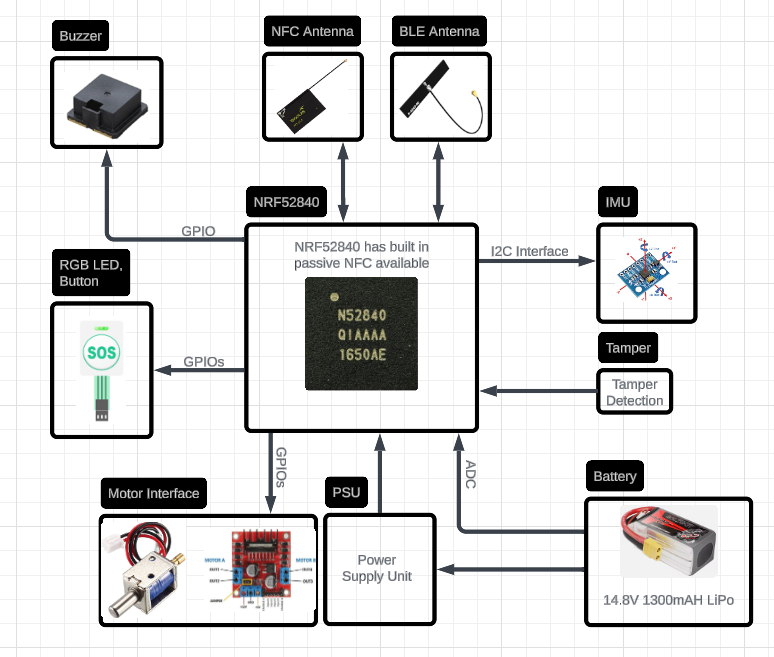
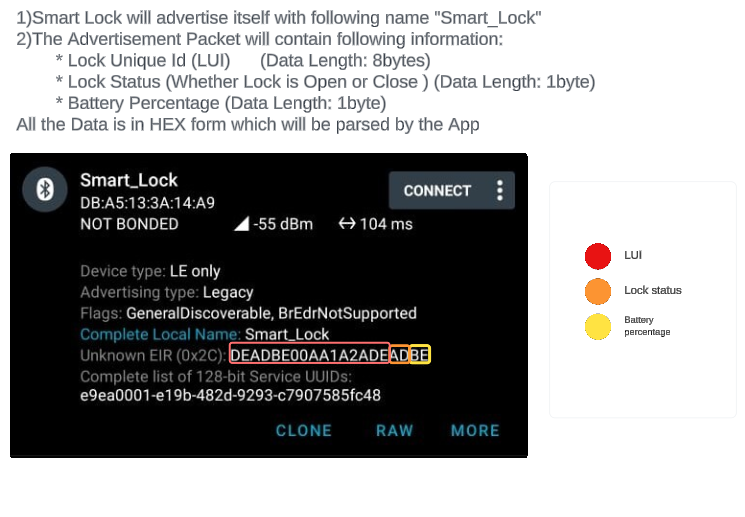
**Smart Lock Complete System**

**Block Diagram:**



**Advertise Packet:**



|  |  |  |
| --- | --- | --- |
| Name | Description | Data Length ( Bytes ) |
| Lock Unique ID (LUI) | This is a unique ID which will be different for each lock and will be stored on Server at flash time. | 8 |
| Lock Status | Shows Status of Lock weather it is open or close In case of close 0x01 and open 0x02 | 1 |
| Battery Percentage | Shows battery level in percentage Example: if battery is 22% here it will only show the numeric part | 1 |

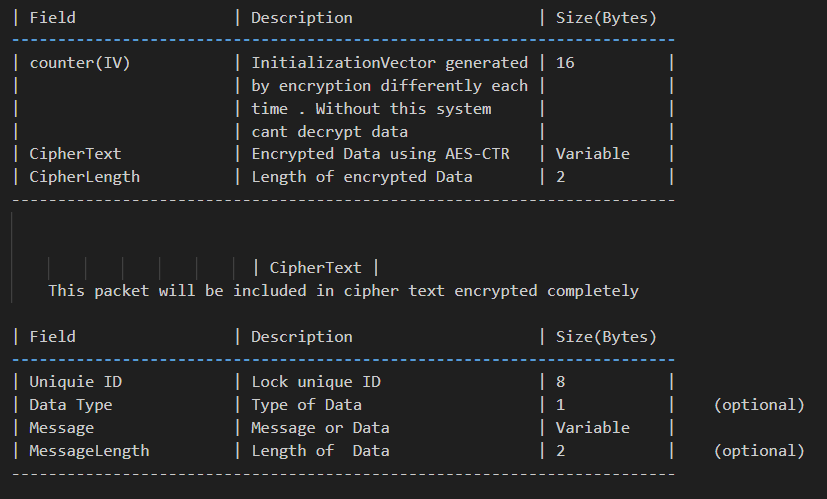
**Encryption Explained:**

**AES-CTR-128**

**1. What is AES-CTR?**  
AES (Advanced Encryption Standard) in CTR (Counter) mode is a secure method of encrypting data, often used in real-time communications.   
  
2. How it WorksAES-CTR encrypts the data by combining (or "XORing") it with a series of encrypted "blocks" that are generated using a **counter value**. The counter changes for each message, making the encryption unique every time.

**3. Counter Value:**   
Imagine a **counter** like a simple number, starting from 0.  
For **each message** sent, the counter value **increases by 1**.  
This counter is combined with a **secret key** and **encrypted** to generate a unique "block" (which looks like random data).  
The data (like commands for the smart lock) is then **XORed** with this block to get the encrypted message.  
  
**4. Why the Counter is important:**   
The counter ensures that even if two messages have the same data (like "unlock the door"), the encrypted version will look completely different. This way, an attacker cannot guess any patterns from the data, because each message uses a **unique counter** for its encryption.

**Communication Packet:**



**Message Packet Explanation:**

Message Structure for encrypted packet

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Description | Size(Bytes) | Example |
| <SOH> | Start of heading | 1 | Indicates start of encrypted message |
| Counter(IV) | Generated by encryption | 16 | Used to decrypte message |
| <EM> | Separator | 1 |  |
| CipherText | Encrypted Message | Length of Data will be given in “Cipher Length Field” | Containing otherinfo and some data |
| <EM> | separator | 1 |  |
| CipherLength | Length of encrypted data | 2 | Length of Cipher Text |
| <EOT> | End of Transmission | 1 | Signals that transmission is complete |

Message Structure for CipherText (when message is decrypted)

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Description | Size(Bytes) | Example |
| <STX> | Start of Transmission | 1 | Indicates start of message |
| Unique ID | Unique ID of the lock | 8 |  |
| <EM> | Separator | 1 |  |
| Data type | Type of message being sent | 1 | 0x01 for command, 0x02 for status and so on |
| <EM> | separator | 1 |  |
| Message ID | Specific Message within Data type | 1 | 0x01 for lock, 0x02 for unlock |
| <EM> | separator | 1 |  |
| Data | The data associated with the message (if applicable) | Length of Data will be given in “Message Length Field” | Some Data |
| <EM> | separator | 1 |  |
| Message Length | Length of data field | 2 | Length of Data |
| <ETX> | End of Transmission | 1 |  |
| <EOT> | End of Transmission (final indicator) | 1 |  |
|  |  |  |  |

Data types and message ID and Data explanation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | Description | Message ID | Data | Lock Response |
| (CMD)0x01 | Instruction to lock/unlock sent from the app to lock | 0x01 : Lock  0x02 : Unlock | NULL | ACK / NACK |
| (Status)0x02 | Request to Check the Lock state or battery status in percentage | 0x01 : Lock status  0x02 : Battery level | NULL | (Will be included in Data field)  0x01 : Locked  0x02: Unlocked  Battery % |
| (Config)0x03 | Settings Change, like time or encryption keys | 0x11 : Set time/date  0x21 : Set Encryption Key  0x12 : Get Time/date  0x22 : get encryption key | 00:12:23|22/12/24  (time/date)  PrivateKey | For setting  ACK/NACK  For case of  0x12: returns time in format 00:12:23|22/12/24  0x22: return saved private key |
| (ACK/NACK) 0x04 | Acknowledgement | ACK:0x06  NACK:0x15 | NULL | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Decimal | HEX | Symbol | Description |
| 0 | 0x00 | NUL | Null character |
| 1 | 0x01 | SOH | Start of Heading |
| 2 | 0x02 | STX | Start of Text |
| 3 | 0x03 | ETX | End of Text |
| 4 | 0x04 | EOT | End of Transmission |
| 25 | 0x19 | EM | End of medium |

**Message Packet Examples**

**Lock Command:**

* App To Lock

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x01><EM><0x01><EM><NUL><EM><0000><ETX><EOT>

 **Data Type 0x01**: Command

 **Message ID 0x01**: Lock

 **Data**: None, as unlocking doesn’t require additional data

 **Message Length**: 0000, since no data is included.

* Lock To app

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x04><EM><0x06><EM><NUL><EM><0000><ETX><EOT>

 **Data Type 0x04**: Ack/Nack

 **Message ID 0x06**: ACK

 **Data**: None, as unlocking doesn’t require additional data

 **Message Length**: 0000, since no data is included.

**UnLock Command:**

* App To Lock

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x01><EM><0x02><EM><NUL><EM><0000><ETX><EOT>

 **Data Type 0x01**: Command

 **Message ID 0x02**: UnLock

 **Data**: None, as unlocking doesn’t require additional data

 **Message Length**: 0000, since no data is included.

* Lock To app

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x04><EM><0x06><EM><NUL><EM><0000><ETX><EOT>

 **Data Type 0x04**: Ack/Nack

 **Message ID 0x06**: ACK

 **Data**: None, as unlocking doesn’t require additional data

 **Message Length**: 0000, since no data is included.

**Lock Status Command:**

* App To Lock

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x02><EM><0x01><EM><NUL><EM><0000><ETX><EOT>

 **Data Type 0x02**: Command

 **Message ID 0x01**: LockStatus

 **Data**: None, as unlocking doesn’t require additional data

 **Message Length**: 0000, since no data is included.

* Lock To app

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x02><EM><0x01><EM><0x01 or 0x02><EM><0001><ETX><EOT>

 **Data Type 0x02**: status command

 **Message ID 0x01**: Lock status

 **Data**: 0x01 or 0x02 depends on if lock is open or closed

 **Message Length**: 0001

**Battery percentage Status Command:**

* App To Lock

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x02><EM><0x02><EM><NUL><EM><0000><ETX><EOT>

 **Data Type 0x02**: Command

 **Message ID 0x02**: Battery Percentage status

 **Data**: None, as unlocking doesn’t require additional data

 **Message Length**: 0000, since no data is included.

* Lock To app

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x02><EM><0x02><EM><75><EM><0001><ETX><EOT>

 **Data Type 0x02**: status command

 **Message ID 0x02**: Battery Percentage status

 **Data**: 75 Battery level

 **Message Length**: 0001

**Set Time/Date Command:**

* App To Lock

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x03><EM><0x11><EM><00:12:23|22/12/24><EM><0017><ETX><EOT>

 **Data Type 0x03**:

 **Message ID 0x11**:

 **Data**: 00:12:23|22/12/24

 **Message Length**: 0017

* Lock To app

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x04><EM><0x06><EM><NUL><EM><0000><ETX><EOT>

 **Data Type 0x04**:

 **Message ID 0x06**:

 **Data**: no data

 **Message Length**: 0000

**Set Encryption Key Command:**

* App To Lock

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x03><EM><0x11><EM>

< 2b7e151628aed2a6abf7158809cf4f3c><EM><0016><ETX><EOT>

 **Data Type 0x03**:

 **Message ID 0x21**:

 **Data**: The Private Key will come here

 **Message Length**: 0016

* Lock To app

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x04><EM><0x06><EM><NUL><EM><0000><ETX><EOT>

 **Data Type 0x04**:

 **Message ID 0x06**:

 **Data**: no data

 **Message Length**: 0000

**Get Date/Time Command:**

* App To Lock

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x03><EM><0x12><EM>< NUL><EM><0000><ETX><EOT>

 **Data Type 0x03**:

 **Message ID 0x12**:

 **Data**: NUL

 **Message Length**: 0000

* Lock To app

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x03><EM><0x12><EM><00:12:23|22/12/24><EM><0017><ETX><EOT>

 **Data Type 0x03**:

 **Message ID 0x12**:

 **Data**: 00:12:23|22/12/24

 **Message Length**: 0017

**Get Encryption Key Command:**

* App To Lock

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x03><EM><0x22><EM>< NUL><EM><0000><ETX><EOT>

 **Data Type 0x03**:

 **Message ID 0x22**:

 **Data**: NUL

 **Message Length**: 0000

* Lock To app

<SOH><Counter (IV)><EM><CipherText><EM><CipherLength><EOT>

CipherText breakdown:

<STX><UniqueID><EM><0x03><EM><0x12><EM><2b7e151628aed2a6abf7158809cf4f3c><EM><0016><ETX><EOT>

 **Data Type 0x03**:

 **Message ID 0x22**:

 **Data**: 2b7e151628aed2a6abf7158809cf4f3c

 **Message Length**: 0016