http://www.atstake.com/research/reports/acrobat/atstake_war_nibbling.pdf

[24] Dr.-Ing. Dipl.-Inform. Bernhard Löhlein, T-Systems, ITC Security, Bluetooth – Sicherheitsarchitektur und Angriffspunkte, 2003
http://www.datensicherheit.nrw.de/Daten/ws051203/Vortraege/Loehlein.pdf

[25] Oddbjørn Heimdal, Bluetooth Security, Protocol, Attacks and Applications, 2004
http://www.item.ntnu.no/fag/ttm4705/kollokvie_presentasjoner_2004/bluetooth_security.ppt

[26] Ollie Whitehouse, Bluetooth: Red Fang, Blue Fang, 2004
http://cansecwest.com/csw04/csw04-Whitehouse.pdf

[27] Shawn Roberts, Bluetooth Encryption, 2004
http://www.cs.odu.edu/~mukka/cs772s04/Presentations/shawn.ppt