ESP32 Classic Bluetooth Security - Secure Simple Pairing

This document briefly describes how the device io capability and security mask affect the process of the Secure Simple Pairing. It will help you figure out how to set the parameter when calling `esp_bt_gap_set_security_param`, or the parameter `sec_mask` when you want to establish a connection associate a certain profile, for example, an SPP connection.

IO Capability

Input and output capabilities of a device are combined to generate its IO capabilities. The input capabilities are described in Table 1 The output capabilities are described in Table 2. Table 3 shows the combination of Input and Output Capability, which are defined as `ESP_BT_IO_CAP_OUT`, `ESP_BT_IO_CAP_IO`, `ESP_BT_IO_CAP_IN` and `ESP_BT_IO_CAP_NONE` in `esp_gap_bt_api.h`.

Input Capability	Description
No input	Device does not have the ability to indicate 'yes' or 'no'
Yes / No	Device has at least two buttons that can be easily mapped to 'yes' and 'no' or the device has a mechanism whereby the user can indicate either 'yes' or 'no' (see note below).
Keyboard	Device has a numeric keyboard that can input the numbers '0' through '9' and a confirmation. Device also has two buttons that can be easily mapped to 'yes' and 'no' or the device has a mechanism whereby the user can indicate either 'yes' or 'no' (see Note below).

Table 1 User Input Capability

Note: 'yes' could be indicated by pressing a button within a certain time limit otherwise 'no' would be assumed.

Output Capability	Description
No output	Device does not have the ability to display or communicate a 6 digit decimal number
Numeric output	Device has the ability to display or communicate a 6 digit decimal number

Table 2 User Output Capability

	No Output	Numeric Output
No input	NoInputNoOutput (`ESP_BT_IO_CAP_NONE`)	DisplayOnly (`ESP_BT_IO_CAP_OUT`)
Yes / No	NoInputNoOutput (`ESP_BT_IO_CAP_NONE`)	DisplayYesNo (`ESP_BT_IO_CAP_IO`)
Keyboard	KeyboardOnly (`ESP_BT_IO_CAP_IN`)	DisplayYesNo (`ESP_BT_IO_CAP_IO`)

Table 3 IO Capability Mapping

Secure Simple Pairing

The primary goal of Secure Simple Pairing is to simplify the pairing procedure for the user. Secondary goals are to protect against passive eavesdropping and man-in-the-middle (MITM) attacks (active eavesdropping). It uses four association models referred to as Numeric Comparison, Just Works, Out Of Band, and Passkey Entry.

- Numeric Comparison: both devices are capable of displaying a six digit number and both are capable of having the user enter "yes" or "no"
- Just Works: at least one of the devices does not have a display capable of displaying a six digit number nor does it have a keyboard capable of entering six decimal digits
- Out Of Band: use Out of Band mechanism to discover the devices and transfer cryptographic numbers
- Passkey Entry: one device has input capability but does not have the capability to display six digits and the other device has output capabilities.

Note: Just Works can be considered as a special kind of Numeric Comparison with automatic accept allowed.

There are four stages defined in the Secure Simple Pairing LM process:

- IO Capability exchange
- Public key exchange
- Authentication stage 1
- Authentication stage 2

IO Capability Exchange

The IO_Capability_Request_Reply command is used to reply to an IO Capability Request event from the controller, and specifies the current I/O capabilities of the host. The content of this command is shown as Table 4. | Command | OCF | Command Parameters | Return Parameters | | ------------------------------ | ------
| -- | ----------------- | |
HCI_IO_Capability_Request_Reply | 0x002B | BD_ADDR,
IO_Capability,
OOB_Data_Present,
Authentication_Requirements | Status,
BD_ADDR | *Table 4 IO_Capability_Request_Reply command*

Authentication Requirements

If an authenticated link key is not required by the Host, the Authentication Requirements parameter may be set to one of the following:

- MITM Protection Not Required – No Bonding
- MITM Protection Not Required – Dedicated Bonding
- MITM Protection Not Required – General Bonding

Note: If both Hosts set the Authentication_Requirements parameter to one of the above values, the Link Managers shall use the numeric comparison authentication procedure and the hosts shall use the Just Works Association Model.

If an authenticated link key is required by the Host, the Authentication Requirements parameter shall be set to one of the following:

- MITM Protection Required – No Bonding
- MITM Protection Required – Dedicated Bonding
- MITM Protection Required – General Bonding

Note: In addition, the following requirements apply:

1. If one or both hosts set the Authentication Requirements parameter to one of the above values, the Link Managers shall use the IO_Capability parameter to determine the authentication procedure.
2. A Host that sets the Authentication_Requirements parameter to one of the above values shall verify that the resulting Link Key type meets the security requirements requested.

IO Capability and Authentication Procedure Mapping

If one or both devices have set the Authentication_Requirements parameter to one of the MITM Protection Required options, the IO capabilities are mapped to the authentication method as defined in the following table. A host has set the MITM Protection Required options shall verify that the resulting Link Key is an Authenticated Combination Key.

	Responder			
Initiator	**DisplayOnly**	**DisplayYesNo**	**KeyboardOnly**	**NoInputNoOutput**
DisplayOnly	Just Works Unauthenticated	Just Works Unauthenticated	Passkey Entry Authenticated	Just Works Unauthenticated
DisplayYesNo	Just Works Unauthenticated	Numeric Comparison Authenticated	Passkey Entry Authenticated	Just Works Unauthenticated

KeyboardOnly	Passkey Entry Authenticated	Passkey Entry Authenticated	Passkey Entry(both need enter) Authenticated	Just Works Unauthenticated
NoInputNoOutput	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated

Table 5 IO Capability and Authentication Procedure Mapping

Security Mask

Security Mask has two function. First, to decide whether or not required MITM protection for Authentication_Requirements parameter. Second, to set the security level for authentication stage. On ESP32, the Security Masks for A2DP and HFP are hard coded with `BTA_SEC_AUTHENTICATE` and `(BTA_SEC_AUTHENTICATE| BTA_SEC_ENCRYPT)`. The following table shows `sec_mask` value for SPP and its definition in BTA layer of ESP32 Bluetooth stack.

SPP sec_mask	BTA layer sec_mask	value	Description
`ESP_SPP_SEC_NONE`	`BTA_SEC_NONE`	0x0000	No security
`ESP_SPP_SEC_AUTHORIZE`	`BTA_SEC_AUTHORIZE`	0x0001	Authorization required (only needed for out going connection) (not support)
`ESP_SPP_SEC_AUTHENTICATE`	`BTA_SEC_AUTHENTICATE`	0x0012	Authentication required
`ESP_SPP_SEC_ENCRYPT`	`BTA_SEC_ENCRYPT`	0x0024	Encryption required
`ESP_SPP_SEC_MODE4_LEVEL4`	`BTA_SEC_MODE4_LEVEL4`	0x0040	Mode 4 level 4 service, i.e. incoming/outgoing MITM and P-256 encryption (not support)
`ESP_SPP_SEC_MITM`	`BTA_SEC_MITM`	0x3000	Man-In-The_Middle protection
`ESP_SPP_SEC_IN_16_DIGITS`	`BTA_SEC_IN_16_DIGITS`	0x4000	Min 16 digit for pin code

Table 6 ESP32 SPP Security Mask

Note:

1. When `ESP_SPP_SEC_AUTHENTICATE` is set, then Security Manager set `ESP_SPP_SEC_MITM` automatically.
2. When `ESP_SPP_SEC_ENCRYPT` is set, then Security Manager set `ESP_SPP_SEC_AUTHENTICATE` automatically.

Security Mode 4

The ESP32 Security Manager supports Security Mode 4 which means that it enforces security requirements before it attempts to access services offered by a remote device and before it grants access to services it offers to remote devices. Security Mode 4 has 4 level of security.

Security Level Required for Service	Link Key type required for remote devices	Comments	IO Capability & Security Mask Setting
Level 4(not support) * MITM protection required * Encryption required * User interaction acceptable	Authenticated (P-256 based Secure Simple Pairing and Secure Authentication)	Highest Security Only possible when both devices support Secure Connections	/
Level 3 * MITM protection required * Encryption required * User interaction acceptable	Authenticated	High Security	IO: >= DisplayYesNo Mask: ESP_SPP_SEC_MITM | BTA_SEC_ENCRYPT
Level 2 * MITM protection not necessary	Unauthenticated	Medium Security	IO: >= DisplayYesNo Mask: BTA_SEC_ENCRYPT

* Encryption desired			
Level 1 * MITM protection not necessary * Encryption desired * Minimal user interaction desired	Unauthenticated	Low Security	IO: >= NoInputNoOutput Mask: BTA_SEC_ENCRYPT
Level 0 * MITM protection not necessary * Encryption not necessary * No user interaction desired	None	Permitted only for SDP and service data sent via either L2CAP fixed signaling channels or the L2CAP connectionless channel to PSMs that correspond to service class UUIDs which are allowed to utilize Level 0.	/

Table 7 Security Mode 4 Security Level mapping to link key requirements

Note: Security Mode 4 always requires **authentication** and **encryption** over establishment of L2CAP connection on ESP32.

ESP32 Secure Simple Pairing performs legacy authentication which means mutual authentication is achieved by first performing the authentication procedure in one direction and then immediately performing the authentication procedure in the opposite direction. So, when the initiator does not require MITM protection, the generated link key type in this direction is unauthenticated. In other direction, if the responder need MITM protection and have the `sec_mask` of `ESP_SPP_SEC_AUTHENTICATE`, then the unauthenticated link key will be regenerated and upgraded to an authenticated one. Each time the link key is regenerated, user will receive a `ESP_BT_GAP_AUTH_CMPL_EVT` event.