Bluetooth Security

Bluetooth Security Features

Five basic security services are specified in the Bluetooth standard:

Authentication: verifying the identity of communicating devices based on their Bluetooth address. Bluetooth does not provide native user authentication.

Confidentiality: preventing information compromise caused by eavesdropping by ensuring that only authorized devices can access and view transmitted data.

Authorization: allowing the control of resources by ensuring that a device is authorized to use a service before permitting it to do so.

Message Integrity: verifying that a message sent between two Bluetooth devices has not been altered in transit.

Pairing/Bonding: creating one or more shared sec ret keys and the storing of these keys for use in subsequent connections in order to form a trusted device pair.

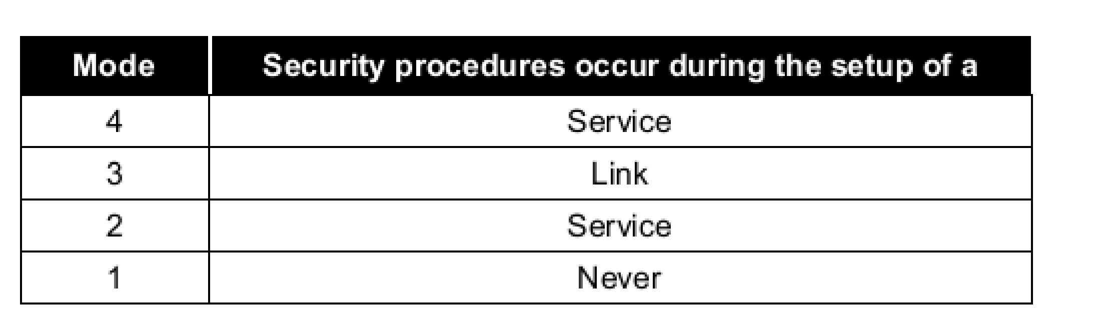
Security Features of Bluetooth

Bluetooth defines **authentication** and **encryption** security procedures that can be enforced during different stages of communication setup between peer devices.

- ✓ **Link - level** enforced refers to authentication and encryption setup procedures which occur before the Bluetooth physical link is completely established.

- ✓ **Service - level** enforced refers to authentication and encryption setup procedures which occur after the Bluetooth physical link has already been fully established and logical channels partially established.

Bluetooth Security Modes



Mode	Security procedures occur during the setup of a
4	Service
3	Link
2	Service
1	Never

✓ **Until Bluetooth 2.0,** **three modes** were defined which specified whether authentication and encryption would be link-level enforced or service - level enforced and that enforcement was configurable.

✓ **In Bluetooth 2.1,** a **fourth mode** was added which redefined the user experience during pairing, and required that if both devices are Bluetooth 2.1 or later, they are required to use the fourth mode.

- ✓ **Security Mode 4** (introduced in Bluetooth 2.1 + EDR) is a **service-level -enforced security** mode in which security procedures are initiated after physical and logical link setup.

- ✓ Security Mode 4 uses **Secure Simple Pairing (SSP)**, in which **ECDH key agreement** is utilized for link key generation

Bluetooth 4.0 & 4.1

- ✓ Until **Bluetooth 4.0**, the **P- 192 Elliptic Curve** was used for the link key generation

- ✓ In **Bluetooth 4.0**, device authentication and encryption algorithms were identical to the algorithms in **Bluetooth 2.0 + EDR** and earlier versions.

- ✓ **Bluetooth 4.1** introduced the **Secure Connections feature**, which allowed the use of the **P - 256 Elliptic Curve** for link key generation.

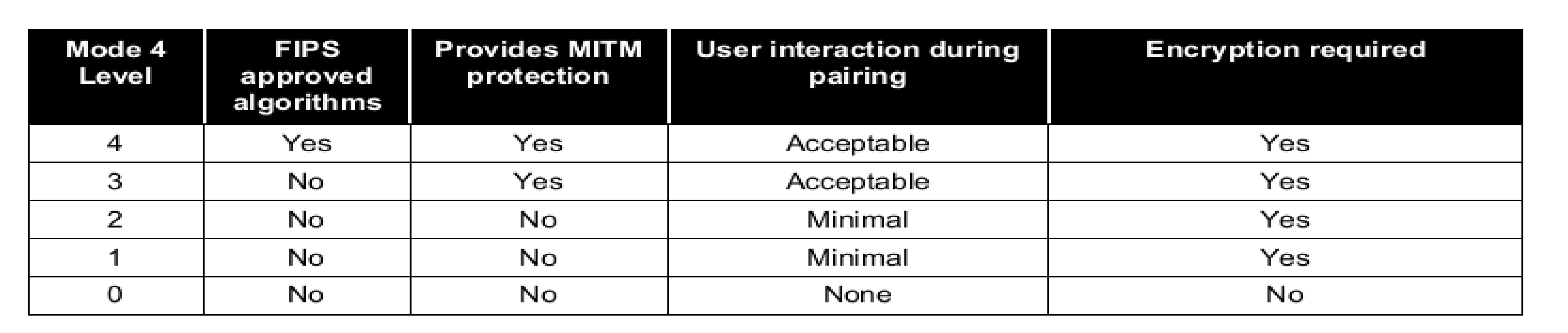
- ✓ **Bluetooth 4.1** the device authentication algorithm was upgraded to the FIPS- approved **HMAC - SHA - 256.**

✓ The encryption algorithm was upgraded to the FIPS-approved **AES - Counter** with **CBC- MAC (AES- CCM),** which also provides message integrity.

Security requirements for services protected by Security Mode 4 must be classified as one of the following

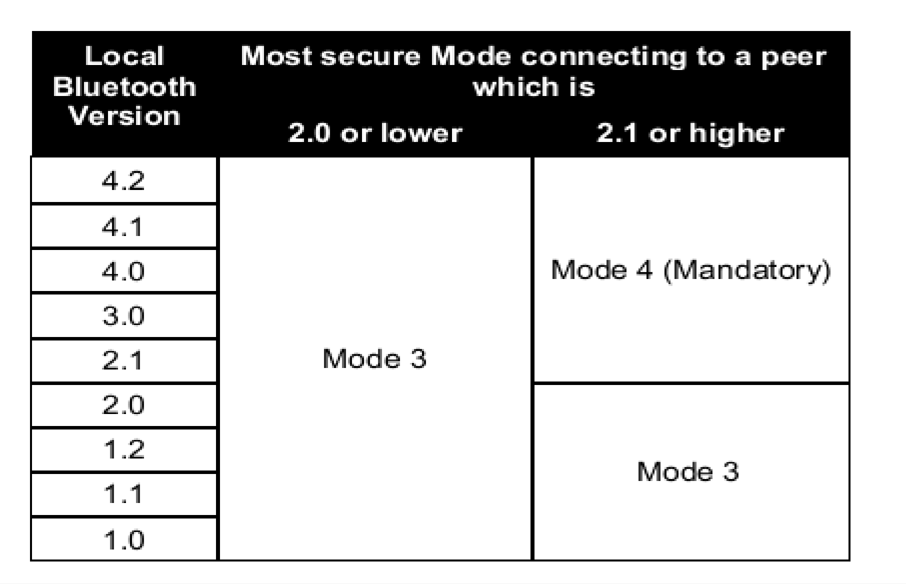
- ✓ **Level 4:** Authenticated link key using Secure Connections required
- ✓ **Level 3:** Authenticated link key required
- ✓ **Level 2:** Unauthenticated link key required
- ✓ **Level 1:** No security required
- ✓ **Level 0:** No security required. (Only allowed for SDP)

Bletooth Security Mode 4 Levels Summary



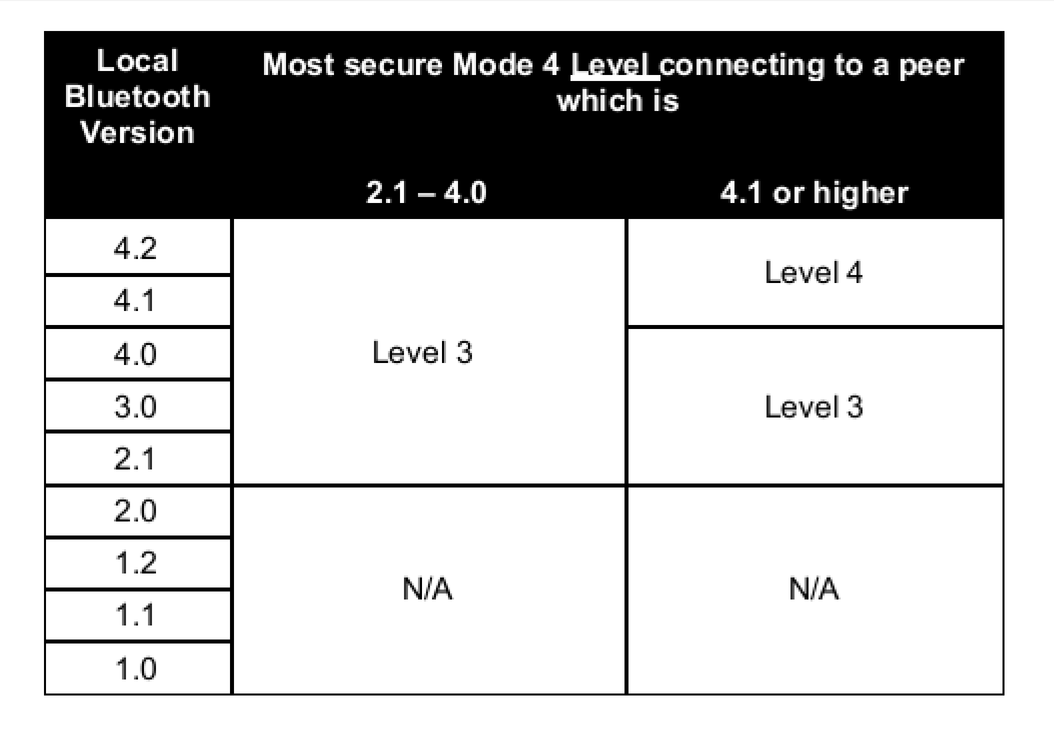
Mode 4 Level	FIPS approved algorithms	Provides MITM protection	User interaction during pairing	Encryption required
4	Yes	Yes	Acceptable	Yes
3	No	Yes	Acceptable	Yes
2	No	No	Minimal	Yes
1	No	No	Minimal	Yes
0	No	No	None	No

Most Secure Mode for a Pair of Bluetooth Devices



Local Bluetooth Version	Most secure Mode connecting to a peer which is	
	2.0 or lower	2.1 or higher
4.2	Mode 3	Mode 4 (Mandatory)
4.1		
4.0		
3.0		
2.1		
2.0		Mode 3
1.2		
1.1		
1.0		

Most Secure Level in Mode 4 for a Pair of Bluetooth Devices



Local Bluetooth Version	Most secure Mode 4 Level connecting to a peer which is	
	2.1 – 4.0	4.1 or higher
4.2	Level 3	Level 4
4.1		
4.0		Level 3
3.0		
2.1		
2.0	N/A	N/A
1.2		
1.1		
1.0		

Bluetooth Security Component

- ✓ **Pairing and Link Key Generation**
  - ✓ PIN/Legacy Pairing
  - ✓ Secure Simple Pairing
- ✓ Authentication
  - ✓ Legacy Authentication
  - ✓ Secure Authentication
- ✓ Confidentiality

Pairing and Link Key Generation

Essential to the authentication and encryption mechanisms provided by Bluetooth is the generation of a secret symmetric key.

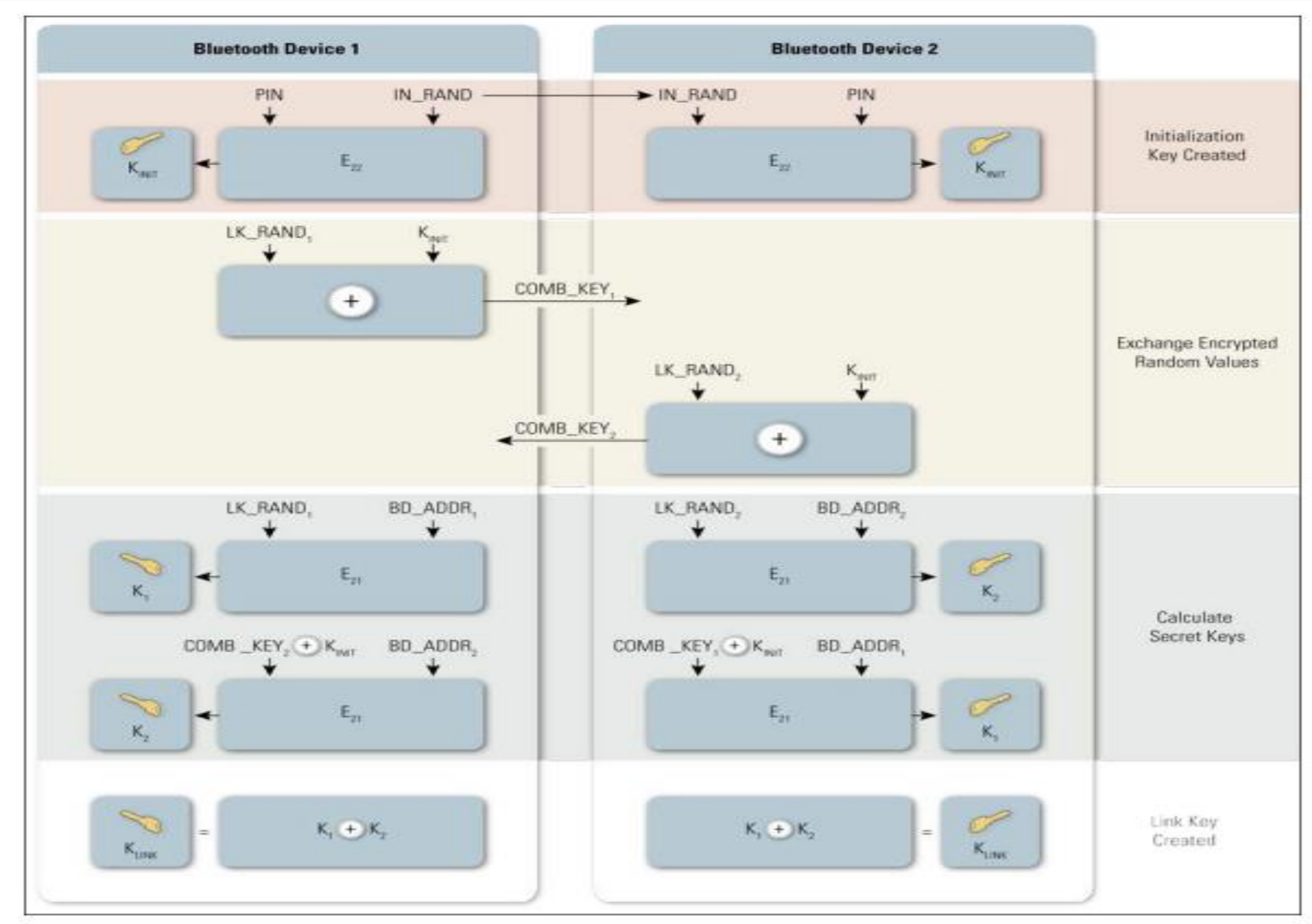
- ✓ In **Bluetooth BR/EDR** this key is called the **Link Key** and in **Bluetooth low energy** this key is called the **Long Term Key**.

- ✓ Bluetooth **BR/EDR** performs pairing (i.e., link key generation) in one of two ways.

- ✓ Security Modes 2 and 3 initiate **link key establishment** via a method called Personal Identification Number (**PIN**) Pairing (i.e., Legacy or Classic Pairing), while Security Mode 4 uses **SSP**.

PIN/Legacy Pairing

For PIN/legacy pairing, two Bluetooth devices simultaneously derive link keys when the user(s) enter an identical **secret PIN** into one or both devices, depending on the configuration and device type.



Secure Simple Pairing

✓ SSP was first introduced in Bluetooth 2.1 + EDR for use with Security Mode 4, and then improved in Bluetooth 4.1.

✓ SSP also improves security through the addition of **ECDH public key cryptography** for protection against passive eavesdropping and man- in -the- middle (MITM) attacks during pairing.

✓ When compared to PIN/Legacy Pairing, SSP simplifies the pairing process by providing a number of association models that are flexible in terms of device input/output capability.

Association models offered in SSP

Numeric Comparison

- ✓ **Numeric Comparison** was designed for the situation where both Bluetooth devices are capable of displaying a six - digit number and allowing a user to enter a “yes” or “no” response for pairing.

- ✓ A key difference between this operation and the use of PINs in legacy pairing is that the displayed number is not used as input for link key generation.

- ✓ Therefore, an eavesdropper who is able to view (or otherwise capture) the displayed value could not use it to determine the resulting link or encryption key.

Passkey Entry

- ✓ Passkey Entry was designed for the situation where one Bluetooth device has input capability (e.g., key board), while the other device has a display but no input capability.

- ✓ As with the Numeric Comparison model, the six- digit number used in this transaction is not incorporated into link key generation and is of no use to an eavesdropper.

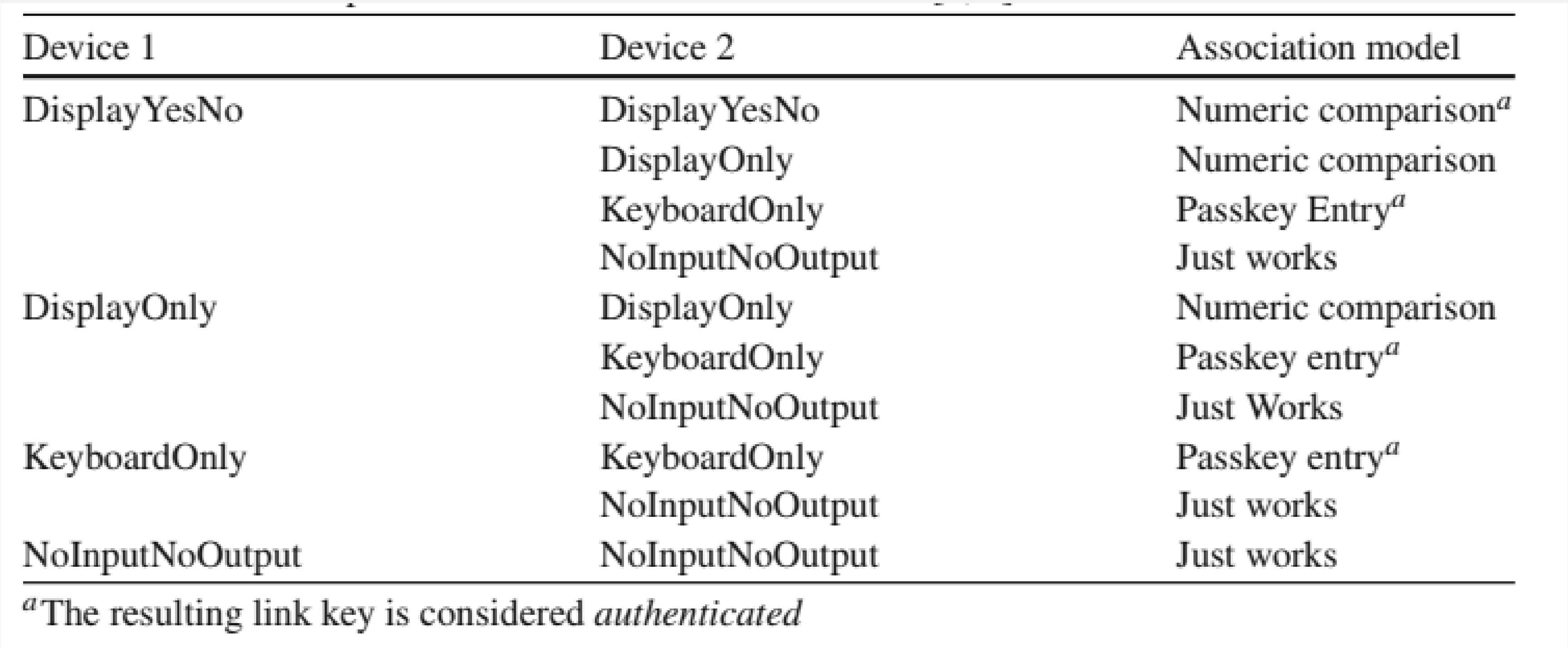
Just Works

- ✓ Just Works was designed for the situation where at least one of the pairing devices has neither a display nor a keyboard for entering digits (e.g., headset).
- ✓ Just Works provides no MITM protection.

Out of Band (OOB)

- ✓ Out of Band (OOB) was designed for devices that support a common additional wireless or wired technology for the purposes of device discovery and cryptographic value exchange.

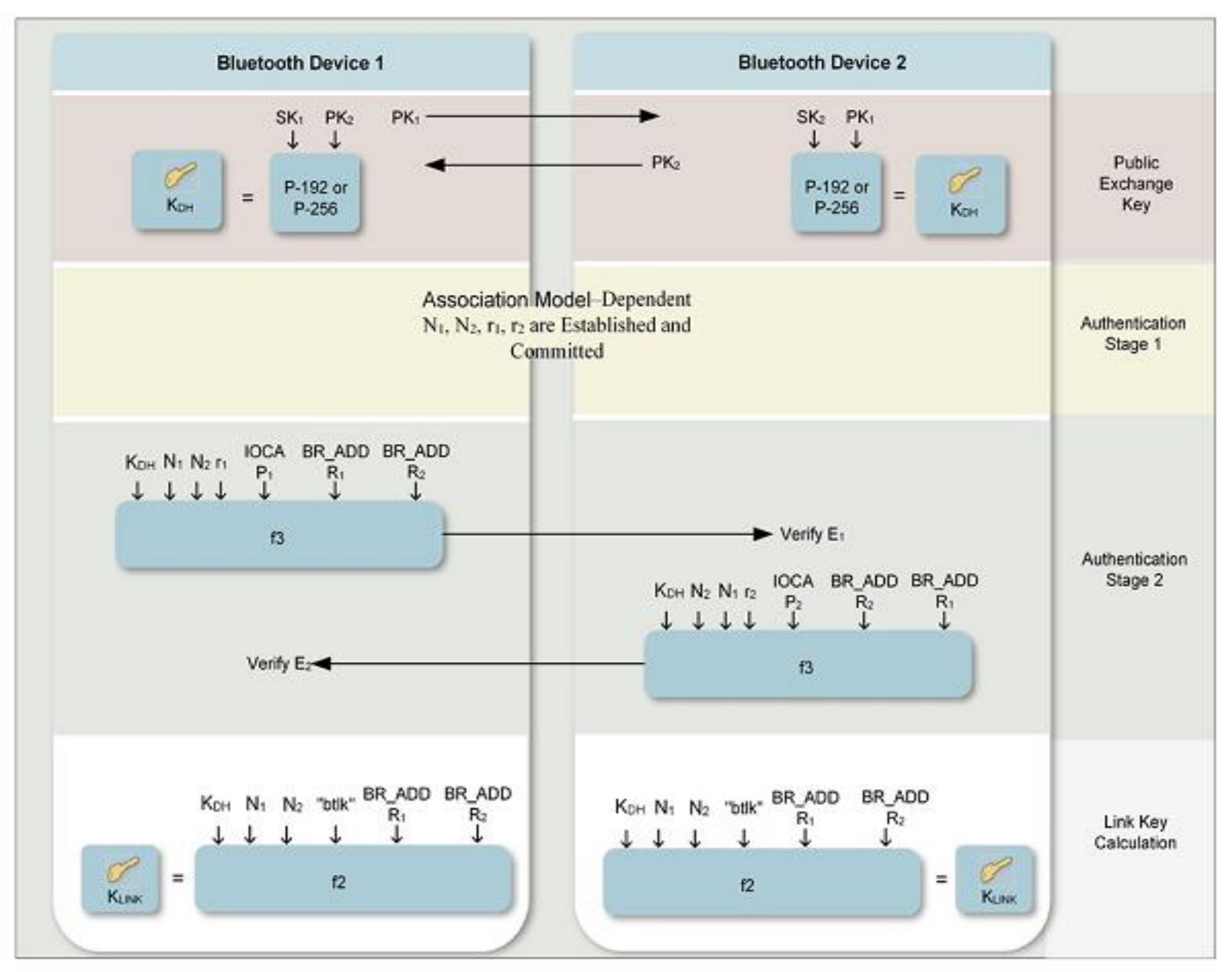
Device capabilities and SSP association models



Device 1	Device 2	Association model
DisplayYesNo	DisplayYesNo	Numeric comparison[a]
	DisplayOnly	Numeric comparison
	KeyboardOnly	Passkey Entry[a]
	NoInputNoOutput	Just works
DisplayOnly	DisplayOnly	Numeric comparison
	KeyboardOnly	Passkey entry[a]
	NoInputNoOutput	Just Works
KeyboardOnly	KeyboardOnly	Passkey entry[a]
	NoInputNoOutput	Just works
NoInputNoOutput	NoInputNoOutput	Just works

[a] The resulting link key is considered *authenticated*

Link key is established for SSP



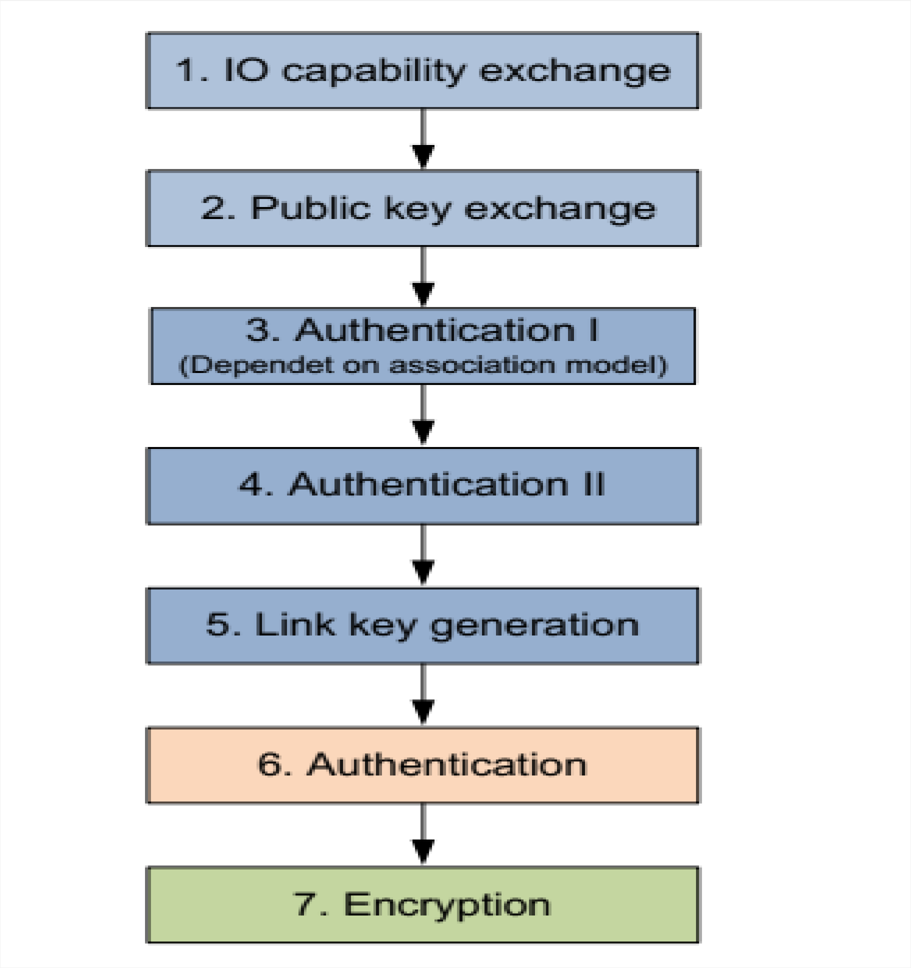
Bluetooth Device 1
Bluetooth Device 2
SK_1 PK_2
PK_1
SK_2 PK_1
PK_2
K_{DH} = P-192 or P-256
P-192 or P-256 = K_{DH}
Public Exchange Key
Association Model–Dependent N_1, N_2, r_1, r_2 are Established and Committed
Authentication Stage 1
K_{DH} N_1 N_2 r_1 IOCA P_1 BR_ADD R_1 BR_ADD R_2
f3
Verify E_1
K_{DH} N_2 N_1 r_2 IOCA P_2 BR_ADD R_2 BR_ADD R_1
f3
Verify E_2
Authentication Stage 2
K_{DH} N_1 N_2 "btlk" BR_ADD R_1 BR_ADD R_2
K_{LINK} = f2
K_{DH} N_1 N_2 "btlk" BR_ADD R_1 BR_ADD R_2
f2 = K_{LINK}
Link Key Calculation

This technique uses **ECDH public/private key pairs** rather than generating a symmetric key via a PIN.

Link key Generation steps

1. Each device generates its own ECDH public - private key pair.
2. When both devices support Secure Connections, P- 256 elliptic curves are used, else P - 192 curves are used.
3. Each device sends the public key to the other device.
4. The devices then perform stage 1 authentication which is dependent on the association model.
5. After this the first device computes a confirmation value E1 and sends it to the second device which checks the value.
6. If this succeeds, the second device does the same and sends its confirmation value E2 to the first device.
7. Assuming the E2 confirmation value checks out correctly, both devices compute the Link Key.

Phases in SSP



1. IO capability exchange

2. Public key exchange

3. Authentication I
(Dependet on association model)

4. Authentication II

5. Link key generation

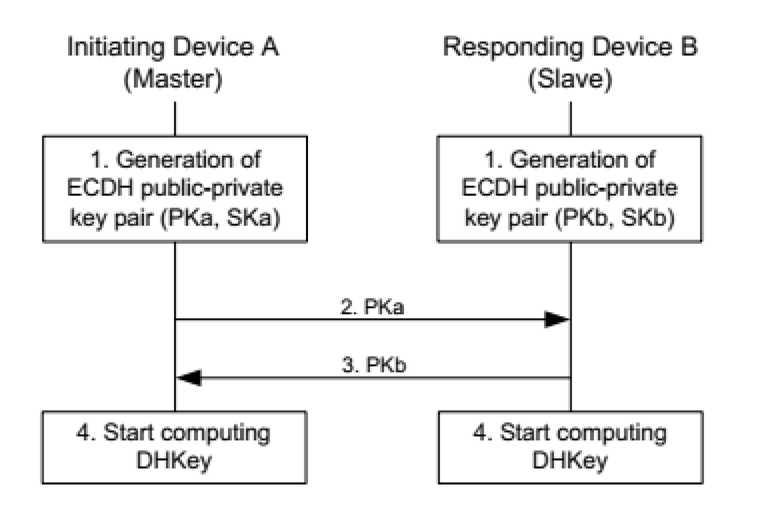
6. Authentication

7. Encryption

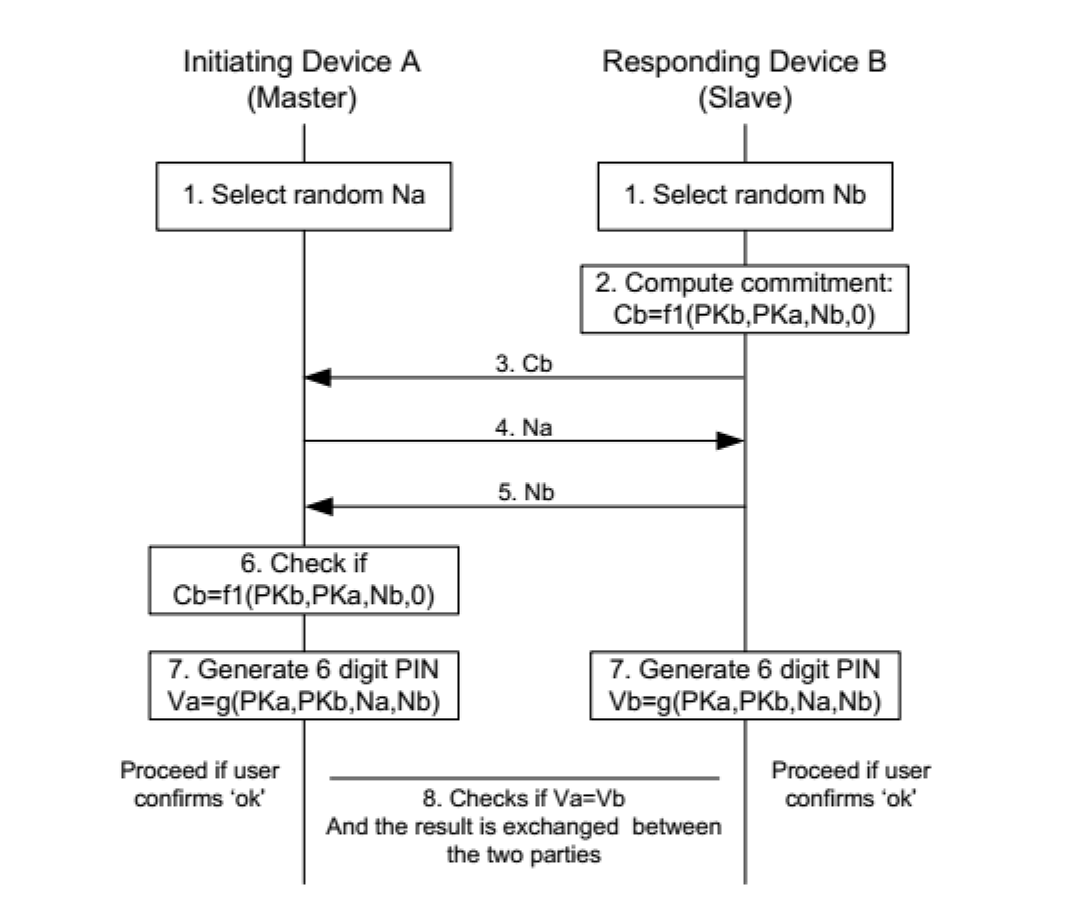
Phase 1 – IO capability exchange

Initially the Input/output capability of the two devices is exchanged in order to determine the appropriate association model to be used

Phase 2 – Public key exchange



Phase 3 – Authentication Stage 1



Initiating Device A
(Master)

Responding Device B
(Slave)

1. Select random Na

1. Select random Nb

2. Compute commitment:
Cb=f1(PKb,PKa,Nb,0)

3. Cb

4. Na

5. Nb

6. Check if
Cb=f1(PKb,PKa,Nb,0)

7. Generate 6 digit PIN
Va=g(PKa,PKb,Na,Nb)

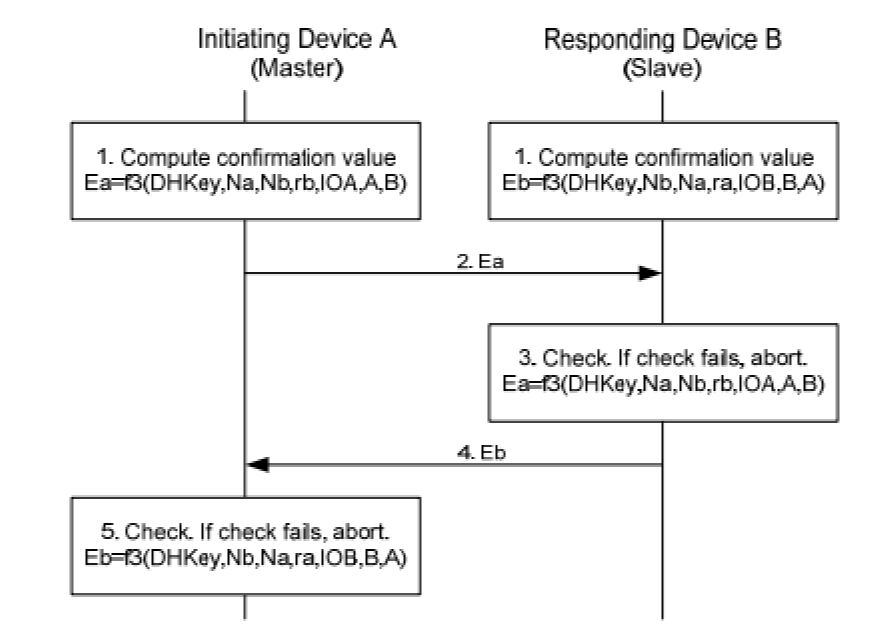
7. Generate 6 digit PIN
Vb=g(PKa,PKb,Na,Nb)

Proceed if user
confirms ‘ok’

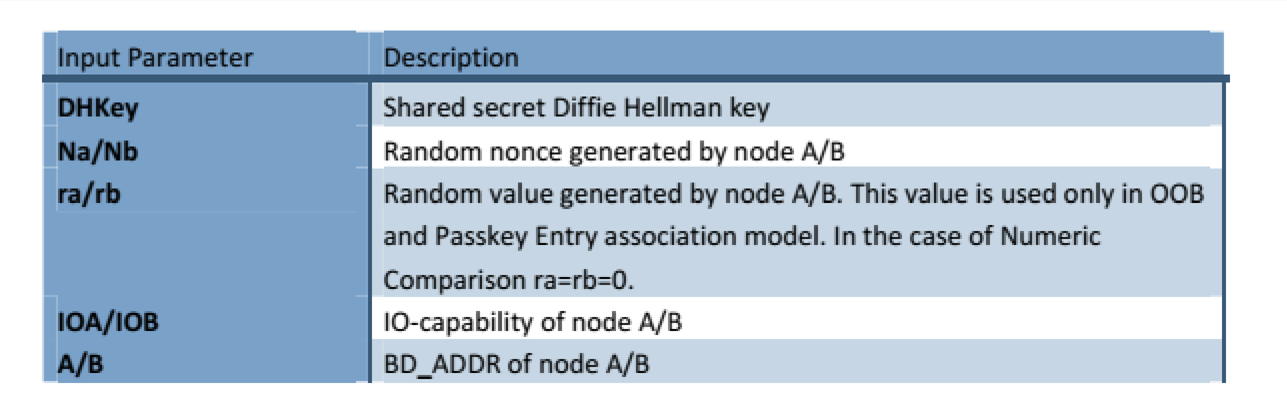
8. Checks if Va=Vb
And the result is exchanged between
the two parties

Proceed if user
confirms ‘ok’

Phase 4 – Authentication Stage 2

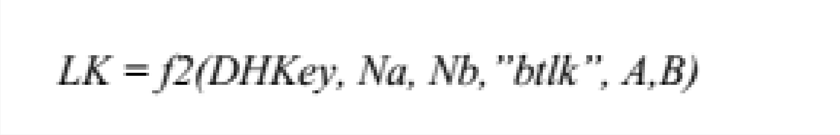

Initiating Device A
(Master)
Responding Device B
(Slave)
1. Compute confirmation value
Ea=f3(DHKey,Na,Nb,rb,IOA,A,B)
1. Compute confirmation value
Eb=f3(DHKey,Nb,Na,ra,IOB,B,A)
2. Ea
3. Check. If check fails, abort.
Ea=f3(DHKey,Na,Nb,rb,IOA,A,B)
4. Eb
5. Check. If check fails, abort.
Eb=f3(DHKey,Nb,Na,ra,IOB,B,A)

Input parameters to f3 function to generate a confirmation value



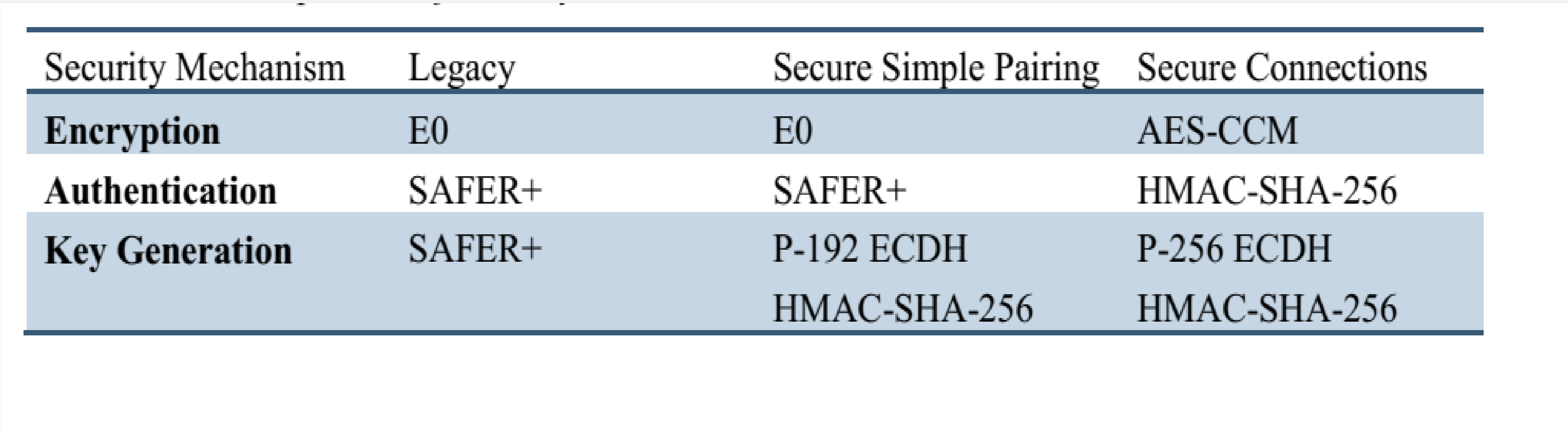
Input Parameter	Description
DHKey	Shared secret Diffie Hellman key
Na/Nb	Random nonce generated by node A/B
ra/rb	Random value generated by node A/B. This value is used only in OOB and Passkey Entry association model. In the case of Numeric Comparison ra=rb=0.
IOA/IOB	IO-capability of node A/B
A/B	BD_ADDR of node A/B

Phase 5 :Link key calculation



$$LK = f2(DHKey, Na, Nb, \text{"btlk"}, A,B)$$

Comparison of Security Schemes in Bluetooth



Security Mechanism	Legacy	Secure Simple Pairing	Secure Connections
Encryption	E0	E0	AES-CCM
Authentication	SAFER+	SAFER+	HMAC-SHA-256
Key Generation	SAFER+	P-192 ECDH HMAC-SHA-256	P-256 ECDH HMAC-SHA-256

Authentication

- ✓ The Bluetooth device authentication procedure is in the form of a **challenge–response scheme**

- ✓ Each device interacting in an authentication procedure can take the role of either the claimant or the verifier or both.

- ✓ The authentication procedure is of two types: **Legacy Authentication** and **Secure Authentication**.

- ✓ Legacy Authentication is performed when at least one device does not support Secure Connections.

- ✓ If both devices support Secure Connections, Secure Authentication is performed.

Legacy Authentication

This procedure is used when the link key has been generated using **PIN/Legacy Pairing** or **Secure Simple Pairing** using the **P-192 Elliptic Curve**.

The steps in the authentication process are as follows:

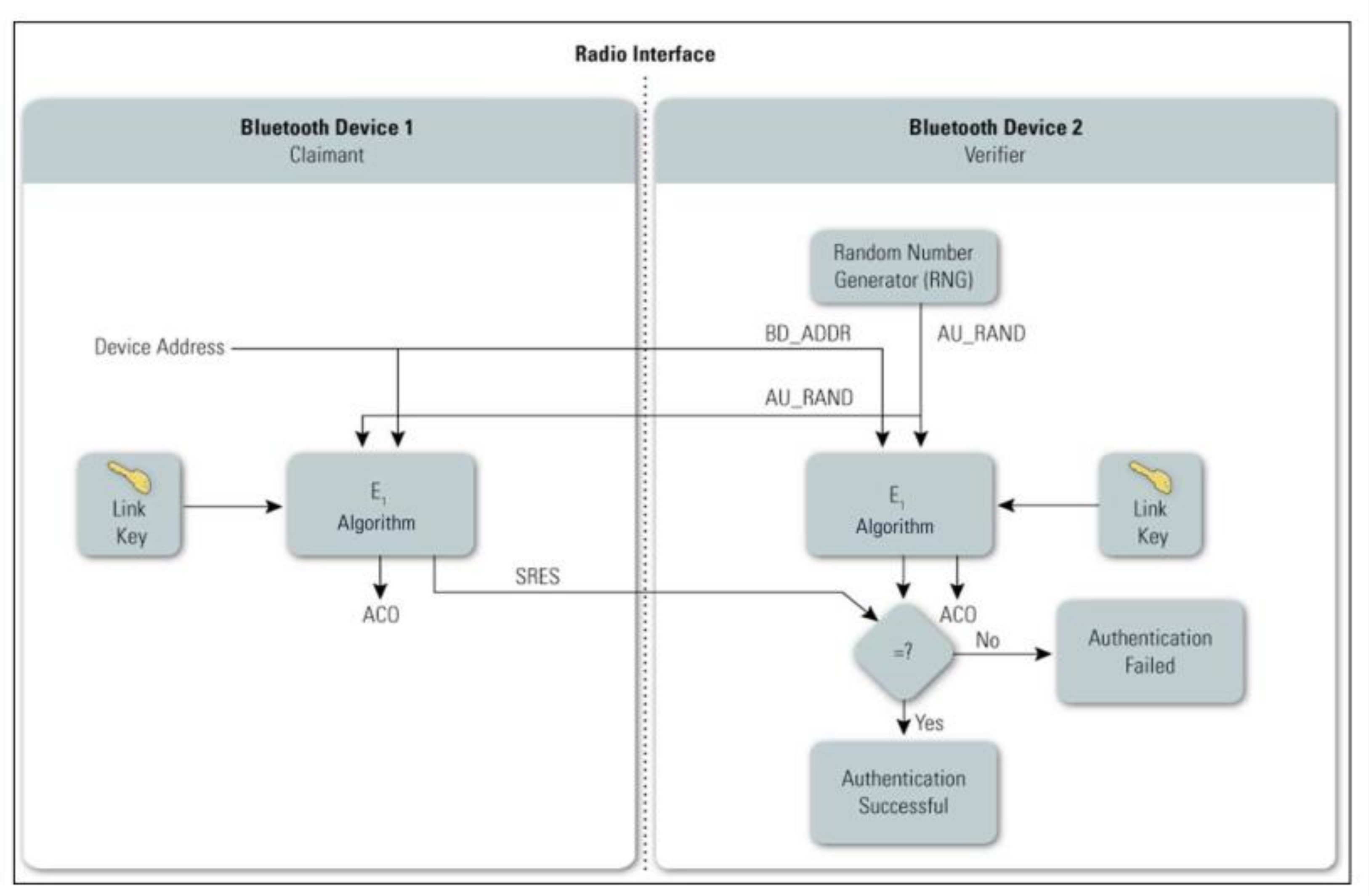
Step 1.

The verifier transmits a **128-bit** random challenge (AU_RAND) to the claimant.

Step 2.

The claimant uses the **E1 algorithm** to compute an authentication response using his or her unique 48-bit Bluetooth device address (**BD_ADDR**), the **link key**, and **AU_RAND** as inputs.

Bluetooth Legacy Authentication



The verifier performs the same computation.

- ✓ Only the **32** most significant bits of the **E1** output are used for authentication purposes.

- ✓ The remaining **96 bits** of the **128-bit** output are known as the **ACO value**, which will be used later as input to create the Bluetooth encryption key

Step 3

The claimant returns the most significant **32 bits** of the **E1** output as the computed response, the **Signed Response (SRES)**, to the verifier.

Step 4

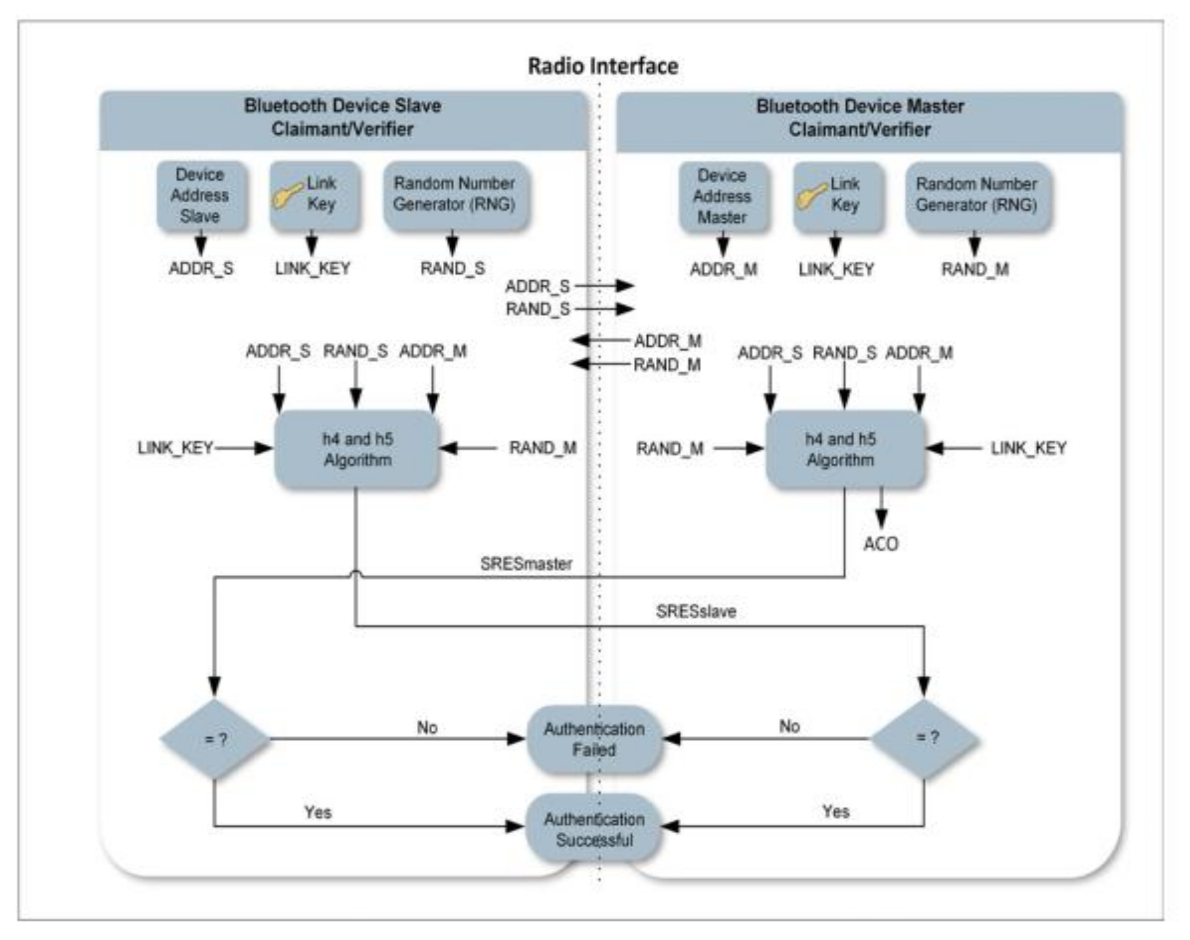
The verifier compares the **SRES** from the claimant with the value that it computed.

Step 5

If the two **32-bit values** are equal, the authentication is considered successful. If the two **32-bit values** are not equal, the authentication fails.

Secure Authentication

This procedure is used when the link key has been generated using **Secure Simple Pairing** with the **P-256** Elliptic Curve.



When the master initiates this authentication process, the steps are as follows:

Step 1.

The master transmits a **128-bit** random challenge **(RAND_M)** to the slave.

Step 2:

The slave transmits a **128-bit** random challenge **(RAND_S)** to the master

Step 3:

Both the master and slave use the **h4** and **h5** algorithms to compute their authentication responses using the unique 48-bit Bluetooth device address of the master **(ADDR_M),** the unique **48-bit** Bluetooth device address of the slave **(ADDR_S),** the link key, the **RAND_M**, and the **RAND_S** as inputs.

- ✓ Only the **32** most significant bits of the **h5** output are used for authentication purposes.
- ✓ The remaining **96 bits** of the **128-bit** output are known as the Authenticated Ciphering Offset (**ACO**) value, which will be used later as input to create the Bluetooth encryption key.

Step 4.

The slave returns the most significant **32 bits** of the **h5** output as the computed response, the **Signed Response (SRESslave)**, to the master.

Step 5:

The master returns the most significant **32 bits** of the **h5** output as the computed response, the **Signed Response (SRESmaster)**, to the slave.

Step 6:

The master and slave compare the **SRES** from each other with the value that they computed

Step 7:

If the two **32-bit** values are equal on both the master and slave, the authentication is considered successful.

If the two **32-bit** values are not equal on either the master or the slave, the authentication fails.

Confidentiality

Bluetooth has three Encryption Modes, but only two of them actually provide confidentiality. The modes are as follows:

Encryption Mode 1—No encryption is performed on any traffic

Encryption Mode 2- Individually addressed traffic is encrypted using encryption keys based on individual link keys; broadcast traffic is not encrypted.

Encryption Mode 3— All traffic is encrypted using an encryption key based on the master link key.