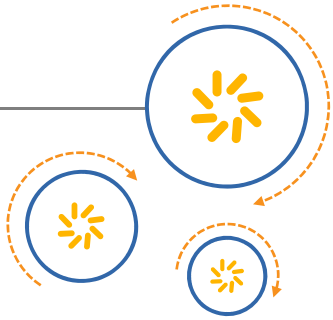
Qualcomm Technologies International, Ltd.



Confidential and Proprietary – Qualcomm Technologies International, Ltd.

(formerly known as Cambridge Silicon Radio Ltd.)

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies International, Ltd.or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies International, Ltd.

Any software provided with this notice is governed by the Qualcomm Technologies International, Ltd. Terms of Supply or the applicable license agreement at https://www.csrsupport.com/CSRTermsandConditions.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. All Qualcomm Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited.

Qualcomm Technologies International, Ltd.
Churchill House
Cambridge Business Park
Cambridge, CB4 0WZ
United Kingdom

CSR

Push every boundary.™

CSR µEnergy®

CSR101x Firmware API Extensions

Application Note

Issue 2

Document History

Revision	Date	History
1	24 MAY 13	Original publication of this document
2	29 JAN 14	Updated to new CSR branding References Bluetooth Core Specification 4.1

Contacts

General information	www.csr.com
Information on this product	sales@csr.com
Customer support for this product	www.csrsupport.com
More detail on compliance and standards	product.compliance@csr.com
Help with this document	comments@csr.com

Trademarks, Patents and Licences

Unless otherwise stated, words and logos marked with ™ or ® are trademarks registered or owned by CSR plc and/or its affiliates.

Bluetooth® and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc. and licensed to CSR.

Other products, services and names used in this document may have been trademarked by their respective owners.

The publication of this information does not imply that any licence is granted under any patent or other rights owned by CSR plc or its affiliates.

CSR reserves the right to make technical changes to its products as part of its development programme.

While every care has been taken to ensure the accuracy of the contents of this document, CSR cannot accept responsibility for any errors.

Use of this document is permissible only in accordance with the applicable CSR licence agreement.

Safety-critical Applications

CSR's products are not designed for use in safety critical devices or systems such as those relating to: (i) life support; (ii) nuclear power; and/or (iii) civil aviation applications, or other applications where injury or loss of life could be reasonably foreseeable as a result of the failure of a product. The customer agrees not to use CSR's products (or supply CSR's products for use) in such devices or systems.

Performance and Conformance

Refer to www.csrsupport.com for compliance and conformance to standards information.

Contents

Tables, Figures and Equations

1. Scope

This document describes how an application should use the Challenge Response API in the CSR µEnergy® firmware.

2. Introduction

The Challenge Response feature within the CSR µEnergy® firmware provides a level of protection against Man in the Middle (MITM) security attacks. In particular the feature protects against the scenario where neither the Master (Bluetooth Smart) nor the Slave (Bluetooth Smart) is within range of each other, but they are both within range of a third Bluetooth Smart device known as the Man in the Middle. The Man in the Middle relays the radio packets transmitted by the two devices to make it appear they are within range of each other.

Use of the Challenge Response feature is optional, however it is recommended in situations where protection is required against the above attack scenario.

3. Limitations of Use

The Challenge Response feature must be performed at times when no other application data transfer is taking place between the Master and the Slave.

The Challenge Response procedure should be performed over a link with a short connection interval to minimise the delay in receiving responses.

The Challenge Response procedure should be performed as soon as possible after connection establishment and link encryption.

The Master and Slave can repeat the Challenge Response procedure after connection establishment. Currently there are no known attack scenarios that require the procedure to be repeated.

4. Challenge Response Procedure

The Challenge Response procedure consists of the following sequence:

1. Establish an encrypted BLE link, see section 4.1
2. The Master application starts T_{MAX}, where T_{MAX} is the maximum duration for the procedure to complete, see section 4.2
3. The Master application instructs the Master side firmware to perform RX timing calculations, see section 4.3
4. The Master application transmits the Challenge 1 packet and starts application timer T_1, see section 4.4
5. The Slave calculates the response as soon as it receives the challenge. The firmware delays sending the response packet over the air until the next challenge is received, see section 4.5
6. T_1 expires and the Master application transmits the Challenge 2 packet
7. When the Slave application receives Challenge 2 it instructs the Slave side firmware to resume normal transmission behaviour, i.e. data transmissions are no longer delayed until further packets are received, see section 4.7
8. The Master side firmware invokes the Master application's LM Event Handler, see section 4.8
9. The Master application instructs the Master side firmware to cease performing RX timing calculations, see section 4.9
10. The Master application determines if the Challenge Response procedure was successful, see section 4.9

These steps are described more fully in the following sub sections.

Two sequence diagrams for the procedure are illustrated below; one demonstrates the Master as the Server, the other demonstrates the Master as the Client.

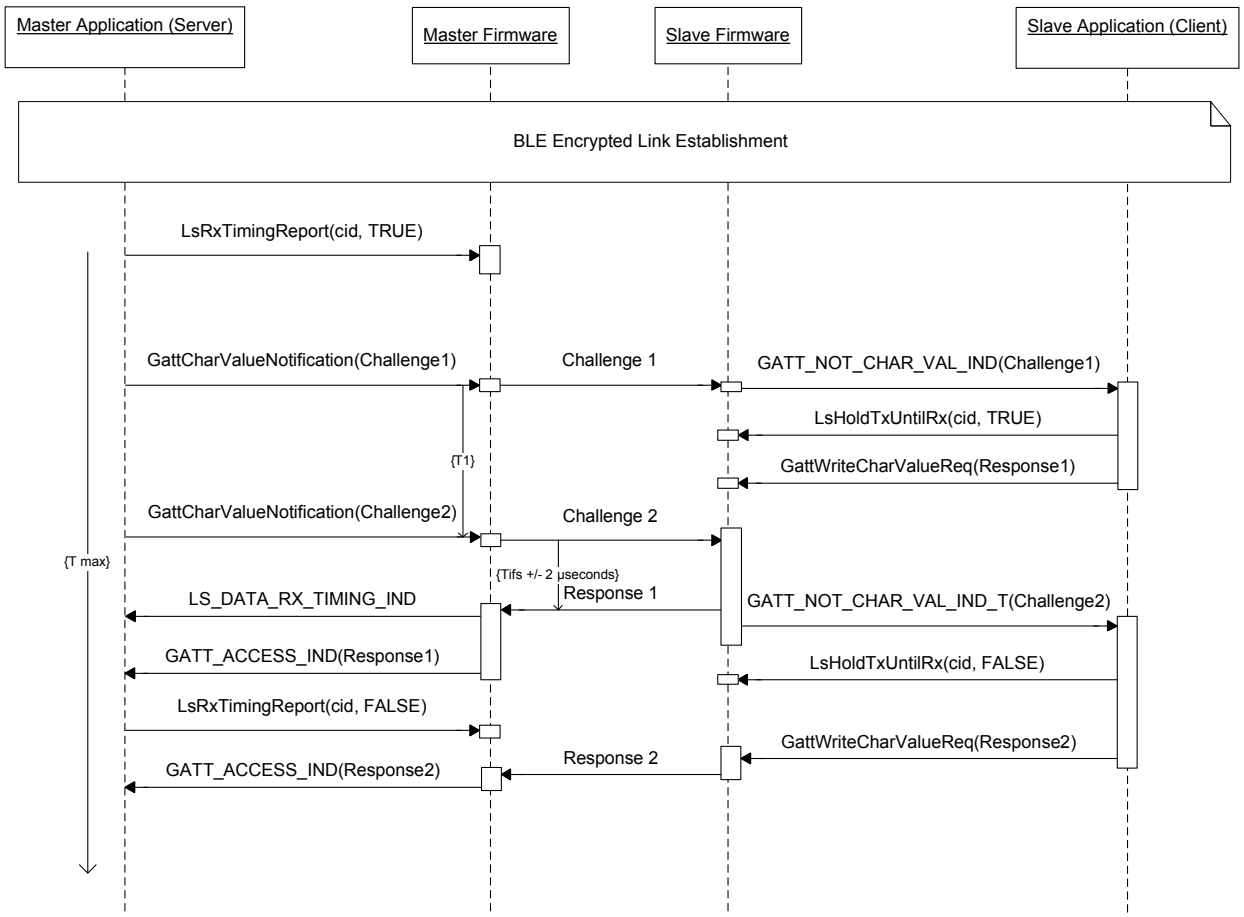


Figure 4.1: Challenge Response Procedure with the Master as the Server

Notes:

- The Application will regard the Challenge Response procedure as having failed if it has not completed before T_{MAX} expires.
- T_1 is the time period that the Master application waits between sending Challenge 1 and Challenge 2.
- T_{IFS} is the minimum duration between the Master application sending Challenge 2 and receiving Response 1, T_{IFS} shall be 150 microseconds, see *Bluetooth Core Specification v4.1*.

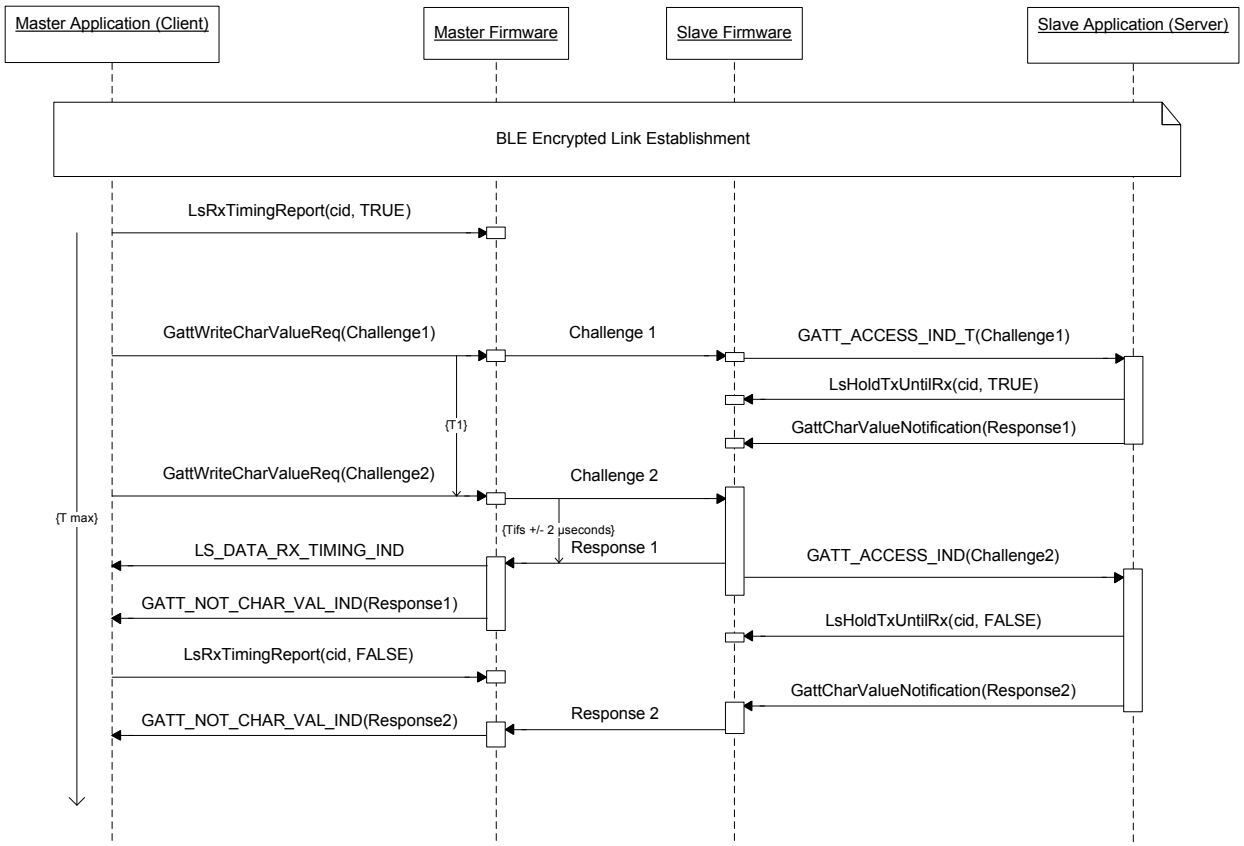


Figure 4.2: Challenge Response Procedure with the Master as the Client

Notes:

- The Application will regard the Challenge Response procedure as having failed if it has not completed before T_{MAX} expires.
- T_1 is the time period that the Master application waits between sending Challenge 1 and Challenge 2.
- T_{IFS} is the minimum duration between the Master application sending Challenge 2 and receiving Response 1, T_{IFS} shall be 150 microseconds, see *Bluetooth Core Specification v4.1*.

4.1. Establish an Encrypted Bluetooth Smart Link

An encrypted link is established using one of the mechanisms available for Bluetooth Smart link encryption, for example Just Works or Passkey Entry, see *Bluetooth Core Specification v4.1*.

4.2. Maximum Challenge Response Procedure Duration: T_{MAX}

The maximum duration of the Challenge Response Procedure (T_{MAX}) is application specific. If the Master application has not received and processed the `LS_DATA_RX_TIMING_IND` within this period then the procedure has failed and the Master application must instruct the firmware to cease performing measurement calculations as described in section 4.9.

If the Slave application has not received Challenge 2 when T_{MAX} has expired then the Slave application must instruct the firmware to cancel the hold on data transmissions.

4.3. Instruct the Firmware to Perform RX Timing Calculations

Instruct the Bluetooth Smart firmware to perform Challenge Response timing calculations by invoking `LsRxTimingReport(cid, TRUE)`.

4.4. The Master Application Transmits a Challenge Packet

The Master application generates a Challenge packet which contains a maximum of 20 octets of data. The data within the Challenge packet should be changed in a non-predictable manner each time a Challenge packet is constructed to protect against replay attacks.

4.4.1. Transmit a Challenge when the Master Application is a GATT Client

If the Master application is in a client role then it transmits the Challenge packet by invoking `attWriteCharValueReq()`.

4.4.2. Transmit a Challenge when the Master Application is a GATT Server

If the Master application is in a server role then it transmits the Challenge packet by invoking `GattCharValueNotification()`.

4.5. The Slave Delays All Data Transmission Until the Next Packet is Received

The Slave application instructs the firmware to delay transmission of all data packets until it receives the next packet. The Slave application does this by invoking `LsHoldTxUntilRx(cid, TRUE)`.

4.6. The Slave Application Responds to Received Challenges

The Slave application generates a Response packet for each received Challenge, the Response packet shall contain a maximum of 20 octets of data. The data within the Response packet is generated from the data received in the Challenge packet using a cryptographic mechanism known to both the Master and Slave.

4.6.1. Transmit a Response when the Slave Application is a GATT Client

If the Slave application is in a client role then it transmits the Response packet by invoking `GattWriteCharValueReq()`.

4.6.2. Transmit a Response when the Slave Application is a GATT Server

If the Slave application is in a server role then it transmits the Response packet by invoking `GattCharValueNotification()`.

4.7. The Slave Resumes Normal Transmission Behaviour

The Slave application instructs the firmware to resume normal transmission behaviour by invoking `LsHoldTxUntilRx(cid, FALSE)`.

4.8. Firmware Sends Packet Timing Data to Application

The application must handle the new `LS_DATA_RX_TIMING_IND` event. The application determines if the timing parameters in the `LS_DATA_RX_TIMING_IND` are within the limits specified in section 5.1.1.

4.9. Determining that the Challenge Response Procedure is Complete

The Challenge Response procedure is complete when any of the following conditions has occurred:

1. The Master application has received and processed the timing data contained in the `LS_DATA_RX_TIMING_IND` event as described in section 5.1.1.
2. The Master application has terminated the Challenge Response procedure after TMAX has expired.

If any of these conditions are satisfied, the Master application must call `LsRxTimingReport(cid, FALSE)` causing the timing measurement reports sent by the firmware to cease.

5. Application Supplied Functions

The Master and Slave applications must each provide and register an implementation of an LM event handler function.

The Master side Bluetooth Smart firmware invokes this function and provides the timing data associated with the Challenge Response procedure within an `LS_DATA_RX_TIMING_IND_T` structure. This function is not specific to the Challenge Response procedure; it is invoked by the firmware whenever any LM event occurs.

Note:

This function is also used by the Slave application, however the `LS_DATA_RX_TIMING_IND` event is never sent to the Slave application.

The Master and Slave applications must each define a customised service with characteristics for registering notifications and write procedures. The applications use these services to manage the Challenge Response procedure.

5.1. LM Event Handler Function

5.1.1. LS_DATA_RX_TIMING_IND

The firmware invokes the registered LM Event Handler Function and supplies a valid event code using the `lm_event_code` parameter and a valid pointer to an `LM_EVENT_T` type using the `event_data` parameter.

If the `lm_event_code` is `LS_DATA_RX_TIMING_IND` then the `event_data` points to an `LS_DATA_RX_TIMING_IND_T` structure, that is the `event_data` should be cast to a `(LS_DATA_RX_TIMING_IND_T*)`.

The `LS_DATA_RX_TIMING_IND_T` contains the following fields:

- `time16 tx_duration`: the duration in microseconds of the most recent packet transmission preceding the currently received packet,
- `time48 tx_event_offset`: the offset in microseconds of the most recent packet transmission within the current connection event,
- `time48 tx_transmit_offset`: the offset in microseconds of the most recent transmission from first possible transmit opportunity.

When performing the Challenge Response procedure the Master application must verify that these values are within the specified limits. If any value is outside of these limits then the Challenge Response procedure has failed and the Master application must take appropriate action. The specified limits are:

- `tx_duration` must be greater than 80 microseconds.
- `tx_event_offset` must be less than or equal to 2 microseconds.
- `tx_transmit_offset` must be less than or equal to 2 microseconds

Document References

Document	Reference
Bluetooth Core Specification Version 4.1	www.bluetooth.org/Technical/Specifications/adopted.htm

Terms and Definitions

API	Application Programming Interface
BlueCore®	Group term for CSR's range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
GATT	Generic Attribute Profile
LM	Link Manager
MITM	Man in the Middle, see *Bluetooth Core Specification v4.1*
RX	Receive or Receiver
T_{IFS}	Inter-Frame Spacing, see *Bluetooth Core Specification v4.1*
T_{MAX}	The maximum time that the Challenge Response procedure should take to complete
TX	Transmit or Transmitter