CHAPTER – 7
Two-Tier Model for Securing Wireless IMDs

Contribution

In this chapter we propose a secure IWBAN architecture based on two-tier model by making use of a proxy device. We use request-response messaging between IMD and Proxy and publish-subscribe messaging between Proxy and external devices. We analyze the available security mechanisms and justify our choices for Tier-1 and Tier-2. Based on the analysis, for tier-1, we present two protocols, first for Proxy initiating communication with IMD and second for IMD initiating communication with Proxy. For tier-2, we present two protocols, first for ED as publisher and second for ED as subscriber. We use a mapping engine in the Proxy which translates requests received from IMDs into subscribe message and response received from IMD into publish message for the external devices. The mapping engine converts Publish message received from ED to a response and Subscribe message received from ED to a request for IMDs. Two-tier proxy based communication model provides confidentiality, integrity, authentication, access control and replay resilience of sensitive information while ensuring availability of information during regular and emergency wireless telemetry access. The use of publish-subscribe allows timely delivery of critical health related information, reduces traffic on IMD thus saving its battery, allows IMDs and EDs to be decoupled and receive the required information without needing to know each other. Our security model is agnostic to underlying networking services. Any IMD that allows bidirectional communication can use the protocol.

7.1 Introduction

IMDs require end-to-end security solution therefore we provide a model for security which works at the application layer. To deduce a suitable security model, we are rethinking the way we store, transmit, process and access the telemetry data from IMDs. This will help us to generalize the solution to provide security to a large range of IMDs. Our model uses a trusted external Proxy device to provide security. Proxy is a handheld device (like PDA or smartphone) acting as a mediator between external devices and different types of IMDs. This allows our defense system to shift security related transformations from IMD to the trusted Proxy device which in turn helps in reserving scarce resources of medical device exclusively for medical functions. The communication protocol is divided into two tiers to provide confidentiality, integrity, authentication, access control and replay resilience for IMD to IMD as well as IMD to External Device communication in an IWBAN. It provides availability of information during regular and emergency wireless telemetry access The first tier uses request-response model and the second tier uses asynchronous publish-subscribe [150] model. Due to selection of such communication model, the sender and receiver need not be synchronized and security mechanism can be selected based on the requirement and constraint of communicating parties.

7.2 Design Goals of Security Model

In this section, we present several criteria that represent desirable characteristics for a secure and lightweight communication system for IMDs which are mentioned below:

1. **Lightweight:** To match the low capabilities of the IMDs, it is important to minimize computation, communication, and storage overhead on the IMDs. Hence, cryptographic algorithms used with IMDs must satisfy these requirements to be resilient to DOS attacks.
2. **Access control:** Security framework should provide different privileges for different types of users. But, emergency situations require immediate medical action wherein access control must not pose a hurdle.
3. **Scalable:** The system should efficiently provide security even in a scenario where multiple IMDs are implanted and many EDs communicate with these IMDs.
4. **Flexible:** The security model should easily support addition or removal of external devices or IMDs.
5. **Minimize Invasiveness:** The security model should make minimal changes in the existing IMDs to increase acceptability by IMD manufacturers and patients**.**

6. **Support for Intrabody and Extracorporeal Communication:** The security model should be able to provide security for IMD-IMD communication as well as IMD-External Device communication.

7.3 Requirements of Two-Tier Security Model

1. All communication between External Device and IMD should pass through Proxy device.
2. The communication between the IMD and Proxy make use of the request response model, which must be supported by both IMD and Proxy.
3. The communication between the External Device and Proxy makes use of publish subscribe model, which must be supported by both External Device and Proxy.
4. The IMD is able to execute minimalist symmetric cryptographic operations for mutual authentication and authenticated encryption of messages and to store cryptographic key, counter and nonce for communicating with proxy.
5. The proxy is able to execute cryptographic operations and to store cryptographic keys for one or more IMDs and for one or more External Devices in tamper resistant manner.
6. The External Devices are able to execute symmetric and asymmetric cryptographic operations and to store cryptographic keys for communicating with proxy.
7. All IMDs constituting IWBAN of a single patient pair up with only one Proxy device

7.4 Assumptions

1. For developing a secure two-tier communication protocol for Implantable Medical Devices, we may assume typical IMD for our case.
2. The protocol assumes the presence of a no frills Transport Layer like UDP or any data transfer service like it to be available.
3. The external devices (ED) are operated by authorized medical staff.
4. One or more IMDs, Proxy Device and authorized external device follow the protocol as designed. IMD and the Proxy operate in Medical Implant Communication Service (MICS) band.
5. Proxy device is a patient's private device and only the patient or his doctor can have access to it.

6. The Proxy device is always present as it is an irreplaceable component of the security protocol.
7. The attacker does not try to physically harm the patient or remove the Proxy device.
8. We assume that the IMD and proxy have already been paired with a shared secret key after key establishment phase. This information can securely be installed when patient is in the hospital.
9. We recommend a different key for each IMD which is renewable, but our protocol imposes no such restriction.
10. We assume the Proxy has a list of legitimate programmers and their corresponding public keys. This information can securely be installed during device registration.

7.5 Overview of Proxy Based Two-Tier Security Model

The architecture of the proposed security system consists of three components: one or more IMDs, a Proxy device and one or more external devices all related to a single patient. The overall architecture is shown in FIGURE 7.1. It shows two-tier communication model for IMDs rendering secure wireless telemetry access. IMDs and Proxy device communicate in a secure manner by making use of request response protocol. Proxy device and External Devices communicate in a secure manner by making use of publish-subscribe protocol. The Proxy Device performs security related transformation on behalf of IMDs. A mapping engine in Proxy device is used to transform request-response messages to public-subscribe messages and vice-versa.

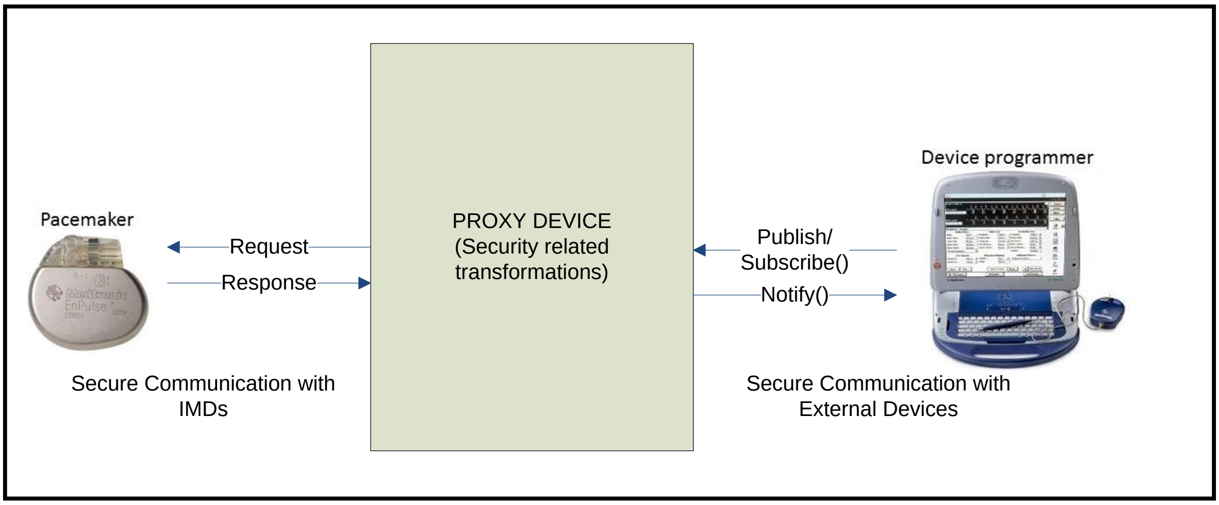


FIGURE 7.1 Overall view of two-tier architecture

Proxy device is responsible for converting the IMD request into a subscribe message and IMD response into a publish message. This allows interaction between the two communication models. Also do not require huge modifications in the IMD. We believe it is resource intensive for IMDs to support APIs provided by publish subscribe messaging middleware. This allows IMD to communicate in a light weight and secure manner. Publish-subscribe communication allows secure and seamless intercommunication of heterogeneous medical devices where one device can subscribe to data feed published by another device without need of knowing its identity.

7.6 Profiling of Security Mechanisms for Tier- 1: IMD and Proxy Device communication

We make use of request response model for communication between IMD and Proxy Device. In order to prevent adversaries from eavesdropping, injecting and tampering with data packets transmitted or received by IMDs, security services are necessary to be introduced into the communication protocol. Mutual authentication, data confidentiality and integrity, data origin authentication and replay protection are basic requirements which need implementation of cryptographic algorithms. Although cryptographic algorithms are already well established and are in use for wireless networks like WSN, the one to be used with IMDs needs to be chosen carefully due to its inevitable resource constraints. An IMD contains electronic circuits that perform data processing and control functions on an extremely small energy budget as explained in Chapter 1. On the other hand, using the security protocols always add additional overhead on the computational, storage and energy resources. Therefore, in order to design an energy efficient security model which suits the resource needs of these embedded devices it is critical to implement cryptography algorithms in a resource and computation efficient manner. The communication paradigm followed in IMDs is data-centric single-hop communication, instead of the route-centric multi-hop communication used in the conventional networks. Due to resource constraints and unique positioning, the use of conventional end-to-end security mechanisms like IPSec [134], TLS [135] SSL [136] in IMDs is obviated. Alternately, the necessary link layer security support may be provided by the underlying hardware based on IEEE 802.15.6 specification [137]. However, using a hardware based solution lacks flexibility and does not provide tailor made solutions to suit the need of recourse constrained IMD communication.

In the below text, we present the essential security services and the security mechanisms to be used in lieu and also the algorithm selected to meet its requirement in the proposed security model. We evaluate various aspects related to our model viz. selection of block ciphers, block cipher modes of operations, MAC sizes, IV generation, replay protection and mutual authentication scheme.

7.6.1 Security Service: Message Confidentiality

At the application layer, confidentiality can be achieved by use of encipherment techniques which is categorized as symmetric and asymmetric. Symmetric Cryptography is less resource intensive and therefore our choice for the communication protocol between IMD and Proxy. Symmetric ciphers are categorized as block cipher which processes plaintext in blocks or stream ciphers which processes plaintext as stream of data by making use of Pseudo Random Key Stream Generator. Available light weight block ciphers are AES, PRESENT, MISTY, XXTEA, BLOWFISH, IDEA and RC6 [128]. Table 7.1 lists some of the lightweight cipher and their parameters viz. the key-size, block-size and the number of rounds, security margin and program size in software. For choosing an appropriate block cipher we went through the published research work available in literature. Authors in [110] evaluates block and stream ciphers for their memory requirement and execution time. Authors in [111] analyses DES, AES and RC5 for energy expense during encryption, hashing and wireless transmission. In [112] author profiles block and stream ciphers for computational requirements and [113] profiles lightweight versions of block cipher for performance, power and memory requirements. In [3] author presents a comparative analysis of symmetric block ciphers for light weight encryption in implants.

Table 7.1 Benchmark suite of symmetric ciphers

Encryption Algorithm	Block Size (bits)	Key Size (bits)	Rounds (#)	Security Margin
3WAY [128]	96	96	11	2002
BLOWFISH [128]	128	128	16	2076
DES [128]	64	56	16	1982
GOST [128]	64	256	34	2243
IDEA [128]	64	128	8.5	2076
LOKI91 [129]	64	64	16	1992
RC5 [128]	64	128	12	2076
SKIPJACK [129]	64	80	32	2013
XXTEA [115]	64	128	32	2076
MISTY1 [114]	64	128	8	2076
RC6 [114]	18	128	20	2076
TWOFISH [114]	128	128	16	2076
RIJNDAEL [114]	128	128	12	2076

128-bits key size is essential for a cipher requiring security margin of 2076. We found the AES cipher Rijndael [132] was among the top five ciphers in all the performance metrics. For critical devices like IMDs we need to choose a stable algorithm which has been cryptanalyzed well. Therefore we use 128-bit key AES cipher Rijndael [132]. In fact, majority of research work [81] [68] [139] for securing implants have opted for the same. A message with multiple blocks can be encrypted in Electronic Codebook Mode (ECB) but as this method is vulnerable to cryptanalysis, therefore block cipher modes are used. Block modes of operation is the way of encrypting a message with a longer block size using algorithms like Cipher-Block Chaining (CBC), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB) and Counter Mode (CTR). The choice of block cipher mode plays a significant role in determining the efficiency of secure communication protocols.

7.6.2 Security Service: Message Integrity and Authentication

A message authentication code (MAC), needs to be selected for assurance of message integrity and authentication. In security model of [69] encryption is done in ECB mode and then MAC is calculated using CMAC. Block cipher is used to encrypt n blocks first and then MAC algorithm is invoked for n blocks requiring a total of 2*n invocations. This overhead can be avoided by using Authenticated-encryption (AE) modes [92] which allow use of a single key to provide confidentiality and authenticity with significantly lower computational cost as compared to sequential encryption and authentication. Moreover it does not require use of the conventional block cipher modes. The popular AE modes are Offset Codebook mode (OCB) [93], Counter with Cipher Block Chaining (CCM) [94], Carter-Wegman + CTR mode (CWC) [121] and Galois Counter Mode (GCM) [95]. Out of these OCB [9] is covered with intellectual property rights. CCM [118] cannot be pipelined or parallelized. In CWC [121] message authentication is performed by 127-bit integer multiplication operation which increases the implementation cost as per [130]. GCM [119] is a two pass combined mode for authenticated encryption which not protected by intellectual property claims and gives high speed of processing. In [130] GCM mode is evaluated for implementation in link layer of wireless sensor network, their findings state that GCM mode can be selected for resource constrained applications that require message confidentiality as well as authentication. On comparing CPU usage cycle, throughput, and energy usage parameters with CBC-Skipjack and CBC-AES, it is shown that AES-GCM incurs overhead of 12% increase in energy and 28% increase in RAM usage, but at the same time offers encryption as well as authentication [130]. Therefore, we make use of AES-GCM for authenticated encryption as explained below.

7.6.2.1 Authenticated Encryption Mode- GCM

Galois/Counter Mode (GCM) is a block cipher mode of operation which uses universal hashing operation over a binary Galois field for authenticated encryption [126]. It was proposed and recommended by NIST in 2007 [127]. As per [126], it is patent free, has high performance, and can be implemented in hardware as well as software; software implementations can make use of table driven field operations to achieve high efficiency. According to [15] GCM can act as a stand-alone MAC when encryption is not required, moreover it can act as an incremental MAC such that with computational cost is proportional to the number of bits that change. The structure of GCM mode is shown in Fig. 7.2.

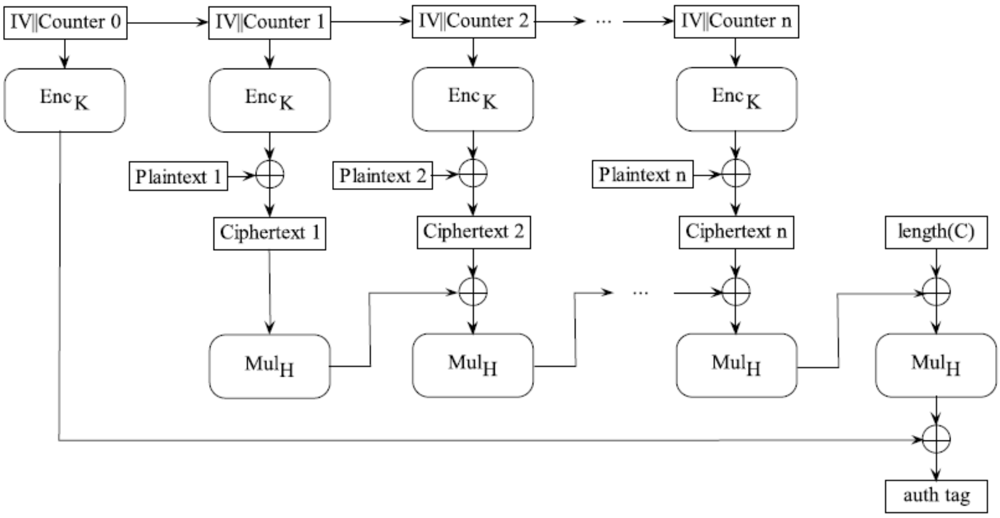


FIGURE 7.2 Structure of GCM [141]

GBM takes four inputs: a secret key, an initialization vector (IV), a plaintext, and additional authenticated data (AAD) which is authenticated but not encrypted. It produces two outputs, a cipher text of the same length as plaintext and an authentication tag as shown in FIGURE 7.2. The bit length of plaintext or cipher text is integer multiple of 128 bits [127]. The IV and Key are used to generate a key stream which is XORed with plaintext to get the cipher text. The Key stream must always have different values to avoid cryptanalytic attacks as XORing two ciphertexts will result into a value which is XOR of two plaintexts as shown below:

$P1 \oplus \text{Encryption Key} = C1$

$P2 \oplus \text{Encryption Key} = C2$

$C1 \oplus C2 = P1 \oplus P2.$

Therefore Encryption Key value should be different for each plaintext. The ciphertext generated for subsequent blocks is used to generate the authentication tag. The block diagram of AES-GCM is shown in Figure 7.3.

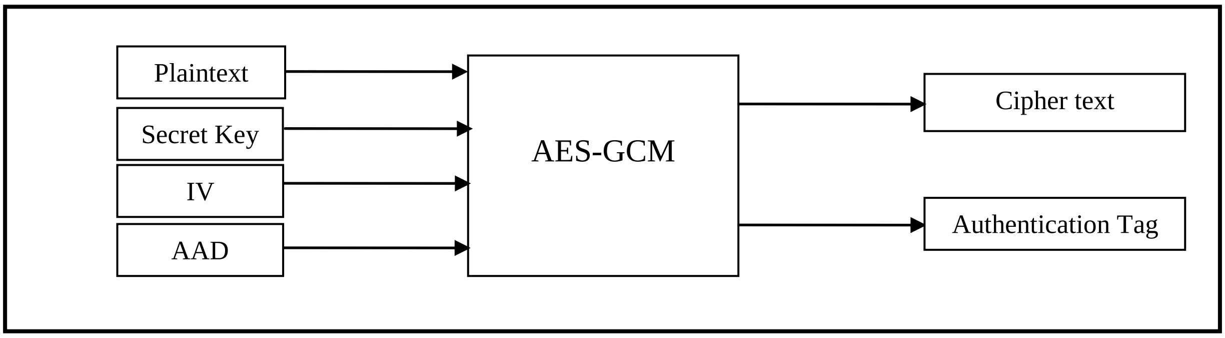


FIGURE 7.3 Block Diagram of AES-GCM

Description of the inputs required by AES-GCM and outputs generated are given below:

Plaintext

For an IMD, sensed physiometric data, patient related data, request for data from other IMDs and commands from external devices constitutes plaintext. For Proxy, it is a request for data or command that another authorized IMD or external device has issued. Plaintext is converted into blocks of 128-bits before encryption.

Secret Key

The IMD and Proxy shares pair wise secret key. According to [131], key size of 128-bits is a sufficiently long to defend against brute force attacks by a powerful adversary. Therefore 128-bit key is used.

AAD

Data like timestamp, device ID requires data integrity and data origin authentication but do not require confidentiality. Such data fields can be included in Additional Authenticated Data (AAD) field.

Initial Vector (IV)

Initial Vector (IV) is an essential component of block cipher mode. For AES-GCM, IV need not be a secret but must be unique for subsequent blocks of data. Instead of sending the full length IV, in Minisec [133] a link layer security protocol of WSN, only few bits of IV are send to reduce transmission overhead. Instead of sending IV, we generate IV by use of the existing data header as explained in the next section.

Ciphertext

Ciphertext is of the same block size as that of plaintext. In order to decrypt the ciphertext generated by AES-GCM only encryption box needs to be installed as AES-GCM doesn't make use of a decryption box.

Authentication Tag

Size of the Message Authentication Code (MAC) employed must be chosen depending on the packet transmission rate of the device under consideration. According to [131], the appropriate MAC size is 8 bytes for embedded devices with transmission rate of 250 kbps in 2-3 meters. The Authentication Tag generated by AES-GCM by default is 12 bytes, which can be truncated to 8 bytes before sending to the peer device.

7.6.2.2 Initial Vector Format for Tier-1

As discussed above, we reduce communication overhead due to transmission of IV by deriving it from the data header of the application layer data. We generate IV by making use of nonce value and counter value which changes for every subsequent data block thus generating a unique IV. As IV need not be a secret it is generated by combining two 32-bit Nonces, one generated by IMD and one by Proxy and a 32-bit counter received from the communicating device to make a 96-bit IV as shown in Fig. 7 .4.

NonceIMD(32 bits)	NonceProxy(32bits)	CounterProxy(32 bits)

FIGURE 7.4 (a) Structure of IV for IMD

Nonce Proxy(32bits)	Nonce IMD(32 bits)	Counter IMD(32 bits)

FIGURE 7.4 (b) Structure of IV for Proxy.

7.6.3 Security Service: Replay Protection

Replay attack is performed by capturing packets of one communication session and replaying such packets later either to gain unauthorized access or to impersonate messages.

One of the areas where most of the security models proposed in literature fail is in provisioning replay resilience. In order to find an application layer replay scheme which can be used between IMD and Proxy, we studied the replay protection schemes given in [143] [144]. In [140] protection schemes are analyzed and categorized as synchronized counter based, nonce based, and bloom filter based.

7.6.3.1 Counters

For our scheme, we refer to [145] [143] and select counter based algorithm as it can be easily incorporated without much overhead. Use of a monotonously increasing counter guarantees semantic security. If a sender sends the same message, the resulting cipher text is different as different counter value and IV value are used. Also, once a receiver observes the counter value, it can rejects packets with an equal or smaller counter value. Therefore, an attacker cannot replay old packets without receiver detecting it. We make use of a 32-bit counter value which allows 2^{32} -1 counter values for a session. In [131] counter is used to drop stale packets. The use of such counter in our security model is twofold, one it is used for replay resilience and two it is used implicitly to construct the IV.

The algorithm [131] is modified for checking counter value at Proxy side and at IMD side as shown below:

Algorithm – 1: Executed by IMD to check the counter value of received message to detect replay of an older message.

```
CounterReplayDetect(CounterReceived,$C_{Proxy}$)
{
     if (CounterReceived <=$C_{Proxy}$)
          replayed = 1;
    else
    {
          replayed = 0;
          $C_{Proxy}$=CounterReceived;
    }
}
```

Algorithm – 2: Executed by Proxy to check the counter value for received message to detect replay of an older message.

```
CounterReplayDetect (Counter Received, $IMD_{ID}$)
{
    id = 0;
```

```
        for id = 1 to lastValidIMDId {
        if (id==$IMD_{ID}$) {
                 if (CounterReceived <=LastCount[$IMD_{ID}$])
                        replayed = 1;
                Else
                {
                        replayed = 0;
                         LastCount[$IMD_{ID}$]=CounterReceived;
                }}}
}
```

7.6.3.2 Nonce

Nonce are random numbers are numbers that a sender associates with a message and receiver repeats in the response message to uniquely associate each message to its reply and ensures message freshness. Nonces are used only during first two exchanges for mutual authentication. They are generated on the fly during protocol execution and only the values associated with the current session are kept in memory, thus requiring minimal memory overhead. Nonce is also used in construction of IV at IMD and Proxy side.

1. **Nonce Generation for IMD**

We make use of physiological value (PV) derived from the human body to generate nonce for the IMDs which in turn is used to secure the data communications against replay. The advantage is we do not require a Pseudo Random Number Generator at IMD side now. The level of randomness of biometric is determined by the amount of its entropy [146]. The required randomness can be obtained by simultaneous use of multiple biometrics or deriving a sequence from multiple instances of measurements. Thus random number generation overhead can be reduced by using a time-varying biometric, known as physiological value (PV). ECG (electrocardiogram) produced by cardiac IMDs such as ICDs and pacemakers are one of the popular and usable PV. Suitably processed ECG samples effectively constitute a low- bandwidth stream of random bits well suited for generation of random numbers. We use this entropy measure to generate Nonce at IMD side.

Algorithm – 3: Executed by IMD to generate random value Nonce.

```
Generate_Rand_IMD (Sensed_PV)
{
        bit_counter=1;
```

```
    for bit_counter=1 to n{
        extract_random_bits (Sensed_PV)
    }
    Combine random bits to generate 32 bit Nonce;
}
```

2. **Nonce Generation for Proxy**

In case of proxy device the pseudo-random number generator is used as it is a resource rich device. As specified in the [147], we use the block cipher AES in counter mode, called CTR. The algorithm is described here:

Algorithm – 4: Executed by Proxy to generate random value Nonce.

Block cipher-CTR mode (compliant with NIST 800-38A)

```
Generate_Rand_Proxy ( )
{
    Oj = AES-CTR(k, Ctrj )
    NProxy = |Oj |0···31
    Ctrj +1 = |Oj |32···64
}
```

7.6.4 Security Service: Mutual Authentication For mutual authentication, ISO/IEC 9798 Part 2 [148] specifies six schemes based on symmetric encryption algorithms [ISO 1999], which provides different degrees of authentication: unilateral authentication, mutual authentication, and authentication with key establishment using a third entity (server). Our proposed scheme is based on the fourth protocol of this standard, as we require mutual authentication between the Proxy and the IMD.

7.6.5 Security Service: Access Control

Although Access Control is implemented in the second tier, it is worth a mention here. IMD devices being resource constrained are incapable of handling Access Control. They trust the Proxy device completely which handles the access control in behalf of IMDs.

Table 7.2 Summary of components adopted in communication protocol for Tier 1: Proxy-IMD communication

Security Service	**Adopted Security Mechanism**
Key Management	Initial secret key distribution (during installation of IMD) Offline distribution and Offline Key Replacement
Message Authentication	AES-GCM
Message Integrity	AES-GCM
Freshness	Nonce and Counter
Confidentiality	Symmetric Encryption using AES
Mutual Authentication	Fourth protocol of ISO/IEC 9798 Part 2

7.7 Profiling of Security Mechanisms for Tier - 2: Proxy Device and External Device communication

We make use of topic based publish-subscribe model [150] for communication between IMD and Proxy Device. Essential security services are Confidentiality, Integrity, Authentication, Access Control and Replay protection [151].

7.7.1 Components of the Communication Model

1. **Publishers:** Publishers are either EDs that are capable of generating data for a specific topic or IMDs that are capable of sending a response when data related to the topic is requested by the proxy.
2. **Subscribers:** Subscribers are either EDs that express interest in a specific topic data or IMDs that have requested for a data from proxy which is related to a specific topic.
3. **Proxy:** Topics are registered with the proxy to whom publishers can send topic data after authentication and validation of its role for a topic. Subscribers can request for topic data after authentication and validation of its role for a topic. Proxy matches the two parties and store and forward topic data to subscribers. List of topics depend on the type of IMD, and may get modified for e.g. when an IMD is added or removed. Security services for mutual authentication, message confidentiality and authentication, replay resilience and access control needs to be provided.

7.7.2 Design Choices for Proxy and ED communication

Point-to-point security solutions like TLS/SSL or IPSec or Kerberos cannot be used here therefore we design a scheme for ensuring integrity and confidentiality of data. Unlike point-to-point communication which secures a stream of information, individual messages for Topics need to be secured. We make use of symmetric 128-bits Topic wise master key. Using shared secret key for publishers and subscribers would mean a group of parties sharing the keys which will increase vulnerability if even one of the parties is compromised. Therefore Topic master key is unique for every Topic. To provide forward and backward security, Topic master key is renewed whenever a subscriber leaves or joins the proxy. The advantage is that if the key for a specific Topic is compromised, still messages related to other Topics cannot be decrypted. For every publisher the topic encryption key is uniquely derived from the Topic master key by making use of Publisher Specific Value (PSV). The advantage is that one publisher cannot decrypt or modify the data send by another publisher on the same topic. For exchanging the symmetric keys and for mutual authentication between Proxy and ED we make use of Public Key cryptography.

7.7.3 Public Key Cryptography

A method for entity authentication and exchange of secret key is to use public key cryptography. Such algorithm uses a set of related keys known as Public and Private Keys. Public keys are exchanged and private keys are kept secret. Exchanged public key can be used for encryption and authentication. The keys used are large in size therefore instead of using them for data encryption; they are used to share secret keys which can then be used for Symmetric Encryption. The Algorithms can also be used to create digital signatures for entity authentication. As both Proxy Device and External Device are rechargeable and have the required computational power, therefore for mutual authentication and secret key exchange we propose use of public key cryptography. Algorithms for Public Key Cryptography are RSA [154] and Elliptic curve cryptography (ECC) [149]. ECC is being used by National Security Agency (NSA) for signature generation and key exchange [150] is preferred.

7.7.4 Security Service: Massage Confidentiality, Integrity and Authentication

Once secret key is shared between Proxy device and External Device using Public Key, Proxy and ED make use of AES-GCM for Message Confidentiality, Integrity and Authentication.

7.7.5 Security Service: Replay protection

Nonce are used for mutual authentication between Proxy and ED. Timestamps are used to ensure message freshness as there can be more that one device publishing to the same topic therefore use of counter will not be relevant.

7.7.6 Security Service: Access Control

An access control list is generated and stored in the Proxy device for granting access. It defines for which are the valid topics, for which topic which device can assume the role of a Publisher and which device can assume the role of a Subscriber.

7.7.7 Security Service: Mutual Authentication

By use of Digital Signatures on a random value nonce, Proxy and External device authenticates each other. Digital Signatures generation algorithm by use of ECC is proposed in [512].

Table 7.3 Summary of components adopted in communication protocol for Tier 2: Proxy-ED communication

Security Services	Devices	Security Mechanism
Source Authentication	Publisher Device, Subscriber Device and Proxy	Digital Signature
Topic Data Confidentiality	Publisher encrypts and Subscriber Decrypts	AES-GCM
Topic Data Integrity and authentication	Ensures Topic Data cannot be modified and source of origin can be proved.	AES-GCM
Anti-Replay	Ensures Topic Data cannot be replayed	Timestamp
Access Control	Ensures only authorized Publishers and Subscribers can communicate.	Access Control List maintained by proxy
Key Management	Publisher Device, Subscriber Device and Proxy	Proxy stores Public Key of devices during registration. Uses Public Key for sharing secret key.

7.8 The Two-Tier Proxy based Architecture and its Components

The secure dissemination of telemetry data between IMDs and external devices occurs with the proxy device as a mediator a shown in Fig. 7.5. IMD performs lightweight authentication and symmetric encryption and generation of random nonces. Rather than storing secret keys of various external devices, it only stores the secret key and counter for proxy device thus requiring less storage. IMD includes battery, sensing or actuation unit, storage unit, processing unit and a wireless transceiver. The dashed line for skin shows that the IMDs are subcutaneous. For communication with IMDs, the proxy device supports secure request response protocol.

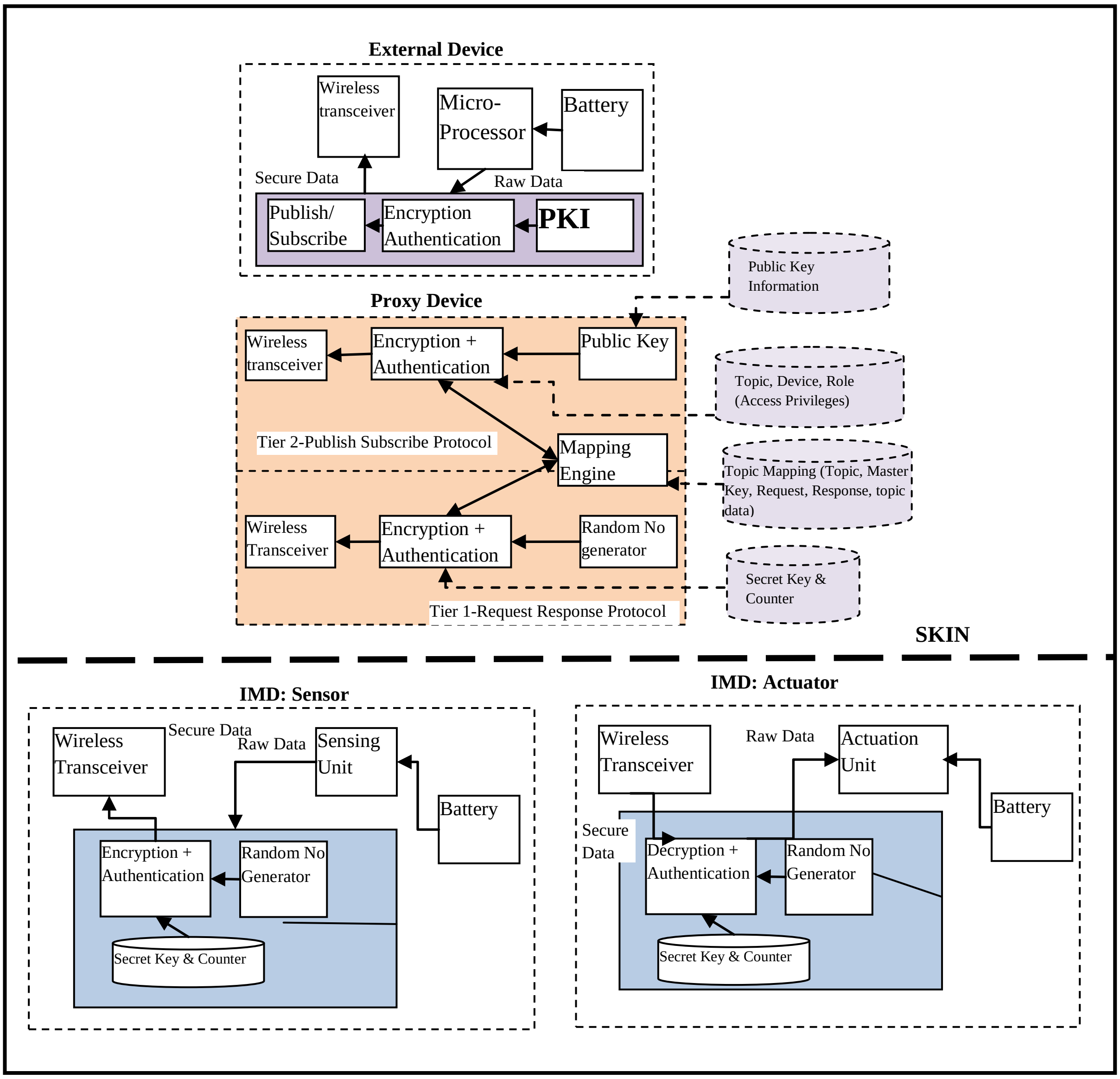


FIGURE 7.5 Architecture of Proxy based Two Tier solutions

Proxy device includes a wireless transceiver, supports lightweight authentication and symmetric encryption, and generation of random nonce. It stores the secret keys of all IMDs with which it is paired. It includes a mapping engine which transforms request-response messages to publish-subscribe messages. It stores Topic Mapping data like Topic Name, Master Key, Request format, Response format which is required by the mapping engine to perform the required conversion.

For secure communication with external devices, proxy device includes wireless transceiver, supports authentication and encryption with the use of Public key stored in

Public Key Information Database. Access Privileges are used by proxy for providing access control by defining specific roles for registered external devices for each topic. It also stores a list of topics, devices, their roles and validity period. It also stores the public key of the external devices which is used for mutual authentication and topic key exchange by use of public keys. For every topic, a list of authorized publisher and subscriber is maintained by the Proxy Device. The external devices are authenticated by the Proxy Device using mutual authentication protocol and for every topic their role of publisher or subscriber is verified. For example, if for a topic Blood Glucose, IMD is publisher and external device A is subscriber then only external device A will receive the notification for topic data and required topic key to decrypt the data. Even though adversary B can snoop over the wireless channel but it will not be able to decrypt the event. The topic data are encrypted using topic keys before disseminating the events in the wireless communication network.

The topic-driven communication is very advantageous for timely and real time information dissemination, for reducing traffic below the level typically required by resource-constrained wireless communication system of IMDs. Our security model functions with both multiple subscribers and multiple publishers which may have different roles for different telemetry event. Publish-subscribe entities operate asynchronously and are unaware of each other's existence.

The first tier constitutes the request-response communication between IMD and Proxy which has two protocols, one for Proxy initiating communication and second for IMD initiating communication. The second tier constitutes publish-subscribe communication with proxy as the mediator which supports two combinations; one with External Device as publisher and second with External Device as subscriber.

7.9 Proxy Device and its role in the two-tier Security Model

This section describes the proxy device and its role in the entire communication protocol. Fig. 7.7 is a flow diagram showing the working of Proxy device in a wireless communication network. One or more IMDs and one or more external medical devices are registered with the proxy device. At step (202), the Proxy device receives a communication request. At step (204), it checks the Device Type (IMD or External Device). At step (206) request-response mode is opted if the communicating device is an IMD. At step (208), light weight symmetric key based mutual authentication is performed for IMD by use of secret key from (210). At (212) if the device is not found authentic, session is terminated at (214). Otherwise, encrypted request from IMD is received at (216). For the request, topic name is retrieved and role of the IMD as subscriber is verified at (218) from (226). At (220) if the device is not found authentic, session is terminated at (214).

The publisher device type for the topic is retrieved at (224) from (226). At (228), if the publisher device is an IMD, request is send to the IMD for data at (230). At (232) light weight symmetric key based mutual authentication is performed for IMD by use of secret key from (210). At step (234), if device authentication fails, session is terminated at (214). Otherwise at step (236), encrypted request is send to IMD and encrypted response is received at (238). At (240), data is decrypted; its integrity and freshness is checked. At step (242), if data is not found authentic it is discarded at (244) and information is logged. At step (246), data is encrypted and stored as published topic data in (226). At step (248), published topic data is send to all registered and available subscribers.

When the Publisher device is an external device, at step (250) the published topic data is converted to a response. At step (252) the data is encrypted and send to the IMD.

When the device sending connection request is an external device, publish-subscribe mode is used at (254). At step (256) public key based mutual authentication is carried out using public key from (258). At (260) if the device is not authentic, session is terminated at (262). Otherwise, at (264) topic name and device role is checked from (266). At step (268) if the topic and device role is valid, at (270) the role is checked for publisher or subscriber. If the device is subscriber for the topic, Topic Master Key is send to the device by encrypting it with subscriber device Public Key at (272). At step (274), request is send to corresponding IMD for data after retrieving IMD device name from (226). At step (276), encrypted response is received from IMD. At step (278), received data is decrypted and converted to topic data.

At (280) topic data is encrypted by a derived topic key. Topic master key generated by proxy, shared with registered subscriber of topic. This key is renewed when a subscriber joins or leaves the proxy. This provides forward and backward security. Topic master key is hashed to generate derived topic key using a Publisher Specific Value (PSV) specific to the publisher device. It is shared with authorized the publishers of topic by use of Public Key encryption. It is renewed when a subscriber joins or leaves the proxy. This provides forward and backward security. Publisher Specific Value (PSV) generated by Proxy for each Publisher Device for each topic. At (282) the encrypted data is notified to all subscribers along with the specific value. At step (284), if the device is a publisher, derived topic key for that publisher is encrypted with Public key of the publisher and send. At (286) encrypted topic data is received from the publisher and stored at (226). At (282) the encrypted data is notified to all subscribers along with the Publisher Specific Value (PSV).

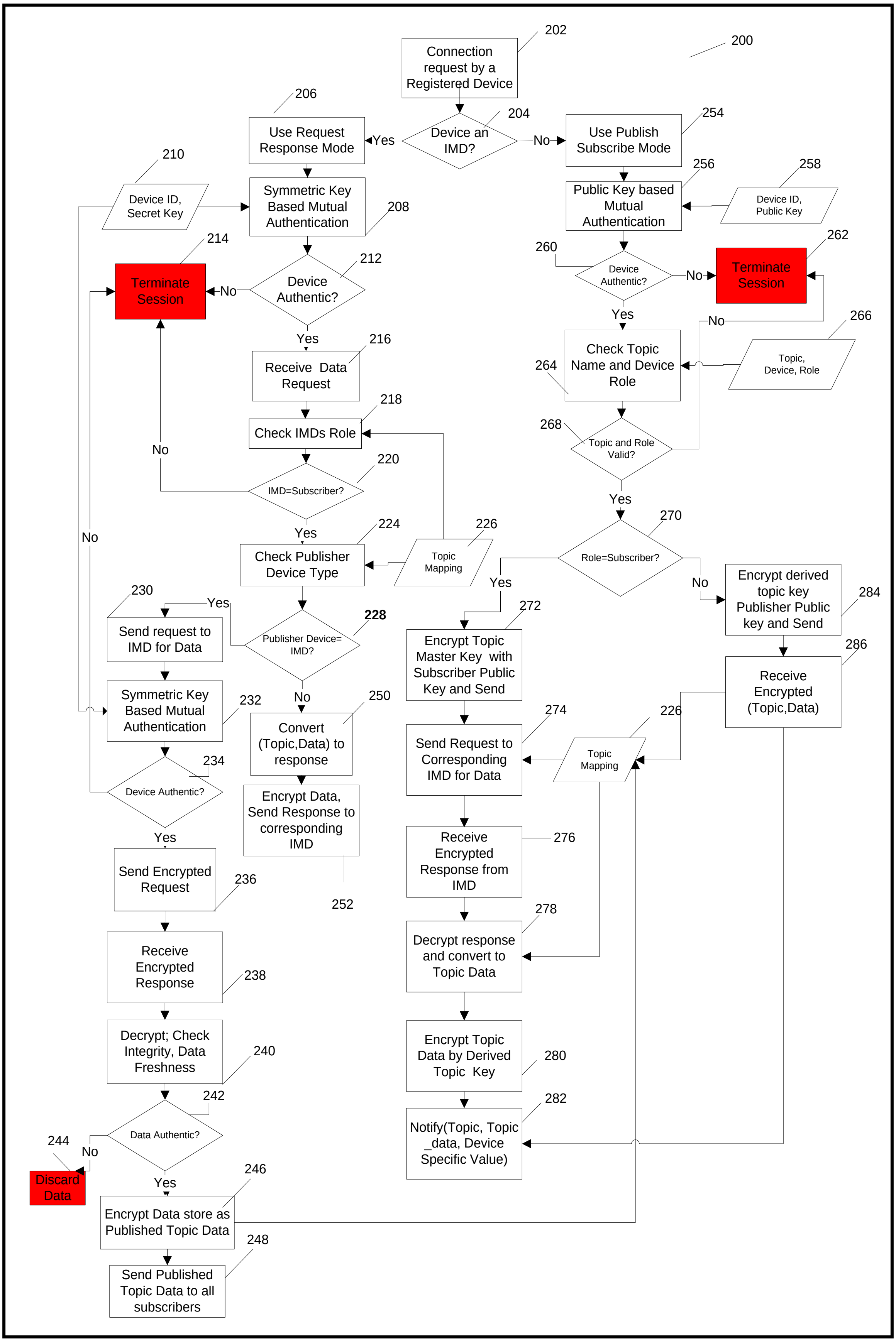

200
202
Connection request by a Registered Device
204
Device an IMD?
Yes
No
206
Use Request Response Mode
254
Use Publish Subscribe Mode
210
Device ID, Secret Key
Symmetric Key Based Mutual Authentication
208
256
Public Key based Mutual Authentication
258
Device ID, Public Key
214
Terminate Session
No
212
Device Authentic?
260
Device Authentic?
No
262
Terminate Session
Yes
216
Receive Data Request
Yes
No
266
264
Check Topic Name and Device Role
Topic, Device, Role
218
Check IMDs Role
268
Topic and Role Valid?
No
220
IMD=Subscriber?
Yes
No
Yes
224
226
270
Check Publisher Device Type
Topic Mapping
Role=Subscriber?
230
Yes
228
Yes
No
272
Send request to IMD for Data
Publisher Device= IMD?
Encrypt Topic Master Key with Subscriber Public Key and Send
Encrypt derived topic key Publisher Public key and Send
284
286
Symmetric Key Based Mutual Authentication
232
No
250
Convert (Topic,Data) to response
274
226
Receive Encrypted (Topic,Data)
234
Device Authentic?
Encrypt Data, Send Response to corresponding IMD
Send Request to Corresponding IMD for Data
Topic Mapping
Yes
276
Receive Encrypted Response from IMD
Send Encrypted Request
236
252
278
Receive Encrypted Response
238
Decrypt response and convert to Topic Data
Decrypt; Check Integrity, Data Freshness
240
Encrypt Topic Data by Derived Topic Key
280
242
282
244
No
Data Authentic?
Notify(Topic, Topic _data, Device Specific Value)
Discard Data
Yes
246
Encrypt Data store as Published Topic Data
248
Send Published Topic Data to all subscribers

FIGURE 7.6 Work Flow Diagram of Proxy Device

7.10 Description of proposed protocol for Tier – 1: IMD and Proxy Communication

The communication protocols between IMD and Proxy is described in this section. Before commencement of the protocol, IMDs are registered with proxy during registration phase and shares secret keys with it.

The notations used by us for describing the protocols are shown in Table 7.4.

Table 7.4 Notations used in Tier 1: Proxy- IMD communication

Notation	Size	Meaning
IMD	-	Medical Device implanted inside human body
Proxy	-	External Device that provides secure communication with one or more IMDs for a patient.
ID_{Proxy}	32-bits	Identifier of Proxy
ID_{IMD}	32-bits	Identifier of IMD
N_{IMD}	32-bits	Nonce generated by IMD
N_{Proxy}	32-bits	Nonce generated by Proxy
$K_{IMD,Proxy}$	128-bits	AES-GCM encryption Key for secure communication from IMD to Proxy. Stored by IMD and Proxy.
$K_{Proxy, IMD}$	128-bits	AES-GCM encryption Key for secure communication from Proxy to IMD. Stored by IMD and Proxy
C_{IMD}	32 bits	Counter of IMD to avoid replay attack
C_{Proxy}	32 bits	Counter of Proxy to avoid replay attack
IV_{IMD}	96 bits	IV used by IMD for AES-GCM encryption. $IV_{IMD}= N_{IMD}\|\| N_{Proxy}\|\| C_{Proxy}$
IV_{Proxy}	96 bits	IV used by Proxy for AES- GCM encryption. $IV_{IMD}= N_{Proxy}\|\| N_{IMD}\|\| C_{IMD}$
REQ	32 bits	Request in plaintext
RESP	N*32 bits	Response of Request in plaintext; multiples of 32 bits

(C,Tag)	-	$(C,Tag)=GCM_K(IV,P,AAD)$
		K=Encryption Key ($K_{Proxy, IMD\ or}\ K_{IMD,Proxy}$)
		IV=Initialization Vector
		P=Plaintext message to be encrypted and authenticated
		AAD=Additional Authenticated Data; This data is authenticated, but not encrypted
		C=Ciphertext
		Tag= Authentication Tag

Tag	64 bits	Authentication Tag

The communication between IMD and Proxy can occur in two scenarios. The proposed protocol for, Proxy initiating communication and IMD initiating communication are given below:

7.10.1 Protocol 1: Proxy initiating communication

When an external device or another IMD requires telemetry data they subscribe to the corresponding Topic with the Proxy. For every Topic and related request-response sequences needed are preconfigured in Proxy. The proxy wakes up the respective IMD associated with that topic as a publisher.

In this scenario, Proxy initiates communication with the IMD. Before communicating, proxy is authenticated by IMD using a light weight challenge handshake protocol as shown in the sequence diagram in Figure 7.7.

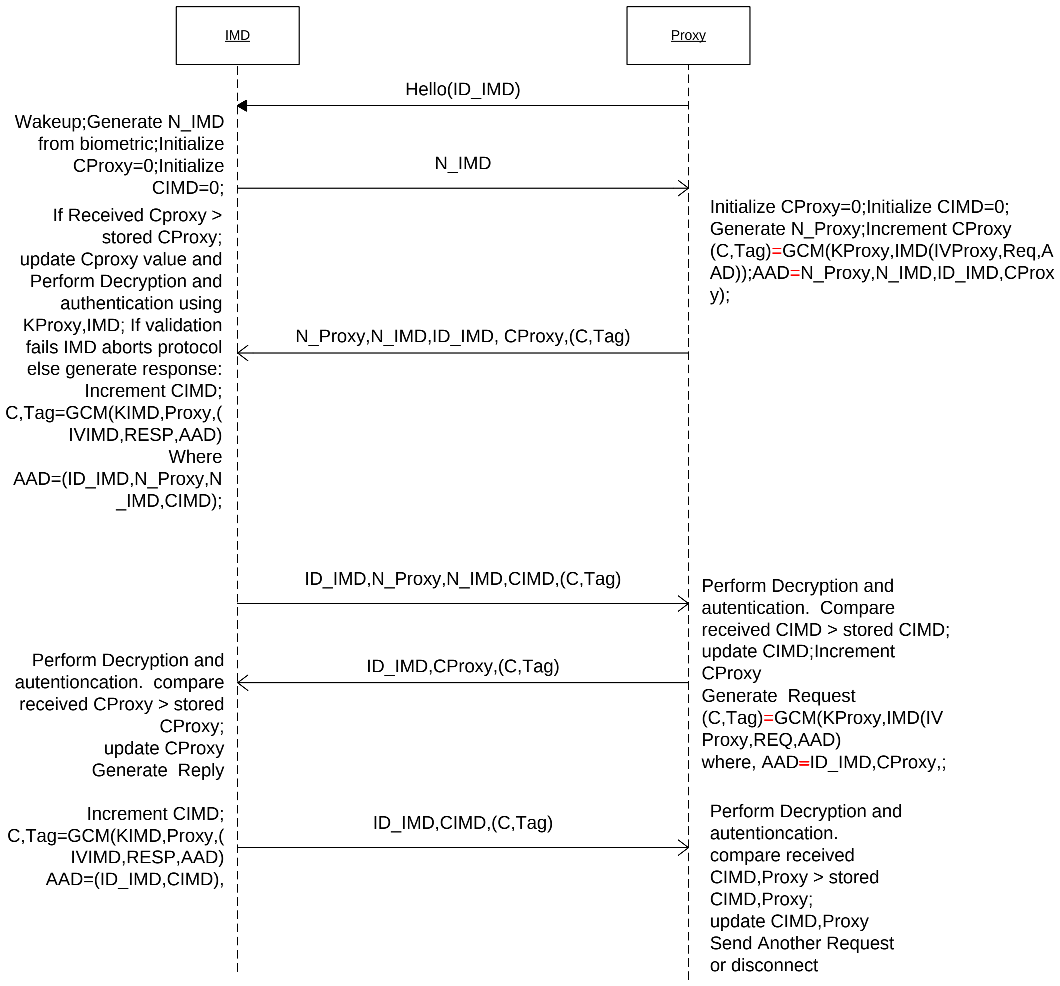

IMD
Proxy
Hello(ID_IMD)
Wakeup;Generate N_IMD from biometric;Initialize CProxy=0;Initialize CIMD=0;
N_IMD
Initialize CProxy=0;Initialize CIMD=0; Generate N_Proxy;Increment CProxy (C,Tag)=GCM(KProxy,IMD(IVProxy,Req,AAD));AAD=N_Proxy,N_IMD,ID_IMD,CProxy);
If Received Cproxy > stored CProxy; update Cproxy value and Perform Decryption and authentication using KProxy,IMD; If validation fails IMD aborts protocol else generate response: Increment CIMD; C,Tag=GCM(KIMD,Proxy,(IVIMD,RESP,AAD) Where AAD=(ID_IMD,N_Proxy,N_IMD,CIMD);
N_Proxy,N_IMD,ID_IMD, CProxy,(C,Tag)
ID_IMD,N_Proxy,N_IMD,CIMD,(C,Tag)
Perform Decryption and autentication. Compare received CIMD > stored CIMD; update CIMD;Increment CProxy Generate Request (C,Tag)=GCM(KProxy,IMD(IVProxy,REQ,AAD) where, AAD=ID_IMD,CProxy,;
Perform Decryption and autentioncation. compare received CProxy > stored CProxy; update CProxy Generate Reply
ID_IMD,CProxy,(C,Tag)
Increment CIMD; C,Tag=GCM(KIMD,Proxy,(IVIMD,RESP,AAD) AAD=(ID_IMD,CIMD),
ID_IMD,CIMD,(C,Tag)
Perform Decryption and autentioncation. compare received CIMD,Proxy > stored CIMD,Proxy; update CIMD,Proxy Send Another Request or disconnect

Figure 7.7: Sequence Diagram for Protocol: Proxy Initiating Communication

The message exchanges related to this protocol and their interpretations are given in Table 7.5. The table shows the sender and the receiver of the message, the message transmitted and also what action is taken by the receiver after receiving the message.

Table 7.5 Description of messages for Protocol: Proxy initiating communication

Sender→Receiver	**Message**	**Interpretation**
Proxy→ IMD	hello(ID_{IMD})	The proxy sends Hello to wake up the implant.
IMD → Proxy	N_{IMD}	The IMD generates a nonce by use of Physiological Value and sends it to the proxy
Proxy→ IMD	N_{Proxy}, N $_{IMD}$,ID_{IMD},C_{Proxy} ,(C,Tag)	The proxy generates a random number and computes an Authentication Tag. It includes the random number received and the one generated by Proxy, the identifier of the target IMD (ID_{IMD}), Counter of Proxy(C_{Proxy}). Additionally, a command field (REQ) is included as a part of this message. Finally, these two random numbers together with the Authentication Tag and an encrypted version of the command are sent to the implant.
IMD → Proxy	ID_{IMD}, N_{Proxy}, N $_{IMD}$, C_{IMD} ,(C,Tag)	The implant decrypts the REQ received, computes a local version of the Authentication Tag, and checks its equality with the received value. Note that only the target implant knows the identifier (ID_{IMD}) used and the two nonces associated with the session. If this authentication fails, the implant aborts the protocol. Otherwise, the IMD generates a response and sends an Authentication Tag, which includes the two nonces and the response command (RESP). It also includes the response in encrypted form.
Proxy→IMD	ID_{IMD},C_{Proxy},(C,Tag)	The proxy decrypts the response. Then, knowing the RESP and the two nonces linked to the current session, the proxy calculates a local version of the Authentication Tag. If the received values and the computed values are equal, the reader and the implant are mutually authenticated and can perform request response. If not, the proxy disconnects and generates log. Proxy sends request which contains Identifier of IMD, Counter, encrypted data and authentication tag.
IMD → Proxy	ID_{IMD},C_{IMD},(C,Tag)	IMD decrypts the request, checks its authentication and sends response in encrypted form. If this authentication fails, the implant aborts the protocol.

Once the mutual authentication between IMD and proxy is over, Nonce is no longer used. To counter Replay Attacks, counters are used at IMD side and Proxy side.

7.10.2 Protocol 2: IMD initiating communication

IMD initiates communication when it requires some telemetry data from sensor IMD. Once an IMD has subscribed for the data for actuation purpose, Proxy is responsible for delivering the data after predefined time intervals. IMD also initiates communication in case of an emergency sensed by IMD. In this protocol, Proxy at random periods broadcasts Nonce. If an IMD wants to communicate with Proxy, it listens for the broadcasted nonce and then executes the protocol shown in Figure 7.8. Once mutual authentication is over, both may communicate further by sending request and receiving response.

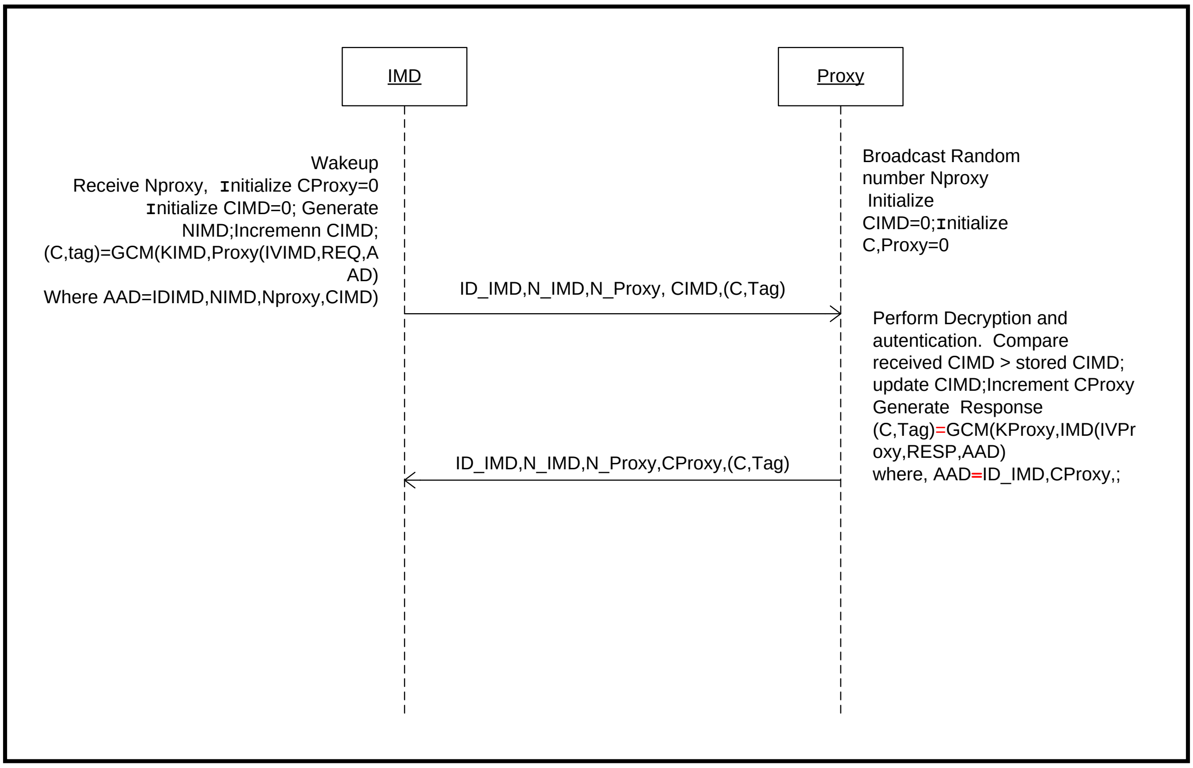

IMD
Proxy
Wakeup
Receive Nproxy, Initialize CProxy=0
Initialize CIMD=0; Generate
NIMD;Incremenn CIMD;
(C,tag)=GCM(KIMD,Proxy(IVIMD,REQ,AAD)
Where AAD=IDIMD,NIMD,Nproxy,CIMD)
Broadcast Random number Nproxy
Initialize CIMD=0;Initialize C,Proxy=0
ID_IMD,N_IMD,N_Proxy, CIMD,(C,Tag)
Perform Decryption and autentication. Compare received CIMD > stored CIMD; update CIMD;Increment CProxy
Generate Response
(C,Tag)=GCM(KProxy,IMD(IVProxy,RESP,AAD)
where, AAD=ID_IMD,CProxy,;
ID_IMD,N_IMD,N_Proxy,CProxy,(C,Tag)

Figure 7.8: Sequence Diagram for Protocol: IMD Initiating Communication

The message exchanges related to this protocol and their interpretations are given in Table 7.6. It shows the sender and the receiver of the message, the message transmitted and also what action is taken by receiver after receiving the message.

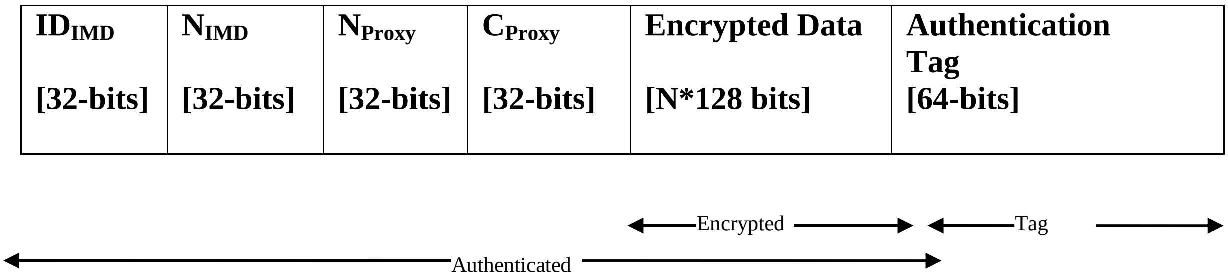
Table 7.6 Description of messages for IMD initiating communication

Sender → Receiver	**Message**	**Interpretation**
Proxy → IMD	Broadcast N_{Proxy}	IMD broadcasts an nonce at random periods, when IMD wants to communicate with proxy, it listens for the broadcast and reads the N_{Proxy}
IMD → Proxy	ID_{IMD},N_{IMD},N_{Proxy}, C_{IMD},(C,Tag)	The IMD generates a nonce by use of Physiological Value and computes an Authentication Tag. It includes the random number received and the one generated by Proxy, its identifier (ID_{IMD}), Counter of IMD(C_{IMD}). Additionally, a command field (REQ) is included as a part of this message. Finally, these two random numbers together with the Authentication Tag and an encrypted version of the command are sent to the proxy.
Proxy → IMD	ID_{IMD},N_{IMD},N_{Proxy}, C_{Proxy},(C,Tag)	The proxy decrypts the response. Then, knowing the RESP and two nonces linked to the current session, the proxy calculates a local version of the Authentication Tag. If the received values and the computed values are equal, the reader and the implant are mutually authenticated and can perform request response. If not, the proxy disconnects and generates log. Proxy sends request which contains Identifier of IMD, Counter, encrypted response and authentication tag.

7.10.3. Message Formats for Tier 1: Proxy-IMD communication

In the communication protocols, it is desired that the overhead of additional data fields due to introduction of security transformations should be minimized as far as possible. The message formats that we have designed for the above described sequence of messages between IMD and Proxy are shown in Figure 7.9.

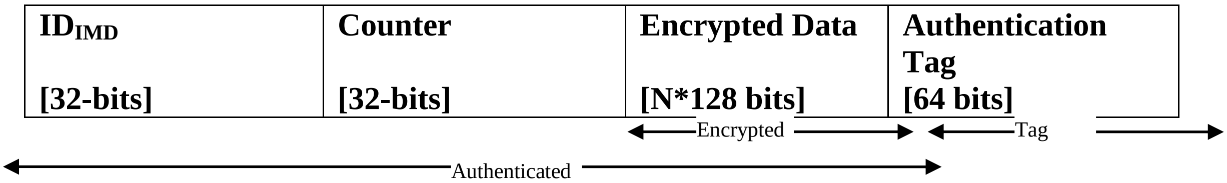
1) Authentication request message send by Proxy to IMD



ID$_{IMD}$	N$_{IMD}$	N$_{Proxy}$	C$_{Proxy}$	Encrypted Data	Authentication Tag
[32-bits]	[32-bits]	[32-bits]	[32-bits]	[N*128 bits]	[64-bits]

FIGURE 7.9 (a) Format of authentication request made by Proxy

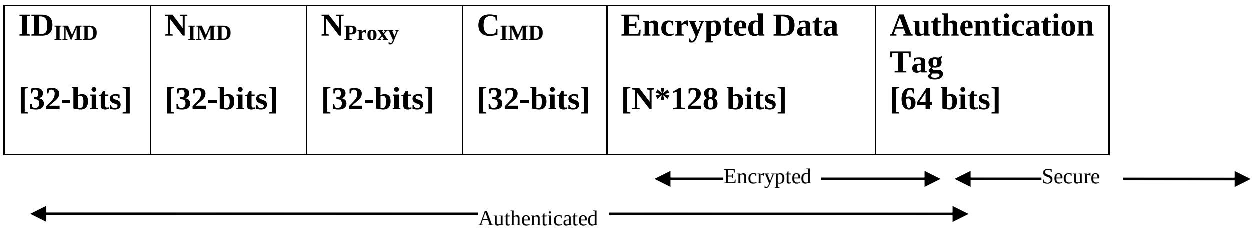
2) Request and Response message between Proxy and IMD



ID$_{IMD}$	Counter	Encrypted Data	Authentication Tag
[32-bits]	[32-bits]	[N*128 bits]	[64 bits]

FIGURE 7.9 (b) Format of request and response messages

3) Authentication Message: IMD to Proxy



ID$_{IMD}$	N$_{IMD}$	N$_{Proxy}$	C$_{IMD}$	Encrypted Data	Authentication Tag
[32-bits]	[32-bits]	[32-bits]	[32-bits]	[N*128 bits]	[64 bits]

FIGURE 7.9 (c) Format of authentication requests made by IMD

7.11 Description of proposed protocol for Tier 2: Proxy and External Device Communication

The protocols for topic based publish-subscribe communications via the Proxy for External Devices is described in this section. Before commencement of the protocol, EDs are registered with proxy during registration phase and store each other's public key.

Tier II uses publish-subscribe communication model [150] with proxy as the mediator between IMDs and EDs. The Proxy performs conversion of IMD response to Publish Message and IMD request to Subscribe Message. Communication may not only be one-to-one, but can also be many-to-one, one-to-many, or many-to-many. Data (telemetry messages and commands) to be communicated or requested is organized into topics which are uniquely identified by a name and stored in the proxy during device registration. Proxy receives topic-data from publisher device and in turn forwards it to the intended subscriber device. Proxy is aware of Publisher's and Subscriber's identity, and authenticates them on behalf of IMD. In a topic based Publish-Subscribe paradigm, instead of addressing messages to actual recipient, sender application puts the name of a topic and delivers it to the proxy. The proxy then sends the message to all the devices that have subscribed to messages on that topic and are currently available (active).

Some Examples of Mapping of Biometric data to Topics are shown in TABLE 7.7.

Table 7.7 Examples of Mapping of Biometric data to Topics

Request Messages	Topic
Hemoglobin	HMB
Blood Glucose	BG
Blood Pressure	BP
Temperature	TEMP
Blood Flow	BF
Emergency	EMER

The states of External Device are given in Table 7.8.

Table 7.8 States of External Device maintained by Proxy

State	Description
Registered	A device for which information is available with the proxy. IMDs are registered and paired with proxy using secret key. EDs are registered using their public key.
Active	A registered device when sends join request to the proxy to Publish or Subscribe data. The mutual authentication between Proxy and IMD occur using Digital Signature. For IMDs the state is always active.
Unregistered	A device which wants to communicate with IMD but is not registered. It needs to be registered with the proxy in case of Normal Condition. But in case of an Emergency Condition when the patient's life is at stake, ED can directly start communicating with the proxy bypassing registration.
Inactive	A device which is registered with the proxy but is not currently associated with the proxy for Publishing or Subscribing to Telemetry Data. EDs can be inactive but IMDs are always active.

The notations used by us for describing the protocols are shown in Table 7.9.

Table 7.9 Description of notations used in Tier – 2 communications

Notations	Generation Technique	Description
PU_{Proxy}	ECC	Public Key of Proxy
PR_{Proxy}	ECC	Private Key of Proxy
PU_{Dev}	ECC	Public Key of Device
PR_{Dev}	ECC	Private Key of Device
K_{Topic_Master}	AES-CTR	Topic wise Master Secret Key generated by proxy, shared with registered subscriber of topic when they are active. Renewed when a subscriber joins or leaves the proxy. This provides forward and backward security.
K_{Topic_Pub}	K_{Topic_Pub} =Hash(PSV_{Pub},K_{Topic_Master})	Publisher Key, K_{Topic_Pub} is generated by proxy from master key K_{Topic_Master} by using PSV_{Pub} for a publisher for each Topic. It is shared with registered publisher of topic when they are active. It is renewed when a subscriber joins or leaves the proxy as Topic Master Key for the topic changes.
PSV_{Pub}	AES-CTR	Publisher Specific Value (PSV) generated by Proxy for each Publisher Device for each topic. This PSV is used to generate K_{Topic_Pub} for each publisher.
Id_{Device}	-	Unique Id of External Device
N_{Device}	-	Nonce generated by External Device
N_{Proxy}	-	Nonce generated by Proxy
DS_{Device}	DS_{Device}=E(PR_{Device},H(Id_{Proxy} ,$Id_{Device,}$ N_{Device}))	Digital Signature generated by External Device for authentication [152]
DS_{Proxy}	DS_{Device}=E(PR_{Proxy},H(Id_{Proxy} ,$Id_{Device,}$ $N_{Device,}$ N_{Proxy}))	Digital Signature generated by Proxy for authentication [152]
Id_{Topic}		This Identifier value is mapped with Topic name
TS	By device publishing data	Timestamp for Data to check for message freshness.

7.11.1. Protocol: Communication between Proxy and ED as Publisher

When an external device (ED) wants to publish data for a Topic registered with Proxy, initial message exchange occurs prior to topic data publication for mutual authentication. When the publisher wants to publish topic data, it first sends a join message to the Proxy using the device identifier, nonce and digital signature for publisher authentication and authorization. When the authentication and authorization is successful, proxy sends a join acknowledgment message to the publisher along with its digital signature for mutual authentication. The publisher requests for the topic key for a particular Topic which is send to it after encrypting with Publisher's Public Key along with the Publisher Specific Value (PSV) allocated to the Publisher. The publisher can encrypt the data and send data to proxy which stores the topic data for forwarding it to the subscribers. The sequence diagram for communication between Proxy and External Device acting as Publisher is shown in Fig. 7.10.

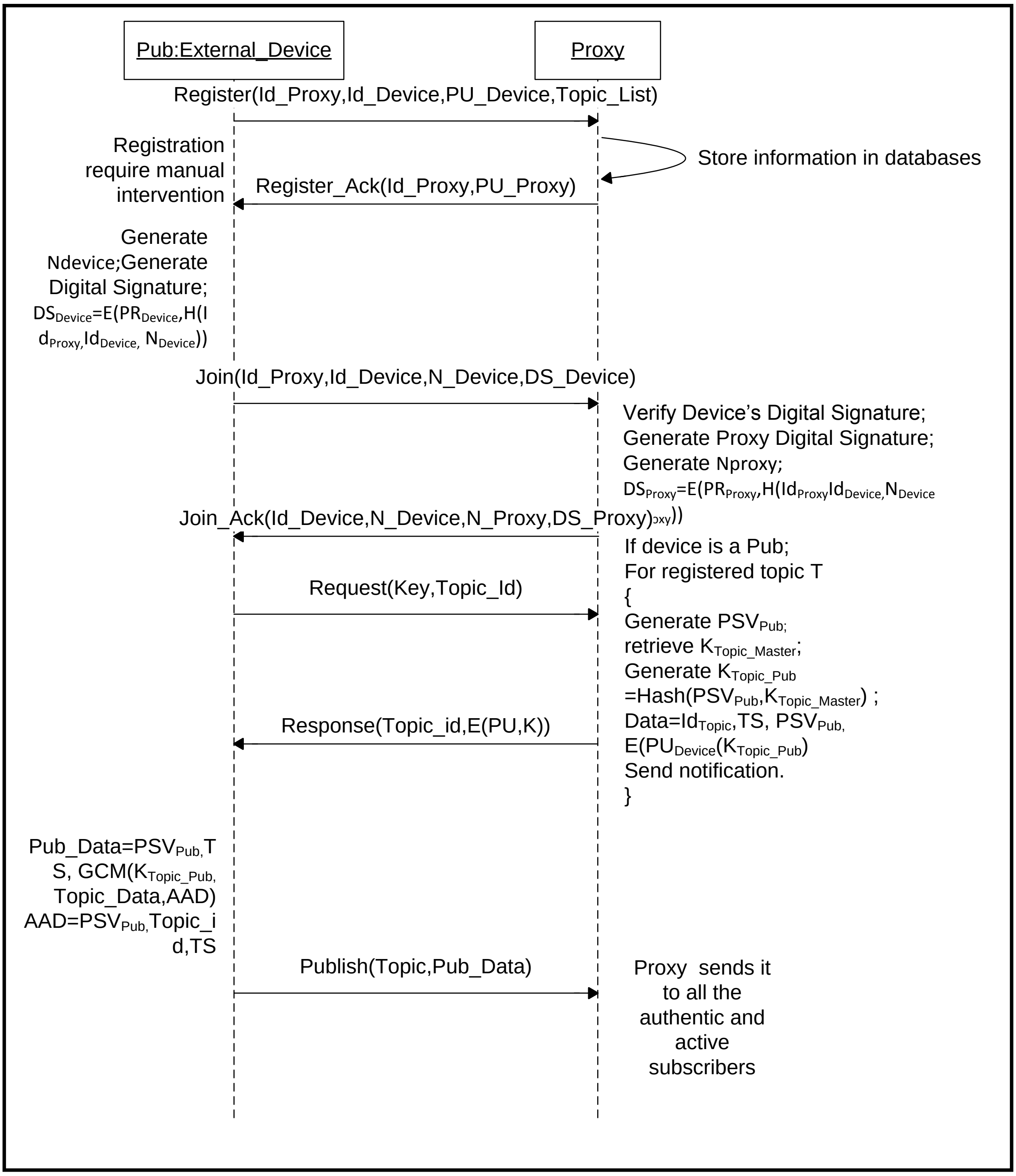

Pub:External_Device
Proxy
Register(Id_Proxy,Id_Device,PU_Device,Topic_List)
Registration require manual intervention
Store information in databases
Register_Ack(Id_Proxy,PU_Proxy)
Generate Ndevice;Generate Digital Signature; DS_{Device}=E(PR_{Device},H($Id_{Proxy,}Id_{Device,}$ N_{Device}))
Join(Id_Proxy,Id_Device,N_Device,DS_Device)
Verify Device's Digital Signature; Generate Proxy Digital Signature; Generate Nproxy; DS_{Proxy}=E(PR_{Proxy},H($Id_{Proxy}Id_{Device,}N_{Device}$ $_{oxy}$))
Join_Ack(Id_Device,N_Device,N_Proxy,DS_Proxy)
If device is a Pub;
For registered topic T
{
Generate $PSV_{Pub;}$
retrieve K_{Topic_Master};
Generate K_{Topic_Pub}
=Hash(PSV_{Pub},K_{Topic_Master}) ;
Data=Id_{Topic},TS, $PSV_{Pub,}$
E(PU_{Device}(K_{Topic_Pub})
Send notification.
}
Request(Key,Topic_Id)
Response(Topic_id,E(PU,K))
Pub_Data=$PSV_{Pub,}$TS, GCM($K_{Topic\_Pub,}$ Topic_Data,AAD)
AAD=$PSV_{Pub,}$Topic_id,TS
Publish(Topic,Pub_Data)
Proxy sends it to all the authentic and active subscribers

Figure 7.10: Sequence Diagram for communication between Proxy and External Device as Publisher

The message exchanges related to this protocol and their interpretations are given in Table 7.10. The table shows the sender and the receiver of the message, the message transmitted and also what action is taken by receiver after receiving the message.

Table 7.10 Description of messages for Proxy and Publisher External Device communication

Sender→Receiver	Message	Interpretation
Publisher ED → Proxy	Join(Id_{Proxy},Id_{Device},N_{Device},DS_{Device})	Join request is sent by a registered ED along with its Nonce and Digital Signature
Proxy→ Publisher ED	Join_Ack(Id_{Device},N_{Device},N_{Proxy},DS_{Proxy})	Proxy verifies the Digital Signature of ED, generates its Nonce value, generated a digital signature and sends to the ED.
Publisher ED → Proxy	Request(Key,Topic Id)	Before publishing Topic data, Publisher ED requests the Proxy to send the Topic Publish Key and Publisher Specific Value (PSV). The proxy on receiving this request verifies the role of the device for the Topic. It generates PSV_{Pub} and retrieves Topic Master Key retrieve K_{Topic_Master}. Generation of Publisher Key K_{Topic_Pub} =Hash(PSV_{Pub},K_{Topic_Master}) ;
Proxy→ Publisher ED	Response(Topic_id, E(PU,K))	The Publisher receives Id_{Topic},TS,$PSV_{Pub,}$,E(PU_{Device}(K_{Topic_Pub}) From which it decrypts the Publisher Key, K_{Topic_Pub} and uses it to encrypt the Topic data using AES-GCM. Pub_Data: TS,PSV_{Pub} ,(C,Tag) (C,Tag)=GCM(K_{Topic_Pub},Topic_Data, AAD) AAD= PSV_{Pub},Topic_id,TS
Publisher ED → Proxy	Publish(Topic_id,Pub_Data)	When proxy receives Topic data from publisher it mediates the data to all the authentic and active subscribers for that topic.

7.11.2. Protocol: Communication between Proxy and ED as Subscriber

When the subscriber wants to receive topic data, it first sends a join message containing digital signature to the Proxy for authentication and authorization. When the authentication and authorization of the subscriber is successful, proxy sends a join acknowledgment message to the subscriber along with its own digital signature for mutual authentication. When subscriber requests topic master key, Proxy encrypts the key with public key of subscriber and sends it. When data is available for the topic, the proxy notifies the subscribers that are active. The sequence diagram for communication between Proxy and External Device acting as Subscriber is shown in Figure 7.11.

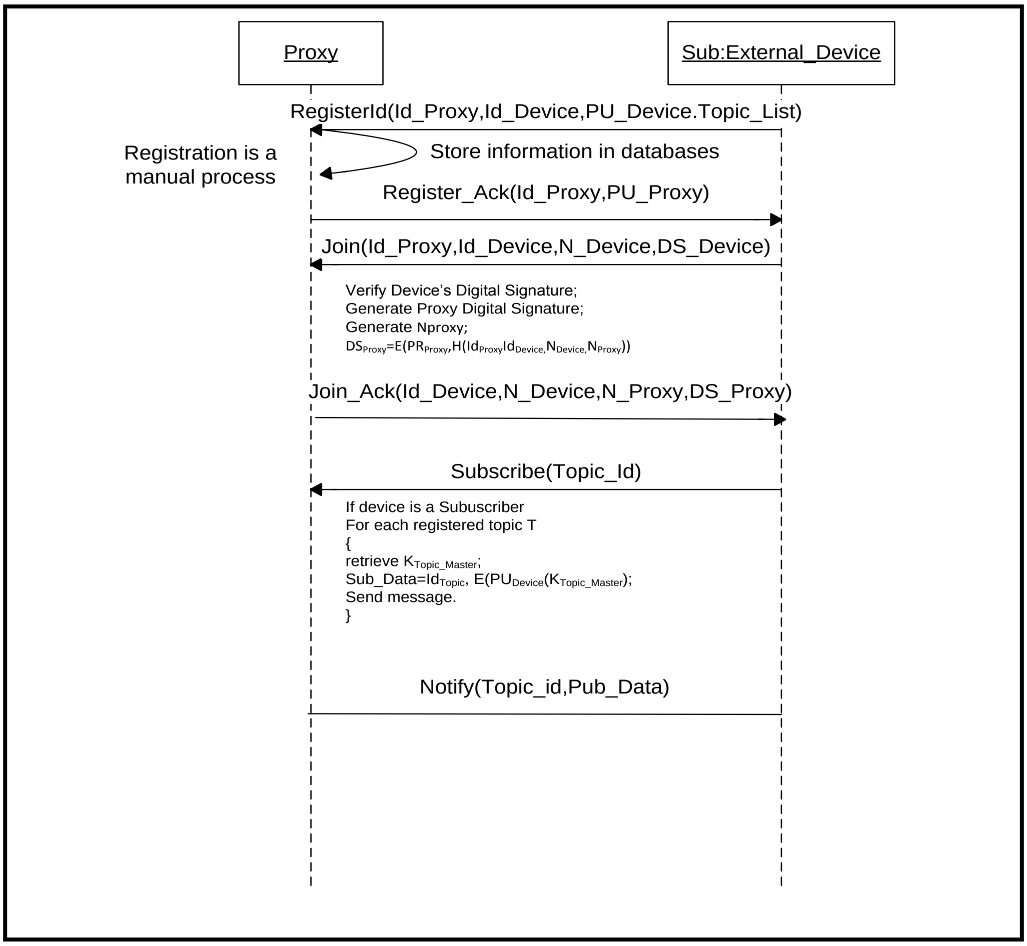

Proxy
Sub:External_Device
RegisterId(Id_Proxy,Id_Device,PU_Device.Topic_List)
Registration is a manual process
Store information in databases
Register_Ack(Id_Proxy,PU_Proxy)
Join(Id_Proxy,Id_Device,N_Device,DS_Device)
Verify Device's Digital Signature;
Generate Proxy Digital Signature;
Generate Nproxy;
DSProxy=E(PRProxy,H(IdProxyIdDevice,NDevice,NProxy))
Join_Ack(Id_Device,N_Device,N_Proxy,DS_Proxy)
Subscribe(Topic_Id)
If device is a Subuscriber
For each registered topic T
{
retrieve KTopic_Master;
Sub_Data=IdTopic, E(PUDevice(KTopic_Master);
Send message.
}
Notify(Topic_id,Pub_Data)

Figure 7.11: Sequence Diagram for communication between Proxy and External Device as Subscriber

The message exchanges related to this protocol and their interpretations are given in Table 7.11. The table shows the sender and the receiver of the message, the message transmitted and also what action is taken by receiver after receiving the message.

Table 7.11 Description of messages for Proxy and Subscriber External Device communication

Sender→Receiver	Message	Interpretation
Subscriber ED → Proxy	Join(Id_{Proxy},Id_{Device},N_{Device},DS_{Device})	Join request is sent by a registered ED along with its Nonce and Digital Signature
Proxy→ Subscriber ED	Join_Ack(IdDevice,NDevice,NProxy,DSProxy)	Proxy verifies the Digital Signature of ED, generates its own Nonce value, generates a digital signature and sends to the ED.
Subscriber ED → Proxy	Subscribe(TopicList)	Proxy validates the role of ED as a subscriber for a particular Topic and retrieves the Topic Master Key. It encrypts the Topic Master Key by Public Key of subscriber and sends it to the ED.
Proxy→ Subscriber ED	Notify(Topic_id,Pub_Data)	If Published data is available, Proxy notifies the Publisher Specific Value (PSV) and the Encrypted data to the ED. ED at its end generates the Key to decrypt data by

		performing one way hash on the Topic Master Key using PSV send with the encrypted message. Subscriber checks timestamp to ensure data is not replayed. The recipient subscriber uses receive PSV_{Pub} to check for data authentication. Uses the topic master key K_{Topic_Master} to generate the Publisher Key to decrypt the Topic data. K_{Topic_Pub} =Hash(PSV_{Pub},K_{Topic_Master}) Plaintext Topic_Data= GCM(K_{Topic_Pub},Topic_Data)

7.12 Essential Functions Provided by Proxy

Proxy device is the heart of our protocol and performs many essential functions to support the proposed security model. We provide a list of such functions and the data structures used for that.

7.12.1. Topic Management

The topics need to be defined and preregistered with the proxy when an IMD is registered. For every topic defined with the proxy, either publisher or subscriber or both are IMD devices. Therefore for every topic proxy that maintains, the corresponding request message and response message for communicating with the IMD is also stored. The format of Data Structure for Topic management is shown in Table 7.12.

Table 7.12 Topic Management Database

Field	Description
Topic Identifier	This field uniquely identifies the topics.
Topic Name	This is a user defined name given to the topic
Description	This is the additional information stored for the topic.
Topic Flag	This flag is used by Proxy to check if the topic is Active or Inactive. If the topic is active it means that for this topic there are IMDs associated with proxy.
Topic Master Key(K_{Topic_Master})	This is the master key related to the topic which is shared with the subscribers. It is also used to generate the Publisher key by using Publisher Specific Value (PSV).
Request Message (REQ)	This stores the format of request message to be sent to the IMD for requesting data for this topic. This is required if Publisher is IMD.
Response Message (RESP)	This stores the format of response message to be sent to the IMD when data is available on this topic. This is required if the Subscriber is an IMD.
Probe Time Interval	With the topic we also keep a probe time interval after which a new request is send to IMD by proxy if there are active subscribers for that topic. This allows the IMD to remain in sleep state this saving battery power.

If no active subscribers are present no data request goes to the IMD in order to save IMD battery power. If a new subscriber joins during the time interval for which response is already received from proxy, the same response is sent to the subscriber without generating a new request for the IMD. For a single Topic if there are two or more IMDs publishing data, Proxy may adopt any cast wherein request may be send to any one of the IMD in general or to a specific IMD by taking into consideration the amount of battery available with IMD. If publisher and subscriber both for a particular Topic are IMDs then proxy can implement a timer which when fires, proxy requests data from Publisher IMD (by request-response) and then delivers the data to subscriber IMD (by request-response).

7.12.2. Device Management

All external devices including proxy have a set of public and private keys. During registration of an ED, information like its Identifier, Topic Identifier, Role of the ED for the topic, and Public key are stored as shown in Table 7.13.

Table 7.13 Device Information Database

Field	Description
Device Id	This field uniquely identifies the device.
Device Name	This is a user defined name given to the device.
Device Type	This field is used to store the type of registered device which can be an either IMD or ED.
Device Flag	For an ED, this flag is true, it means that the device is currently active and is associated with proxy by sending a join message. By default for all IMDs currently installed, the flag is true.
Device Description	This is the additional information about the device.
Public Key	If the device is an ED, this field stores the RSA or ECC Public key of the device.

7.12.3. Access Management

ED that is a Subscriber for a topic when registered with proxy can only receive messages that they are authorized for. Publisher when registered with proxy can publish to one or more topics only if are authorized for the topic. For every topic the role and validity period for a device is stored as shown in Table 7.14.

Table 7.14 Device Access Control Database

Field	Description
Topic Id	This field uniquely identifies a topic.
Device Id	This field uniquely identifies the device associated with the topic.
Role	This field stores the role of the device which can be either publisher or subscriber.
Valid From	This stores the validity period start date and time.
Valid To	This stores the validity period end date and time.

7.12.4. Key Management

The Topic Master key needs to be renewed from time to time. Especially when an ED subscriber joins or leaves the proxy, the master key needs to renewed in order to render forward and backward security. In order to do this, Proxy generates new Topic master key (K_{Topic_Master}) and notifies all the subscribers who are active for a specific topic. For each active publisher of the topic, it uses the Publisher Specific Value (PSV_{Pub}) and to derive a new Topic Publish Key(K_{Topic_Pub}) and notifies it to the corresponding publisher.

7.12.5. Emergency aware Access Management

When an IMD initiates communication with the proxy device in case of a patient emergency condition the Proxy notifies all the active external devices by publishing on emergency topic. This enables the health provider to take an immediate action for patient health restoration. In case none of the registered ED are available in the vicinity, the proxy may allow unregistered external device to access the IMD for a specific period of time which can be calculated with the help of solution given in [124].

7.13 Deployment Model

The proposed security model can be deployed by use of an additional handheld device like PDA or smartphone which is readily available. It may provide a user interface for registration of IMD and EDs and also for defining topics. Figure 7.12 show the deployment model with IMDs implanted in human body and registered with proxy device, EDs registered with proxy device and proxy device providing the functionalities for Tier-1 and Tier-2. To keep it simple we do not show the data that is being passed along with the methods.

Methods used for Tier-1 Request-response protocols between IMD and Proxy are mentioned below:

2.1 Registration: This method allows registration of an IMD with the proxy device. A pair of secret key is shared and the IMD related information is stored.

2.2 Mutual Authentication: This method allows IMD and Proxy to authenticate each other before requesting for data or generating response.

2.3 Request: This method allows proxy device or the IMD to generate a request and send it in a secure manner.

2.4 Response: This method allows proxy device or the IMD to generate a response and send it in a secure manner.

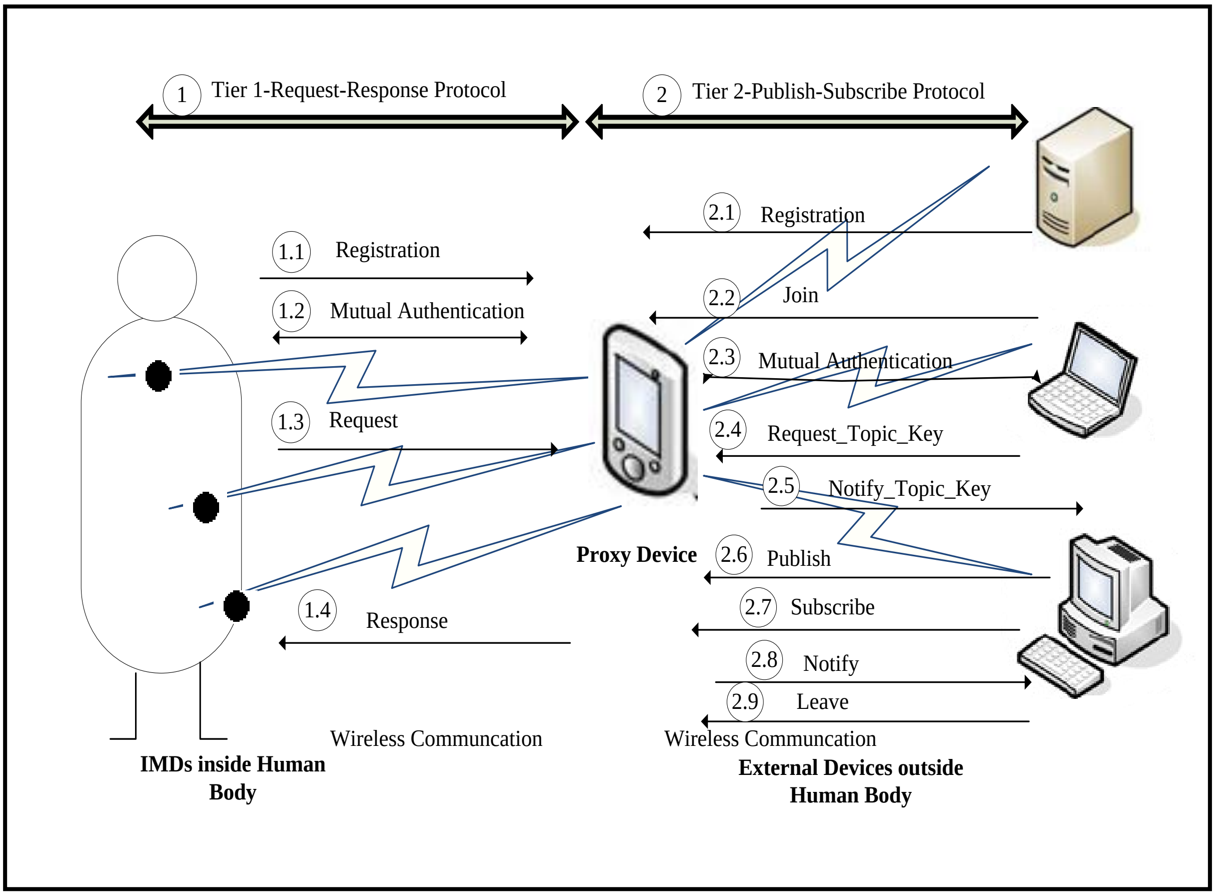


Figure 7.12 Deployment Model

The Methods used for Tier-1 Request-response protocols between Proxy and External Device are mentioned below:

2.1 Registration: This method allows use of User Interface for input of the device information like public key, a set of topics and the role of the device for that topic.

2.2 Join: This method allows a registered external device to get associated with the proxy device.

2.3 Mutual Authentication: This method allows ED and Proxy to authenticate each other before requesting for data or generating response.

2.4 Request_Topic_Key: This method allows ED to request for a symmetric key related to a topic either for encrypting data to be published for the topic of for decrypting the received topic data.

2.5 Notify_Topic_Key: This method allows Proxy to notify the Topic related key to the active EDs.

2.6 Publish: This method allows Publisher ED to publish data related to a topic in secure manner. It also allows proxy to publish data on behalf of IMD.

2.7 Subscribe: This method allows Subscriber ED to subscribe to data feed related to a topic. It also allows proxy to subscribe to data feed on behalf of IMD.

2.8 Notify: This method allows Proxy to send available data for a topic to a subscriber.

2.9 Leave: This method allows EDs to get disassociated from the proxy device when they no longer want to receive data.