The goal of this example is to remove the Over The Air (OTA) write for the Devic Name Characteristic.

Start with the OTA write enabled and from application controller . (In the nRFgo studio settings in nRF8001 configuration -> GAP settings -> Writeable Device name.)

To remove the OTA write alone, we need to change the Characteristic properties of the Device Name characteristic to only Read.

We also need to change the permissions of the Device Name Characteristic value to have only the Read Permission.

=

CRC used:

Polynomial for CRC-16-CCITT https://upload.wikimedia.org/wikipedia/en/math/d/0/2/d0279fbd8e8844f8cbef81a76f006b93.png([X.25](https://en.wikipedia.org/wiki/X.25), [V.41](https://en.wikipedia.org/wiki/V.41), [HDLC](https://en.wikipedia.org/wiki/HDLC), [XMODEM](https://en.wikipedia.org/wiki/XMODEM), [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth), [PACTOR](https://en.wikipedia.org/wiki/PACTOR), [SD](https://en.wikipedia.org/wiki/Secure_Digital_card), many others; known as CRC-CCITT)

Initial seed value for the CRC is FFFF

The CRC is done on all setup ACI packets including the opcode and length excluding only the last 2 bytes which is the CRC.

=

The format of all the Setup messages generated by the nRFgo studio is

<Len> <Opcode = Setup i.e. 0x06> <Target> <Offset within the Target>

The Target we want to modify is the "ATTDB".

The value of target "ATTDB" = 0x20.

The device name is usually the first message in this target so the offset we use is 0x00.

Look for the message <Len> 0x06 0x20 0x00.

{0x00,\

{\

{ header } { attdb\_head\_t }{handle}{UUID }{Value } **0x1f,0x06,0x20,0x00**,0x04,0x04,0x02,0x02,0x00,0x01,0x28,0x00,0x01,0x00,0x18,0x04,0x04,0x05,0x05,0x00,\

0x02,0x28,0x03,0x01,0x0e,0x03,0x00,0x00,0x2a,0x04,0x14,0x14,\

},\

},\

=

Change 0x14 to 0x04 for the permission of the value of the Device Name to be changed to Read.

After this is done. Regenerate the CRC and update the CRC in the setup data. The last 2 bytes of the setup data is the CRC. The CRC is done on all setup ACI packets including the length and opcode excluding only the last 2 bytes which is the CRC. Ignore the 0x00 for all CRC calculations.

==

To change the Read permission bits of any 1 Characteristic value (with Read in its Characteristic property) from "plaintext" (which is created from the GATT Server marked as "No Security Required", to "authenticated encrypted" b11.

Modify the 2 Read Permissions bits for the Characteristic Value

uint8\_t rd:2;         /\*\*< Read  permisions (none, plaintext, authenticated plaintext, authenticated encrypted) \*/

You also need to set the Device setting bits which map to nRF8001 configuration -> Device Security back to “Security Required”

You can find the bits by also comparing the Setup between "Security Required" and "No Security Required".

The message uses the structure as below

/\*\* @file

\*

\* @brief This header file contains ATT DB internl data types

\* Inside the ATTDB, handles and UUIDs are stored in BE and values in LE

\* Please note that the ATT protocol requires data in LE format

\* ATTDB\_P2D macros store data in BE, for handles and UUIDs

\* uuid\_encode with UUID\_P2U\_UINT16 stores data in LE, for values

\*/

/\*\* @brief Attribute permissions and length field

\*/

typedef struct {

/\* octet 0 \*/

uint8\_t mlen1:2; /\*\*< MSb of max len, refer Macros section in attdb.c \*/

uint8\_t valid:1; /\*\*< Valid record \*/

uint8\_t bcast:1; /\*\*< Broadcastable record, functionality implemented in upper layer, present for info purposes \*/

uint8\_t notify:1; /\*\*< Notifiable record, functionality implemented in upper layer, present for info purposes \*/

uint8\_t ind:1; /\*\*< Indicatable record, functionality implemented in upper layer, present for info purposes \*/

uint8\_t flighty:1; /\*\*< Flighty/volatile record \*/

uint8\_t locked:1; /\*\*< Lock (RFI) \*/

/\* octet 1 \*/

uint8\_t vlen1:2; /\*\*< MSb of actual len, refer Macros section in attdb.c \*/

uint8\_t rd:2; /\*\*< Read permisions (none, plaintext, authenticated plaintext, authenticated encrypted) \*/

uint8\_t wr:2; /\*\*< Write permisions (none, plaintext, authenticated plaintext, authenticated encrypted) \*/

uint8\_t ard:1; /\*\*< Read authorization (RFI) \*/

uint8\_t awr:1; /\*\*< Write authorization (RFI) \*/

/\* octet 2 \*/

uint8\_t mlen0:8; /\*\*< LSb of max len, refer Macros section in attdb.c \*/

/\* octet 3 \*/

uint8\_t vlen0:8; /\*\*< LSb of actual len, refer Macros section in attdb.c \*/

} attdb\_head\_t;

/\*\* @struct attdb\_record\_head\_t

\*

\* @brief Attribute DB values table record header type

\*/

typedef struct {

/\*\* Attribute DB permissions and length \*/

attdb\_head\_t head;

/\*\* Attribute DB entry handle \*/

attdb\_handle\_t handle;

/\*\* UUID of the attribute \*/

uuid\_t uuid;

/\*\* Attribute Value \*/

uint8\_t value[1];

} attdb\_record\_head\_t;

Overall format of the nRF8001 Setup area

==

/\*\*

\* @struct dm\_mrs\_data\_t

\* @brief Memory Retention Setup Data

\*/

typedef struct

{

dm\_setup\_signature\_t setup\_signature; /\* This is the first byte written/read to/from OTP \*/

dm\_dev\_settings\_t device\_settings;

uint8\_t CLASS\_XDATA \*attdb\_heap;

dm\_service\_t CLASS\_XDATA \*rem\_svcs;

dm\_attr\_t CLASS\_XDATA \*attrs;

uint128\_le\_t CLASS\_XDATA \*vs\_uuids;

dm\_ext\_attr\_t CLASS\_XDATA \*ext\_attrs;

dm\_custom\_ad\_type\_t CLASS\_XDATA \*cust\_ad\_types;

} dm\_mrs\_data\_t;

The target we want to modify is the custom advertising type.

The value of target “CUST\_AD\_TYPES” is 0x70

Look for the message <Len> 0x06 0x70 0x00 <Custom advertising data follows>

Interpret the bytes following the above using the structure below.

typedef struct  
{  
uint8\_t type;  
uint8\_t len;  
uint8\_t adv\_data[ACI\_CUSTOM\_AD\_TYPE\_MAX\_DATA\_LENGTH];  
} dm\_custom\_ad\_type\_t;

#define ACI\_CUSTOM\_AD\_TYPE\_MAX\_DATA\_LENGTH 20