Bluetooth

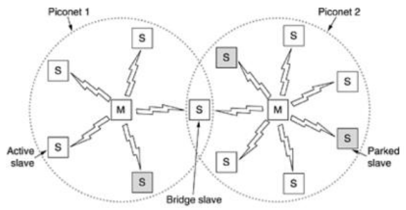
Ibéria Medeiros

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa

Overview

- Bluetooth employs short-range radio frequency communications to support data transmission over short distances from fixed and/or mobile devices, creating **wireless personal area networks** (WPANs)
- Since a radio communications system is used, devices do not have to be in line of sight of each other, and can even be in other rooms, as long as the received transmission is powerful enough
- Bluetooth is primarily designed for **low power consumption**, with a short transmission range (*class1*: 100 m; *class2*: 10 m; *class3*: 1 m)
- There is some flexibility on the **security mechanisms** that are employed in Bluetooth, which depend on the needs of the users, the importance of the exchanged data and the type of devices
- Security mechanisms are defined at the **link layer level**, and the applications are free to add other protection measures

Network Organization (**not** for LE version)



- A Bluetooth device can act as ***master*** or as ***slave*** node, however, devices can switch roles and a slave can become the master at any time
- Typically, the master only transfers data to one slave device, and in order to support several devices, the master switches nodes in round-robin fashion
- A master can communicate with up to seven devices, and this group of eight devices forms a ***piconet*** (up to 255 further devices can be inactive, or *parked*, which the master can bring into active status at any time)
- Two or more piconet can be connected to form a ***scatternet***, with some devices acting as a **bridge** by simultaneously playing the master role and the slave role

Protecting Connections using Physical Security

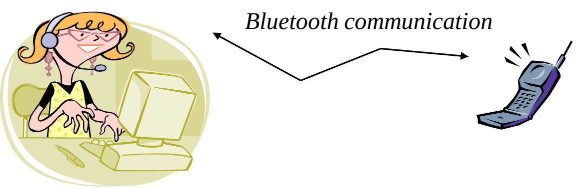
- The communication is based on frequency hopping spread spectrum
  - the data is transmitted during a very short interval at a certain frequency and then the device moves to another frequency
  - the schedule of the frequencies that are used is a pseudo-random sequence defined for each device (and depends on the device identifier and local clock (counter) value)
  - when the device is accepting connections (*discoverable mode*) it provides the necessary information for the other device to be able to exchange data
- Bluetooth also provides radio link power control
  - the strength of the radio signal is adjusted to the needs of the environment (e.g., distance, noise) to enable exchange of data

- These measures make attacks more difficult (eavesdropping), but they were shown **ineffective** for a determined attacker

Application Profiles

- Bluetooth defines a set of application profiles that represent possible utilizations of this network technology to support well-known kinds of applications
  - Cable replacement: replaces a variety of cables, such as those traditionally used for peripheral devices (e.g., mouse), printers, and wireless headsets that interface with desktops, laptops, cell phones, etc.
  - Ease of file sharing: can form a *piconet* to support file sharing capabilities with other Bluetooth devices, such as laptops.
  - Wireless synchronization: provide automatic synchronization between Bluetooth enabled devices (for example, contact information between smartphones and automobiles)
  - Internet connectivity: a device with Internet connectivity can share that access with other Bluetooth devices

Headset Profile



- There are usually two devices, a *Headset (HS)* and a *Audio Gateway (AG)* (mobile phone, laptop, CD, radio)
- It is assumed that the HS has a very simple interface with the user, and therefore some of the configuration aspects might be defined with the help of the AG (list of devices it can connect, ...)
- The objective is the creation of a secure association, i.e., authenticate and encrypt messages, between the two devices
- Depending on the Bluetooth version, security will be based on a secret ***passkey*** (or **PIN**)

Example Difficulties: Key Management

- The HS will probably only store a **single** passkey in non-volatile memory because the user interface is very simple/limited
- To ensure higher security, non-volatile memory should be ***tamper resistant***
- Since the passkey is setup at the HS factory,
  - each HS should be initiated with a distinct passkey
  - passkeys should have a reasonable dimension to prevent brute force attacks
- If there is a way to **physically** connect the HS to an external device (PDA or laptop) then the passkey could be changed

- Depending on the Bluetooth version, the passkey is used during the connection setup to generate further keys used for authentication and encryption
- When a existing device is about to be sold, these keys should be deleted because they might be re-used to improve the efficiency of the authentication process

Main Enhancements in Bluetooth Versions

Version		**Description**
1.1 + **Basic Rate (BR)**		BR up to 1 Mbps
1.2 + BR	2003	
2.0 + **Enhanced Data Rate** (EDR)	2004	EDR up to 3 Mbps
2.1 + EDR	2007	Introduced Secure Simple Pairing (SSP)
3.0 + **High Speed** (HS)	2009	Provides significant data rate improvements
4.0 + **Low Energy** (LE)	2010	Supports smaller, resource-constrained devices; allows for an unlimited number of slaves but no scatternet; introduces new security mechanisms
4.1+LE	2013	Minor improvements in security
4.2+LE	2014	

NOTE: Devices are normally **backward compatible!!**
Low energy also has other names such as **Smart, Wibree, Ultra Low Power**

Main differences: BR/EDR/HS vs. Low Energy

	BR / EDR / HS		**Low Energy**	
	Prior to 4.1	**4.1 onwards**	**Prior to 4.2**	**4.2 onwards**
Discovery/connect	Inquiry/paging		Advertising	
Device Address Privacy	None		Private addressing available	
Pairing Algorithm	**Prior to 2.1**: E21/E22/SAFER+ **2.1 – 4.0**: P-192 Elliptic curve, HMAC-SHA-256	P-256 Elliptic Curve, HMAC-SHA-256	AES-128	P-256 Elliptic Curve, AES-CMAC
Device Authentication	E1/SAFER	HMAC-SHA-256	AES-CCM	
Encryption	E0/SAFER+	AES-CCM		

MAC based on AES Cipher Block Chaining

***AES**-**C**ounter Mode **C**ipher Block Chaining-**M**AC Protocol*

Mostly insecure

Security Model

- Main threats
  - denial of service
  - eavesdropping
  - man-in-the middle
  - message modification and replays
  - resource misappropriation

  ….
- Basic security services
  - **authentication**: verify the identity of the devices based on their **address**
  - **confidentiality**: prevent the disclosure of data in the link
  - **authorization**: ensures that a **service** can only be accessed by authorized devices
  - **message integrity**: ensure that changes to messages are discovered
  - **pairing/bonding**: create and use shared keys for comms protection

Security Modes of Operation

- Devices must operate in one of the modes, which **dictate when security is initiated**
- **Security mode 1:** considered *non-secure* as auth + encryption is never initiated
- **Security mode 2:** security is enforced at *service level*, after physical link is created but before logical channel establishment
  - service discovery can be performed before auth+encryption is executed
  - a central security manager decides (**authorizes**) which services can be connected to

  link keys created through **Personal Identification Number (PIN) pairing**
- **Security mode 3:** *link level* enforced security
  - all connections must be auth+encryption and service discovery only occurs **after** that
- **Security mode 4:** security enforced at *service level* after physical link is created

  link keys created through **Secure Simple Paring**

Devices should use the most secure mode available but for compatibility reasons they **might downgrade** to a weaker mode**!**

PERSONAL IDENTIFICATION NUMBER (PIN) PAIRING

(SECURITY MODES 2 AND 3)

Will be studied during the TP class. This is a LEGACY approach!

SIMPLE PARING MODE

(SECURITY MODE 4)
(ONWARDS BLUETOOTH V2.1 + EDR)

Simple Pairing Protocol

- Bluetooth version "v2.1 + EDR" started supporting a new pairing mode, which aims at simplifying the user experience with Bluetooth devices, while increasing the strength of security

- Main goals
  - protection against passive eavesdropping (or passive attacks)
  - protection against man-in-the-middle attacks (or active attacks)

  achieving a higher security level than the one provided by a 16 character alphanumeric PIN (of version "2.0 + EDR" or earlier)

Protections Against Attacks

- Passive Attacks
  - achieved with a *strong link key + encryption algorithm*
  - key calculated using Elliptic Curve Diffie-Hellman public key crypto
  - ensures keys with an entropy in the order of 95 bits (16-digit PIN has 53 bits of entropy; 16-alphanumeric characters PIN has 95 bits of entropy)

> NOTE: This level of protection should be achieved <u>independently</u> of the length of the PIN

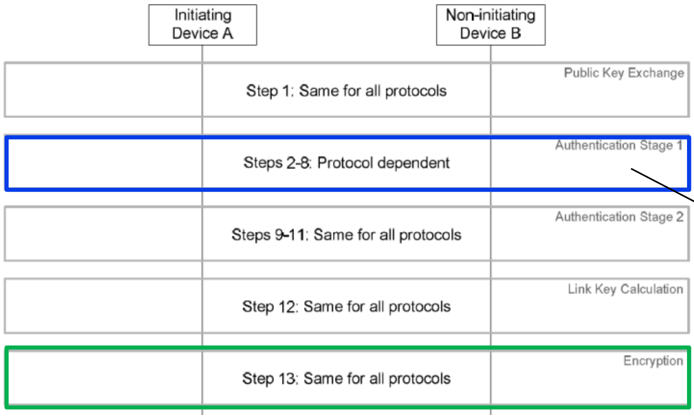
- Active (Man-in-the-middle) Attacks
  - objective of 1 in 1 million chance that MITM attack is successful
  - based on 6-digit number for comparison (plus the passkey)

> NOTE: The idea is not to prevent MITM, but to make it sufficiently hard to ensure that the user is warned while the attack is in progress (the connection is aborted a number of times)

Models of Operation (or Association)

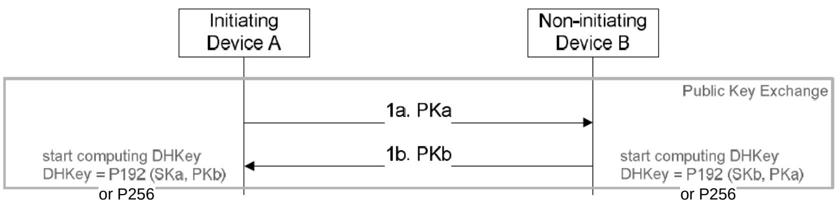
- **Numeric Comparison** : for scenarios where both devices can 1) display 6 digits and 2) can say “yes” or “no”, ensuring that
  - the right devices are connected to each other (even if addresses are not unique)
  - MITM can not be performed
- **Just Works** : at least one of the devices does not have a display and keyboard
  - works similar to Numeric Comparison, but since the user can only accept the connection, only protects from passive attacks
- **Out of Band (OOB)** : an OOB mechanism is used to discover the devices (i.e., obtain the Bluetooth address) and to transfer crypto parameters
  - OOB should be resistant to MITM and/or ensure privacy
  - depending on the OOB, a one-way or a two-way authentication is performed
- **Passkey Entry** : one device has input capabilities and the other has output capacity to display 6 digits
  - 6 digits are displayed in a device and the user has to enter the number on the other device

Overview of the Protocol



Specific for each mode of operation (Just Works is similar to Numeric Comparison)

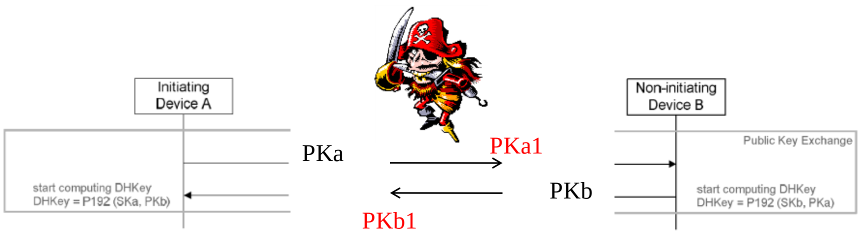
Public Key Exchange



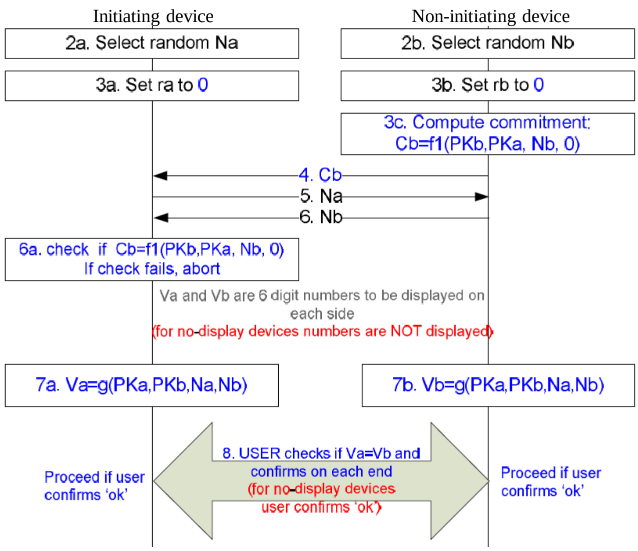
- Each device generates its DH private-public key pair, and then they exchange the public part
  - the DH key pair only needs to be generated once per device and its computation should occur prior to the authentication
  - the device can at any time discard its DH key pair and generate another
  - the DH key pair can be re-used in several authentications

Man-In-The-Middle

NOTE: this first phase is vulnerable to MITM attacks!



Authentication Stage 1: Numeric Comparison / Just Works



Nx – 128-bit nonce created by x
rx – random generated at x
f1 – generates 128-bit commitment values
g – generates 6-digit check values

Notes:
-the nonces are used to prevent replay attacks
- ra = rb = 0

Q: Can an active attacker perform a MITM?

Simple Man-In-The-Middle for Numeric Comparison

2a. Select random Na

3a. Set ra to 0

Knows: PKa, PKa1, PKb, PKb1
Get random Nb1
Send Cb1 = f1(PKb1, Pka, Nb1,0)
Let Na go
Send Nb1

Test should be OK

2b. Select random Nb

3b. Set rb to 0

3c. Compute commitment:
Cb=f1(PKb,PKa1 Nb, 0)

4
5
6

Cb
Na
Nb

6a. check if Cb=f1(PKb1PKa, Nb1 0)
If check fails, abort

Va and Vb are 6 digit n
ea
(for no-display devices r

umbers to be displayed on
h side
umbers are NOT displayed)

7a. Va=g(PKa,PKb1,Na,Nb1

Hope that
g(PKa,PKb1, Na, Nb1)
=
g(PKa1,PKb, Na, Nb)

which has 1 in 1000000
probability of occuring

7b. Vb=g(PKa1PKb,Na,Nb)

Proceed if user
confirms 'ok'

8. USER che
confirms
(for no-di
user co

ks if Va=Vb and
on each end
play devices
nfirms 'ok')

Proceed if user
confirms 'ok'

MITM: (Attempt to) Engineer matching display values

Knows: PKa, PKa1, PKb, PKb1

2b. Select random Nb

3b. Set rb to 0

3c. Compute commitment:
Cb=f1(PKb,PKa1 Nb, 0)

………….. capture Cb …………
but then needs to produce Cb1 based in some Nb1

Cb
Na
Nb

To produce Va = Vb, the attacker needs to find a Nb1 and a Na1 such that

g(PKa, PKb1, Na, Nb1)
=
g(PKa1, PKb, Na1, Nb)

knowing Cb
BUT
without knowing Nb or Na!!!

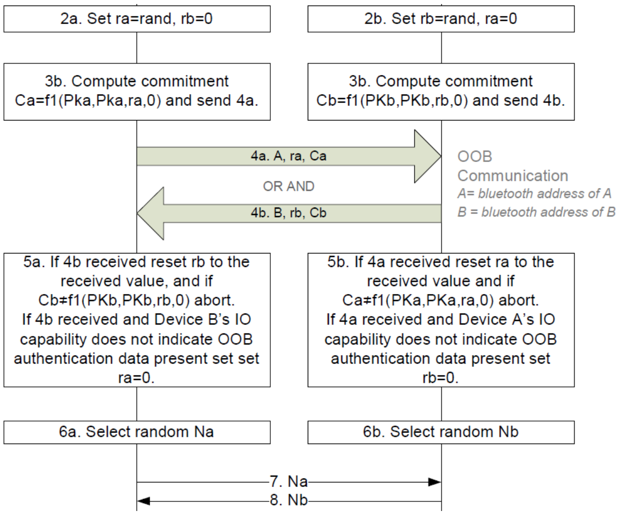
umbers to be displayed on
h side
umbers are NOT displayed)

7b. Vb=g(PKa1PKb,Na,Nb)

ks if Va=Vb and
on each end
play devices
nfirms 'ok')

Proceed if user
confirms 'ok'

Authentication Stage 1: OOB



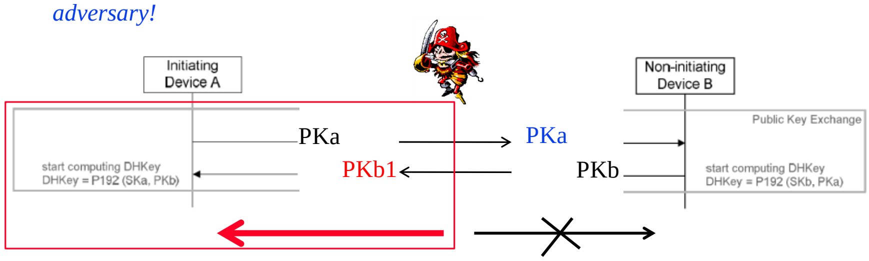
Nx – 128-bit nonce created by x
rx – random generated at x
f1 – generates 128-bit commitment values

Notes:
5a/5b – considers also the case were I try to send my data OOB, but the other side is unable to receive it; in this case, I set my rx = 0

Depending on the OOB capabilities, one-way or two-way authentication is performed

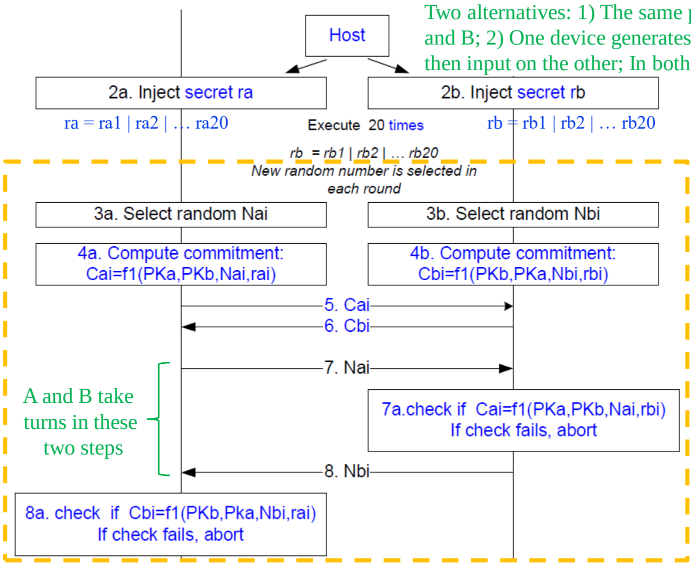
Should one side auth be enough?

- Imagine that A, ra, Ca are received by Device B, but not the other way around
  - this allows B to check that it received the correct PKa
  - to avoid detection, the adversary has to let the correct Pka go to B during the 1 phase
  - this prevents the adversary from setting up a key with B **and avoids the MITM!**
- What about a **personification attack** where the adversary tries to convince A that is B?
  - *since A is unable to check the information received from B, she will be deceived by the adversary!*



Authentication Stage 1: Passkey

Two alternatives: 1) The same passkey (or secret) is input in A and B; 2) One device generates and displays the passkey, which is then input on the other; In both cases, we need to check this!

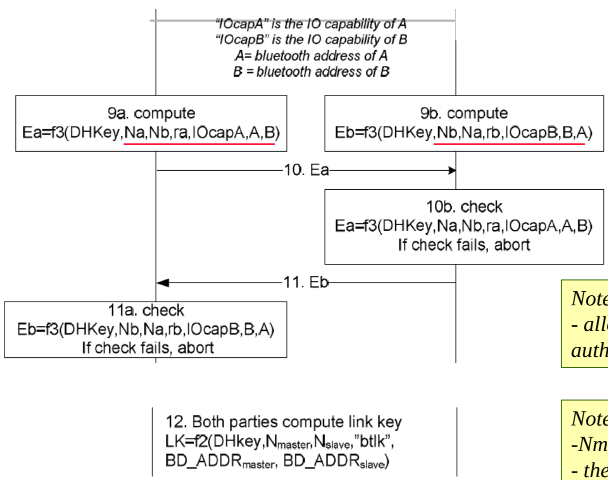


rx – passkey of x (ra should be ***equal*** *to rb)*
Nxi – 128-bit nonce created by x for round i
f1 – generates 128-bit commitment values

Notes:
-(2a/2b) the secret is entered in both devices, or displayed in one and entered on the other
-steps 3-8 are executed k times for a k-bit secret (in figure k=20 for a 6-digit passkey)

-At the end, Na=Na20 and Nb=Nb20

Authentication Stage 2 & Link Key Calculation

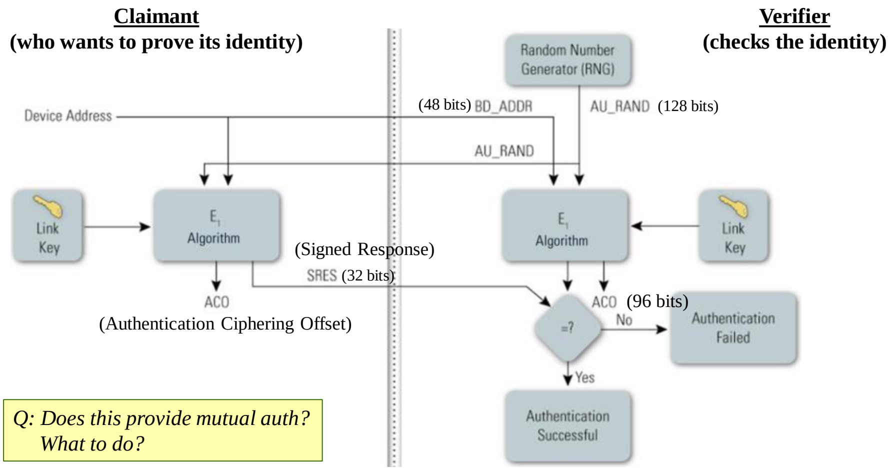


DHKey – generated DH key
Nx – 128-bit nonce created by x
rx – random generated at x
f2 – compute link key
f3 – compute check values

Notes:
- allows each device to confirm that authentication went correctly

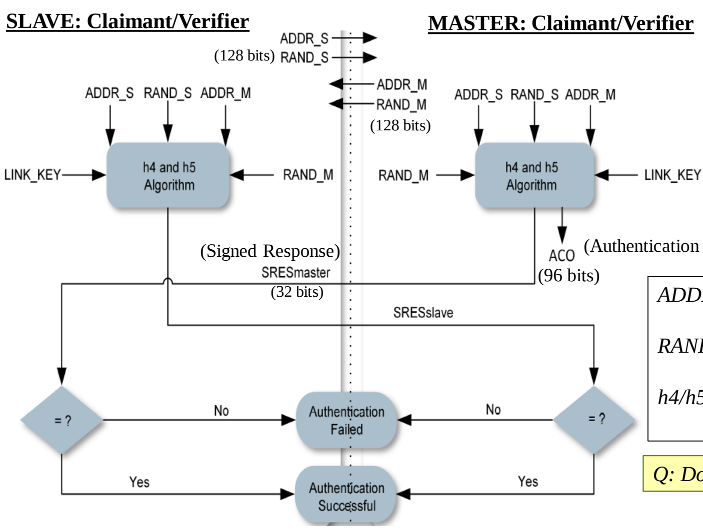
Notes:
-Nmaster = Na; Nslave = Nb
- the encryption key is then derived from the ***link key*** *LK*

Legacy Authentication (prior to Bluetooth version 4.1)



Q: Does this provide mutual auth? What to do?

Secure Authentication (Bluetooth version 4.1 onward)



ADDR_x – address of slave/master
RAND_x – random generated at slave/master
h4/h5 – based on HMAC-SHA-256

Q: Does this provide mutual auth?

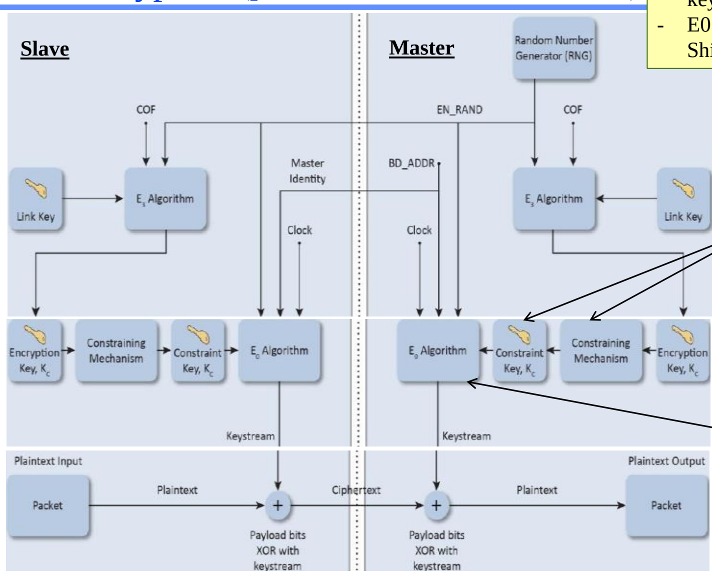
Encryption

- Three encryption modes exist
  - Mode 1: **no** encryption is performed in any traffic
  - Mode 2: individually addressed traffic is encrypted using a pair wise key; broadcast traffic is **not** encrypted
  - Mode 3: all traffic is encrypted using a key based on the master link key

- The key size can vary from 1 byte to 16 bytes depending on the country
  - the actual used size is defined during a negotiation phase
  - there should be a minimum key size parameter set on the device to prevent downgrade attacks that cause the key to become smaller than acceptable

Data Encryption (prior to Bluetooth version 4.1)

Notes:
- COF is basically the ACO for link keys
- E0 is based on Linear Feedback Shift Registers



STEP 1

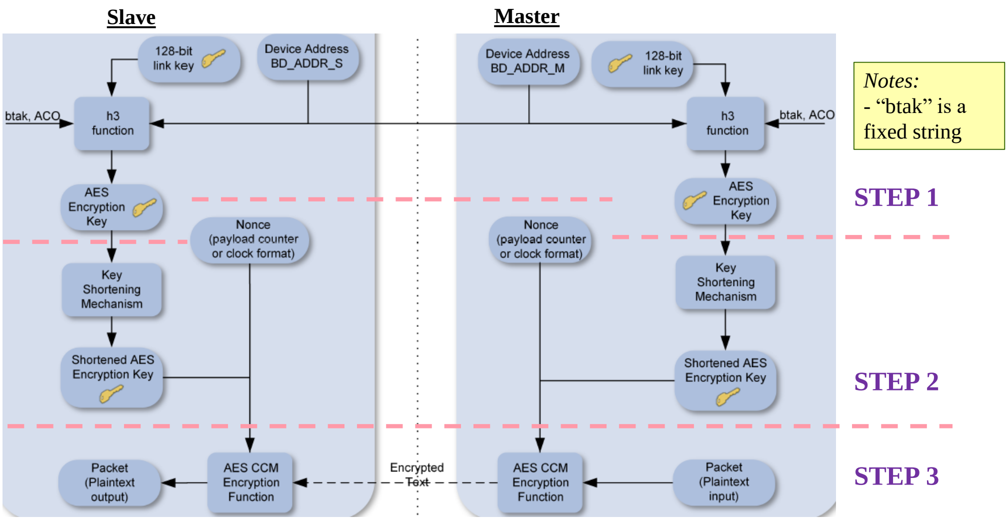
Due to restrictions on specific countries, one may use a ***constraint encryption key*** with less than 128 bits

STEP 2

A ***payload key*** is generated per packet using the clock (slot number), BD_ADDR, EN_RAND and Kc

STEP 3

Data Encryption (Bluetooth version 4.1 onward)

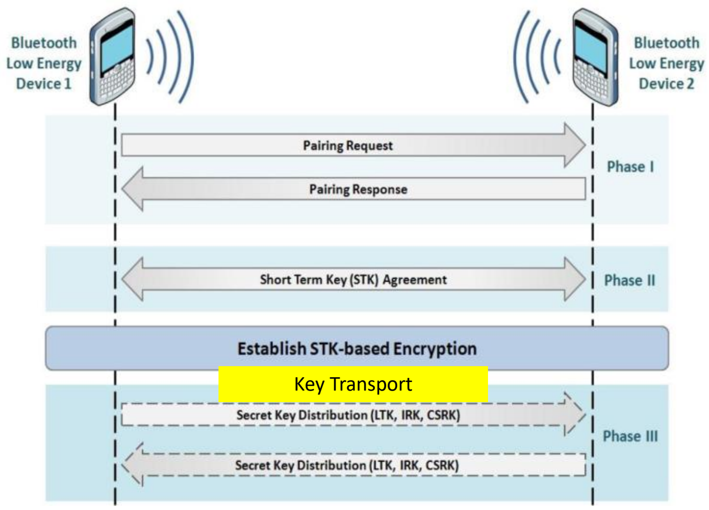


Bluetooth Smart (or LE)

- Due to the characteristics of the devices being targeted, very constrained devices, it is necessary to create a new security approach different than BR/EDR/HS
  - **LE paring** produces a **Long-Term Key** LTK (instead of a *link key*)
    - » *LE Legacy Paring*: LTK is shared using a *key transport protocol* instead of a key agreement protocol
    - » *LE Secure Connections*: LTK started again to use a *key agreement protocol*
  - **protects the data frames** using two approaches
    - » **AES-CCM** is used to authenticate, integrity and encrypt
    - » "sign" data frames by appending a message authentication code based on a **Connection Signature Resolving Key** CSRK
  - allows for the use o ***private addresses*** that change periodically to prevent tracking of the device
    - » the private address is an encrypted form of the *public address* using a **Identity resolving Key** IRK

LE Legacy Pairing (versions 4.0 & 4.1)

Considered INSECURE

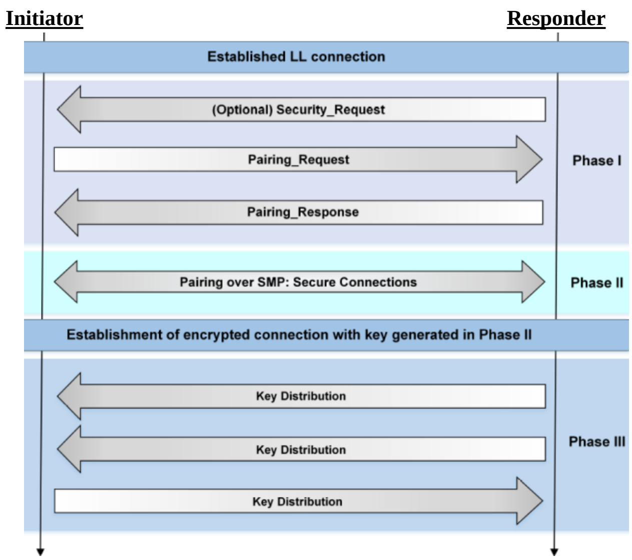


*II. Exchange material to generate a **temporary key** STK*

- *Based on different methods but called OOB / Passkey Entry / Just Works, **without using ECDH**, they obtain a Temporary Key TK*
- *With exception of OOB, the other two methods have no protection for eavesdropping => insecure!*
- *Exchange random values which together with TK is used to generate the STK*

III. Generate LTK, IRK and CSRK and exchange them encrypted with STK

LE Secure Connections (version 4.2)



- *Exchange I/O capabilities and security requirements (auth requirements & encryption key sizes)*
- *Exchange public DH keys and establish the LTK using the modes already discussed for secure pairing (e.g., numeric comparison) with a few adaptations*
- *Generate IRK and CSRK and exchange them protected with LTK*

Bibliography

- *Guide to Bluetooth Security*, NIST Special Publication 800-121, Revision 2, May 2017
- *Simple Pairing Whitepaper*, Bluetooth Special Interest Group, August 2006