NXDN®

# NXDN Technical Specifications

## Part 1:
## Air Interface

## Sub-part D:
## Security

**NXDN TS 1-D Version 1.3**

**November 2011**

**NXDN Forum**

## Contents



 

## Figures



## Tables



 

**Disclaimer**

The information presented here is intended to be for clarification and/or information purpose only, and care has been taken to keep the contents as neutral and accurate as possible.

The use or practice of contents of the information may involve the use of intellectual property rights ("IPR"), including pending or issued patents, or copyrights, owned by one or more parties. The NXDN Forum makes no search or investigation for IPR, nor the NXDN Forum makes no arrangement of licensing negotiation for IPR between the user and the owner of IPR.

All warranties, express or implied, are disclaimed, including without limitation, any and all warranties concerning the accuracy of the contents, its fitness or appropriateness for a particular purpose or use, its merchantability and its non-infringement of any third party's IPR.

The NXDN Forum expressly disclaims any and all responsibilities for the accuracy of the contents and makes no representations or warranties regarding the content's compliance with any applicable statute, rule or regulation.

The NXDN Forum shall not be liable for any and all damages, direct or indirect, arising from or relating to any use of the contents contained herein, including without limitation any and all indirect, special, incidental or consequential damages (including damages for loss of business, loss of profits, litigation, or the like), whether based upon breach of contract, breach of warranty, tort (including negligence), product liability or otherwise, even if advised of the possibility of such damages.

The foregoing negation of damages is a fundamental element of the use of the contents hereof, and these contents would not be published by the NXDN Forum without such limitations.

**Document Copyrights**



**Trademarks**

NXDN® is a registered trademark of JVC KENWOOD Corporation and Icom Incorporated.

AMBE+2™ is a trademark of Digital Voice Systems, Inc.

# 1. Introduction

This document describes the services relating to security of an NXDN system and the operational procedures for secrurity in a conventional system and a trunked radio system.

The security services include encryption to protect user information such as voice and text data, and authentication to prevent unauthorized use. A trunked radio system addressed in this document means Type-C trunked system of which the trunking procedure is described in REF [3].

# 2. References

References documents are listed below. This document and the references are mutually supplemented.

| | |
|---|---|
| REF [1] | Part 1-A Common Air Interface Version 1.3 |
| REF [2] | Part 1-B Basic Operation Version 1.3 |
| REF [3] | Part 1-C Trunking Procedures Version 1.3 |
| REF [4] | FIPS 46-3 |
| REF [5] | FIPS 81 |
| REF [6] | FIPS 197 |
| REF [7] | NIST 800-38A |

# 3. Abbreviations

To help understanding this document, abbreviations are listed below.

| | |
|---|---|
| AES | Advanced Encryption Standard |
| BCCH | Broadcast Control Channel |
| CAC | Common Access Channel |
| CAI | Common Air Interface |
| CCCH | Common Control Channel |
| CR | Conventional Repeater |
| CRS | Conventional Repeater Site |
| DES | Data Encryption Standard |
| DMO | Direct Mode Operation |
| FACCH1 | Fast Associated Control Channel 1 |
| FACCH2 | Fast Associated Control Channel 2 |
| IV | Initialization Vector |
| L1 | Layer 1 |
| L2 | Layer 2 |
| L3 | Layer 3 |
| LICH | Link Information Channel |
| MS | Mobile Station |
| OFB | Output Feedback |
| PBX | Private Branch Exchange |
| PSTN | Public Switched Telephone Network |
| RAN | Radio Access Number |
| RCCH | RF Control Channel |
| RDCH | RF Direct Channel |
| RTCH | RF Traffic Channel |
| SACCH | Slow Associated Control Channel |
| SU | Subscriber Unit |

| | |
|---|---|
| TC | Trunking Controller |
| TR | Trunking Repeater |
| TRS | Trunking Repeater Site |
| UDCH | User Data Channel |
| UPCH | User Packet Channel |
| USC | User Specific Channel |
| VCH | Voice Channel |

# 4. Security

This section outlines the security methodology of an NXDN system.

## 4.1. Outline

For Public Safety parties who require security, for users who do not want to be eavesdroped the communications, or for SMR operators who want to prevent spoofing, it is important to protect information transferred on an air that can be received easily by anyone. The following security functions are provided to meet these requirements.

- Encryption function for voice information of vocoder.
- Encryption function for user data information including texts and images.
- Authentication function

These functions can be used for both trunked radio system and conventional system. These functions can be implemented in the system or subscriber units as options, and are available depending on the required security levels.

The object of encryption is user information only, and control information such as Group ID is exempted from encryption.

## 4.2. Type of Encryption

The requirements for security for voice information or user data information varies in each system, from the low level security such as a voice inversion scramble method in analog FM radio to a high level security which uses more advanced and undecipherable method.

The following three encryption algorithms are provided.

No Guard Level

This is normal calls without encryption.

Low Guard Level

This is low level security and encrypts user information using a bit scramble processing with PN sequence.
This is described as "Scramble Encryption".

High Guard Level

This is high level security and encrypts user information using DES algorithm specified in REF [4] and AES algorithm specified in REF [6]. They are described respectively as "DES Encryption" and "AES Encryption".

### 4.3. Authentication Method

An important function for system operators is to confirm whether a SU that accesses to a TRS or CRS is an eligible SU. The method of identifying an eligible SU is the authentication function, which verifies the SU is eligible using a unique number which the manufacturer writes into the memory of each SU at the factory.

This unique number is called Electronic Serial Number (ESN) and it is 48 bits length. A SU transmits the authentication value calculated using the ESN stored in the memory, and the authentication facility verifies using the received value whether the SU is eligible to access the site.

# 5. Encryption Procedure

This section outlines the encryption function and its operational procedure.

## 5.1. Layer for Encryption

The encryption shall be done on an end-to-end basis, and the encrypted state shall be maintained throughout the communication path to ensure the security. Since an object of encryption is voice data or user data information, the encryption protocol resides in an upper layer than layer 3 and is implemented inside SUs to handle mainly the voice and user data, or if necessary, inside network consoles.

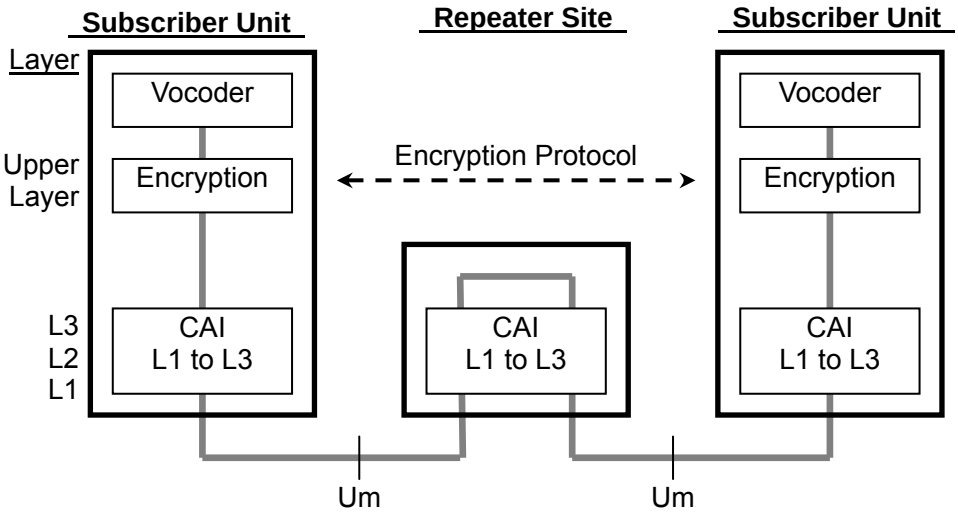


Figure 5.1-1 Layer for Encryption Protocol

## 5.2. Encryption Processing

Figure 5.2-1 shows the procedure to apply the encryption function to the voice data of vocoder.

The encryption processing is added between the voice coding processing and the FEC coding processing in a normal vocoder processing without encryption. The decryption processing is processed in reverse.

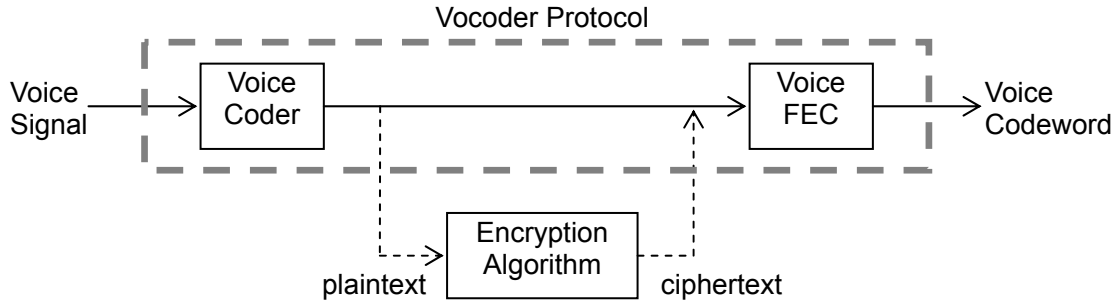


Figure 5.2-1 Encryption Processing Diagram

In the case of encryption to user data information such as a text message, the same encryption processing is applied by regarding the user data as a plaintext. An output ciphertext is embedded into the User Data field in a packet as well as the case of non-encrypted data packet, so an encrypted data call is achieved.

## 5.3. Encryption Algorithm

### 5.3.1. Scramble Encryption

Scramble encryption is an encryption algorithm that is a random bit inversion processing using a bitwise exclusive-or operation between bit sequence of voice or other data and PN bit sequence. Figure 5.3-1 shows a block diagram of this algorithm.

PN sequence uses the polynomial $P(x) = X^{15} + X + 1$ which has a repeat period of 32,767 bits, and an encryption key is used as default for the PN sequence. As PN sequence is generated by the 15-stage shift register, the encryption key is selectable from 32,767 keys except All zero.

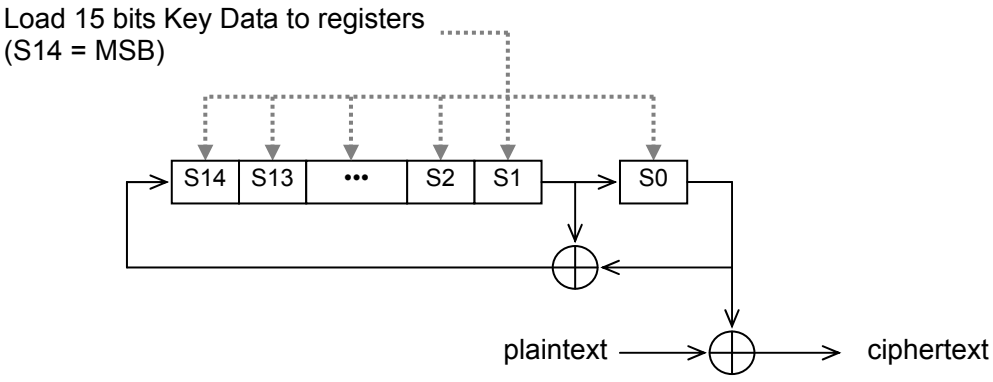


Figure 5.3-1 Scramble Encryption Algorithm

### 5.3.2. DES Encryption

DES encryption is the encryption algorithm that applies DES algorithm to bit sequence of voice or other data. DES encryption is 64-bit block encryption and uses an encryption key of 56 bits (but this is expressed in 64 bits with parity bits).

#### 5.3.2.1. OFB Mode

Modes of Operation for DES encryption are defined in REF [5] and each mode has the different error propagation characteristic in a decrypting process. Since a bit error easily occurs in radio communication path under a fading environment, the mode where the error doesn't spread to other bits when the bit error is caused is suitable for the encryption of wireless communications. Therefore, Output Feedback Mode (OFB Mode) is adopted for the reason that one bit error arises after decrypting when one bit error is contained in the ciphertext.

OFB Mode uses a data sequence of Initialization Vector as well as an Encryption Key. The generation method of IV is described in Section 5.3.2.2.

Figure 5.3-2 shows the processing flow of DES encryption and DES decryption.

 

e-i) Set the Encryption Key Data used.

e-ii) Input a seed value to IV Generator and obtain an output data sequence from IV Generator.

e-iii) Input the output data sequence into Input Block by connecting the switch of Input Block to IV Generator.

e-iv) In the first round of DES processing, encrypt the data sequence of Input Block and output the DES-encrypted data sequence to Output Block.

e-v) Change the switch of Input Block to Output Block side and initialize Input Block by the data sequence of Output Block again.

At this time, since the path from Output Block to EXOR opens, the encryption processing to a plaintext has not been executed yet.

e-vi) Close the path from Output Block to EXOR when the second round of DES processing is executed. Execute the EXOR operation between the data sequence of plaintext and the data sequence of Output Block that is obtained by DES-encrypting the data sequence of re-initialized Input Block, and obtain the ciphertext.

d-i) The seed value for IV Generator is input to IV Generator and also sent to a receiver.

d-ii) The receiver processes steps e-i) to e-vi) in manner similar to the transmitter, executes the EXOR operation between the data sequence of received ciphertext and the data sequence of Output Block, and decodes the plaintext.

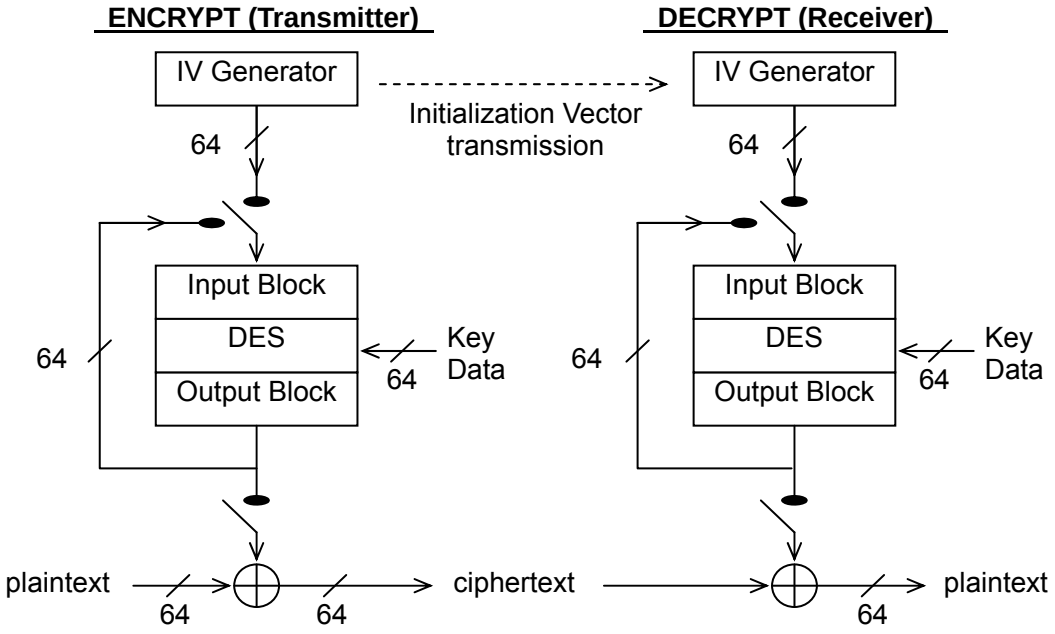


Figure 5.3-2 DES Encryption Algorithm

#### 5.3.2.2. 64-bit Initialization Vector Generation

IV Generator in DES encryption generates a random bit sequence of 64-bit length in order to initialize Input Block in OFB Mode. Figure 5.3-3 shows the block diagram of IV Generator.

IV Generator is expressed by the polynomial $P(x) = 1 + X^{15} + X^{27} + X^{38} + X^{46} + X^{62} + X^{64}$ and consists of the Linear Feedback Shift Register (LFSR) of 64-stage shift register.

When beginning transmission, a 64-bit random data as seed value is loaded into the registers b63-b0 in LFSR. The generation method of seed value is not defined. Since the IV Generator in receiver needs the seed value, the seed value is sent to the receiving side too. Next, the first 64-bit output of IV Generator is obtained by shifting the LFSR 64 times.

In the receiving side, the received seed value is loaded into the LSFR, and then the 64-bit output is generated from the IV Generator by using the same procedures as the transmitting side.

When the following encryption session begins while transmitting, the current register data b63-b0 which is used to generate IV for the following encryption session is sent to the receiving side. With the same procedures as the first encryption session, the LFSR is shifted 64 times and the output data of IV Generator is generated. The subsequent encryption sessions repeat the same procedures.

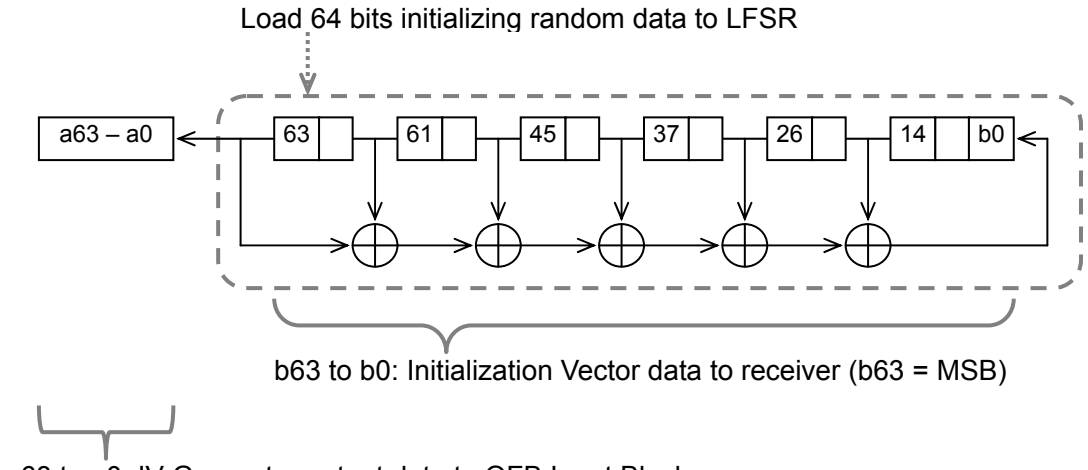


Figure 5.3-3 64 bits IV Generator

### 5.3.3. AES Encryption

AES encryption is the encryption algorithm that applies AES algorithm to bit sequence of voice or other data. AES encryption is 128-bit block encryption and can use an encryption key of 256 bits.

#### 5.3.3.1. OFB Mode

Modes of Operation for AES encryption are defined in REF [7], and OFB Mode is adopted for the same reason as DES encryption.

OFB Mode uses a data sequence of Initialization Vector as well as an Encryption Key. The generation method of IV is described in Section 5.3.3.2.

Figure 5.3-4 shows the processing flow of AES encryption and AES decryption. The differences with the flow of DES encryption are the IV generation processing, the bit length of IV and the encryption key length.

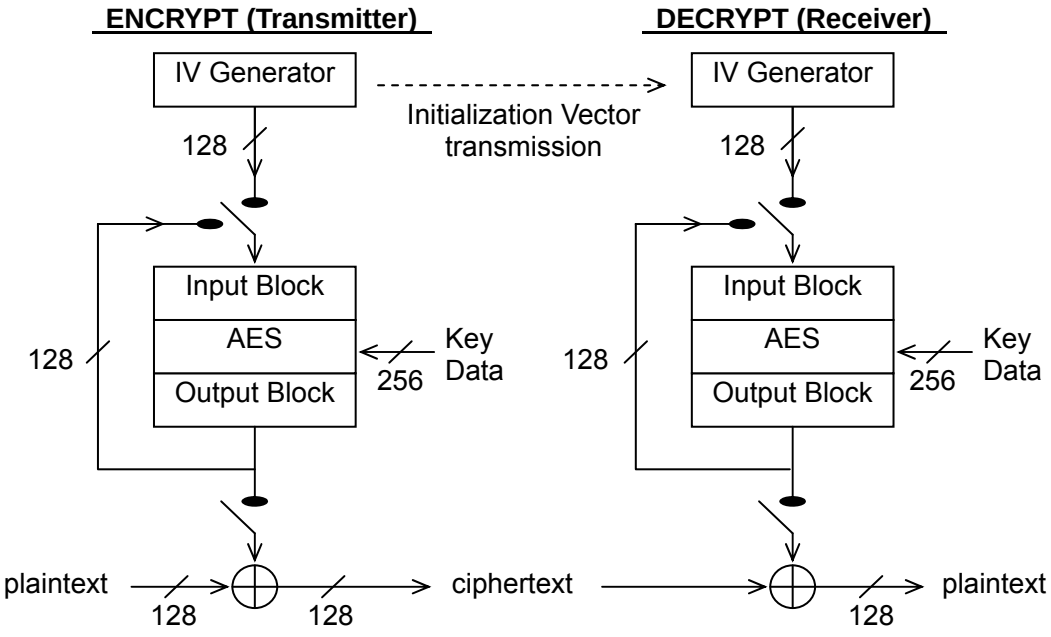


Figure 5.3-4 AES Encryption Algorithm

#### 5.3.3.2. 128-bit Initialization Vector Generation

IV Generator in AES encryption generates a random bit sequence of 128-bit length in order to initialize Input Block in OFB Mode. Figure 5.3-5 shows the block diagram of IV Generator.

IV Generator is expressed by the polynomial $P(x) = 1 + X^{15} + X^{27} + X^{38} + X^{46} + X^{62} + X^{64}$ and consists of the Linear Feedback Shift Register (LFSR) of 64-stage shift register and the 64-stage shift register to buffer the output data of the LFSR.

When beginning transmission, a 64-bit random data as seed value is loaded into the registers b63-b0 in LFSR. The generation method of seed value is not defined. Since the IV Generator in receiver needs the seed value, the seed value is sent to the receiving side too. Next, the LFSR is

shifted 64 times and the output data is stored into the 64-stage buffer a63-a0. Finally, the 128-bit data composed of the LFSR and the buffer is the first output of IV Generator.
In the receiving side, the received seed value is loaded into the LSFR, and then the 128-bit output is generated from the IV Generator by using the same procedures as the transmitting side.
When the following encryption session begins while transmitting, the current register data b63-b0 which is used to generate IV for the following encryption session is sent to the receiving side. With the same procedures as the first encryption session, the LFSR is shifted 64 times and the output data of IV Generator is generated. The subsequent encryption sessions repeat the same procedures.

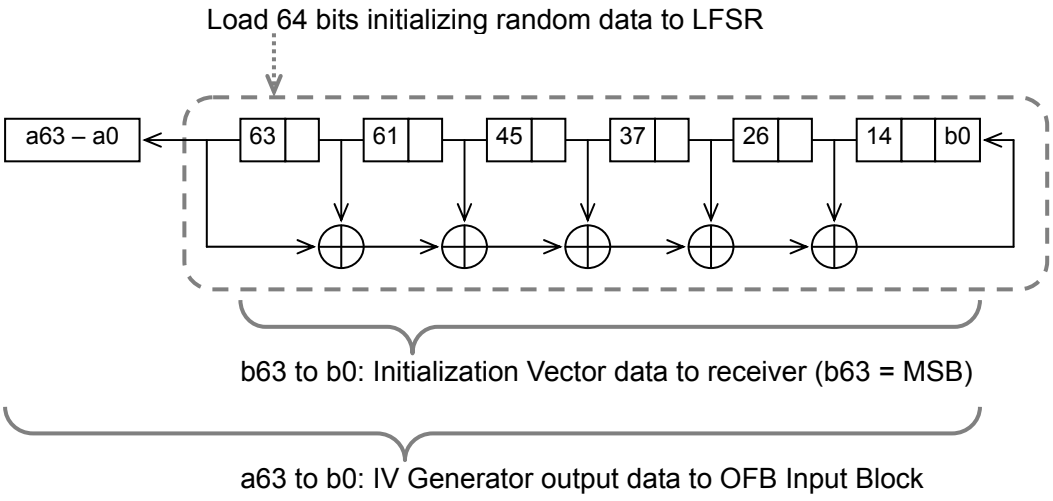


Figure 5.3-5 128 bits IV Generator

## 5.4. Procedure for Encrypted Call

This section outlines the procedures of voice calls and data calls using an encryption.

### 5.4.1. Information Element in Encrypted Call

A VCALL message is used in voice calls, while a DCALL message and a SDCALL_REQ message are used in data calls. In DES and AES encryptions, a VCALL_IV message and a SDCALL_IV message are used in addition to those messages. Table 5.4-1 shows the information elements in these messages used to identify whether encrypted call or not.

| Information Element | Length | Description |
|---|---|---|
| Cipher Type | 2 bits | Information element to identify encryption algorithm.<br>3 types of encryption algorithms can be identified. |
| Key ID | 6 bits | Information element to indicate alias to identify encryption key. Up to 64 Key IDs can be identified for each Cipher Type. |
| Initialization Vector | 64 bits | Information element to use as a seed value of LFSR in DES and AES encryptions. |

Table 5.4-1 Information Element in Encrypted Calls

Table 5.4-2 shows other elements needed inside of transceivers.

| Element | Length | Description |
|---|---|---|
| Encryption Key | 15 bits (Scramble)<br><br>56 bits (DES)<br><br>256 bits (AES) | Key data actually used in encryption algorithm.<br>In Scramble encryption, this is default of 15-stage shift register.<br>The encryption key and Key ID shall be interrelated inside of transceivers. |
| Key Name | Optional | Name of key to let users easily identify an encryption key. |

Table 5.4-2 Transceiver Internal Information for Encrypted Calls

### 5.4.2. Scope of Encryption

The scope of encryption in a voice call is the voice coding data sequence generated by a voice coding processing to an audio signal.

AMBE+2 EHR = 49 bits voice coding data
AMBE+2 EFR = 72 bits voice coding data

The scope of encryption in a data call is the data sequence of Use Data field which includes Message CRC and Null too. To be more specific, the scope of encryption is Octet 2-21 in a DCALL (User Data) message and Octet 2-15 (UPCH) and Octet 2-9 (FACCH1) in a SDCALL_REQ (User Data) message.

 

### 5.4.3. Bit Allocation of Initialization Vector

Initialization Vector as the seed value of LFSR is sent by using a VCALL_IV message in a voice call and using a DCALL (Header) message and a SDCALL_IV message in a data call.

Figure 5.4-1 shows the relationship between the Initialization Vector information element in the message and the LSFR registers of Figure 5.3-3 and Figure 5.3-5.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | b63 | b62 | b61 | b60 | b59 | b58 | b57 | b56 |
| 2 | b55 | b54 | b53 | b52 | b51 | b50 | b49 | b48 |
| 3 | | | | | | | | |
| 4 | | | | | | | | |
| 5 | Initialization Vector | | | | | | | |
| 6 | | | | | | | | |
| 7 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 |
| 8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |

Figure 5.4-1 Bit Allocation of Initialization Vector

### 5.4.4. Encrypted Voice Calls

This section describes how to apply the encryption processing to a voice call.

The following rules shall be observed:

- It is prohibitive to change the encryption algorithm, including non-encryption mode, during a call.
- It is prohibitive to change the encryption key during a call.

#### 5.4.4.1. Voice Calls with Scramble Encryption

In scramble encryption, one encryption session consists of 4 frames in order to synchronize with the period of sending a VCALL in a SACCH with superframe structure. As one frame contains 4 VCHs, the PN sequence of the scramble encryption is applied for 16 VCHs in one encryption session. Figure 5.4-2 shows the relationship between vocoder voice coding data in VCH and PN sequence for EHR and EFR.

The encryption procedure of EHR is described below. The procedure of EFR is the same as that of EHR, but the only difference from EHR is the relationship between vocoder voice coding data and PN sequence .

At the start of transmission, the 15-stage register of the scramble pattern generator is initialized by a proper encryption key. The first bit of PN sequence output from the scramble pattern generator is expressed as P0 here. The first bit of the voice coding data of the first VCH in the frame of the first SACCH to send a VCALL is performed the exclusive-or operaiton with P0. Since the voice coding data is 49 bits length, P0 to P48 are applied to the first VCH and P49 to P97 are applied to the second VCH. The PN sequence is applied to the following third and fourth VCH every 49 bits, and the same operation is performed to frames containing the second to the fourth SACCH. In the end, P783 of the PN sequence is applied to the last bit of voice coding data of the fourth VCH in the fourth frame, and one encryption session ends.

Since the first SACCH starts from the next superframe again, the scramble pattern generator is initialized by the encryption key and the previous PN sequence is applied to the next four frames. As shown in REF [2] Section 4, in a superframe structure of 9600bps/EHR, VCH is used in the first and third frame, and FACCH1 is used in the second and fourth frame. Even in this case, the relationship between voice coding data and PN sequence shown in Figure 5.4-2 is maintained. Therefore only P0 to P195 for the first frame and P392 to P587 for the third frame are used, and the other PN sequences for the second and fourth frame are not used.

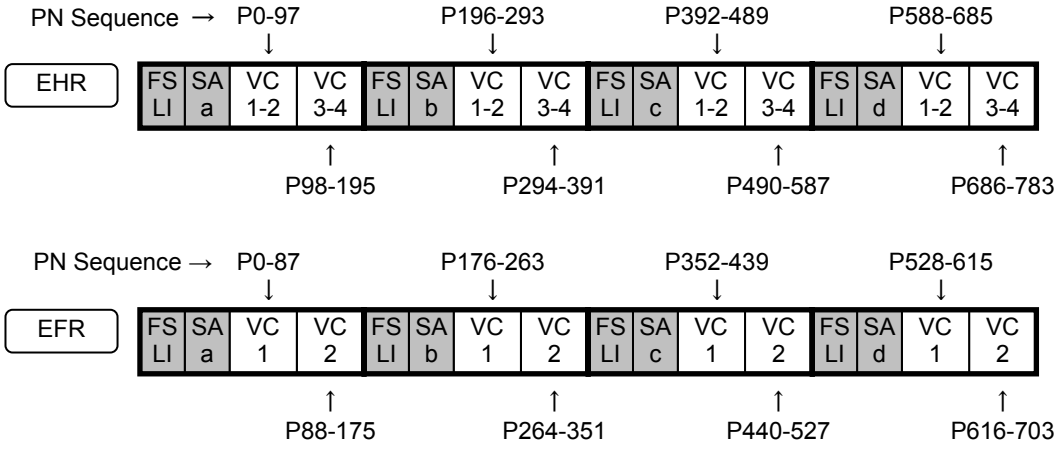


Figure 5.4-2 Relationship between PN sequnece and VCH in Scramble Encryption

#### 5.4.4.2. Voice Calls with DES/AES Encryption

In DES and AES encryptions, since VCALL message carries the information relating to encryption and VCALL_IV message carries the Initialization Vector, VCALL message and VCALL_IV message are sent in SACCH alternately. Therefore, two superframes composed of 8 frames is equal to one encryption session.

(1) DES Encryption

Figure 5.4-3 shows the relationship between vocoder voice coding data in VCH and DES OFB output data sequence for EHR and EFR. The encryption procedure of EHR is described below. The procedure of EFR is the same as that of EHR, but the only difference from EHR is the relationship between vocoder voice coding data and DES OFB output data sequence.

Since the DES encryption is 64-bit block encryption, each OFB iteration outputs 64 bits data. The first bit of the voice coding data of the first VCH of the first frame in the first superframe is exclusive-ORed with the first bit of 64-bit OFB output data. Since the voice coding data is 49 bits length, the last bit of the first OFB output data is exclusive ORed with the 15th bit of the 2nd VCH. Next, the first bit of the 2nd OFB output data is applied to the 16th bit of the 2nd VCH. Subsequently, the new OFB output data that is generated when all 64 bits of OFB output data are used up is applied to the voice coding data of VCH in order, and finally, the first bit to the 32nd bit of the 25th OFB output data is applied to the latter part of the 4th VCH of the 4th frame in the 2nd superframe, and one encryption session finishes.

 

(2) AES Encrypiton

Figure 5.4-4 shows the relationship between vocoder voice coding data in VCH and AES OFB output data sequence for EHR and EFR. The encryption procedure of EHR is described below. The procedure of EFR is the same as that of EHR, but the only difference from EHR is the relationship between vocoder voice coding data and AES OFB output data sequence.

Since the AES encryption is 128-bit block encryption, each OFB iteration outputs 128 bits data. The first bit of the voice coding data of the first VCH of the first frame in the first superframe is exclusive-ORed with the first bit of 128-bit OFB output data. Since the voice coding data is 49 bits length, the last bit of the first OFB output data is exclusive ORed with the 30th bit of the 3rd VCH. Next, the first bit of the 2nd OFB output data is applied to the 31st bit of the 3rd VCH. Subsequently, the new OFB output data that is generated when all 128 bits of OFB output data are used up is applied to the voice coding data of VCH in order, and finally, the first bit to the 32nd bit of the 13th OFB output data is applied to the latter part of the 4th VCH of the 4th frame in the 2nd superframe, and one encryption session finishes.

(3) Procedure for 9600bps/EHR

As well as the scramble encryption of Section 5.4.4.1, the relationship between the voice coding data and OFB output data sequence shall be maintained even if using FACCH1. Therefore, the OFB output data for the 2nd and 4th frames is not used.

(4) Relationship between Initialization Vector and Superframe

Each encryption session uses the different Initialization Vector. Initialization Vector is an information element carried by VCALL_IV message. The relationship between the sending order of frame and VCALL_IV message is shown in Figure 5.4-5 in order from the 1st frame sent in the beginning of transmission.

The VCALL message which indicates DES encryption or AES encryption and the VCALL_IV message containing the 1st Initialization Vector are sent in the 1st frame using two FACCH1. Next, the same message as VCALL message sent by FACCH1 is sent by SACCH of the 1st superframe consisting of the 2nd frame to the 5th frame, and VCALL_IV message containing the 2nd Initialization Vector is sent by SACCH of the 2nd superframe consisting of the 6th frame to the 9th frame. From the 2nd frame to the 9th frame constitutes one encryption session, and the OFB output data using the Initialization Vector sent by FACCH1 of the 1st frame is applied to these frames.

Initialization Vector contained in VCALL_IV message of the 2nd superframe is the data generated in the procedure shown in Section 5.3.2.2 and Section 5.3.3.2 in order to use in the next encryption session.

In the superframes after this, VCALL message and VCALL_IV message are alternately sent by SACCH, and the encryption session of every two superframes is repeated by using new Initialization Vector contained in VCALL_IV message.

 

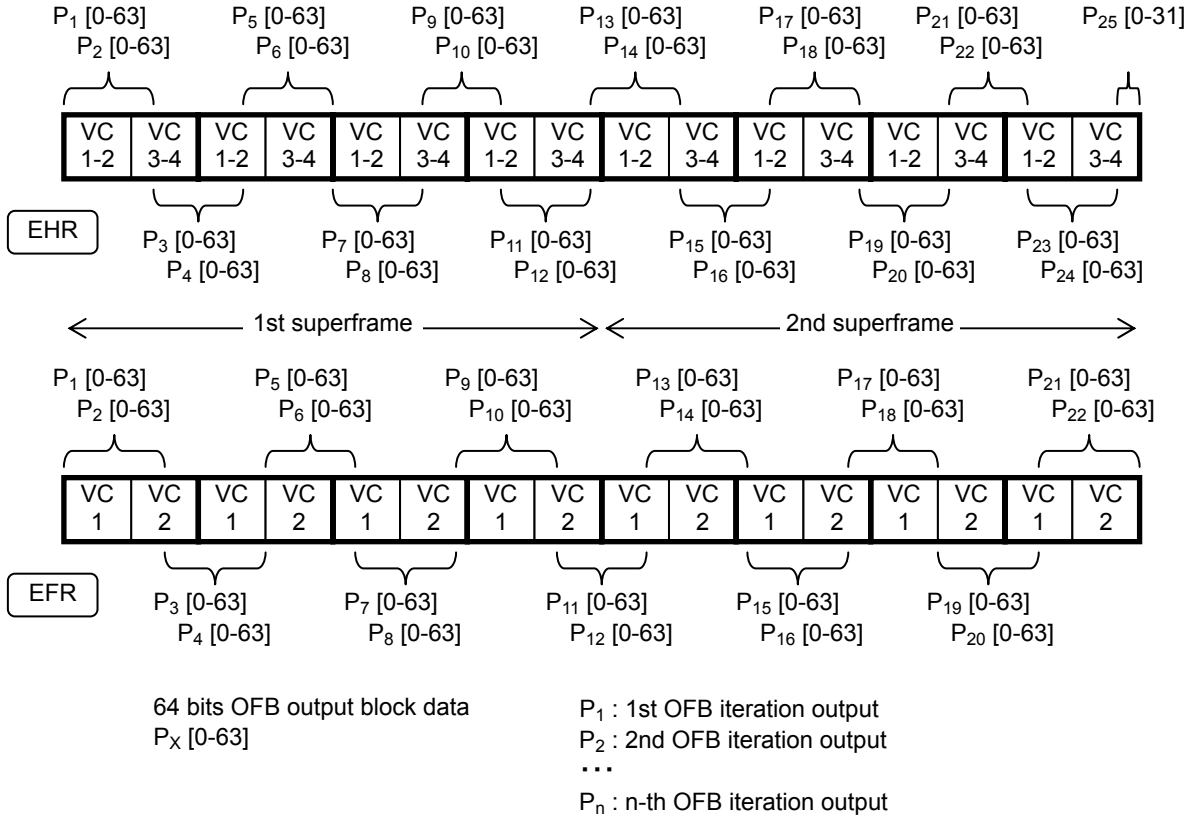


Figure 5.4-3 Relationship between DES OFB output data and VCH

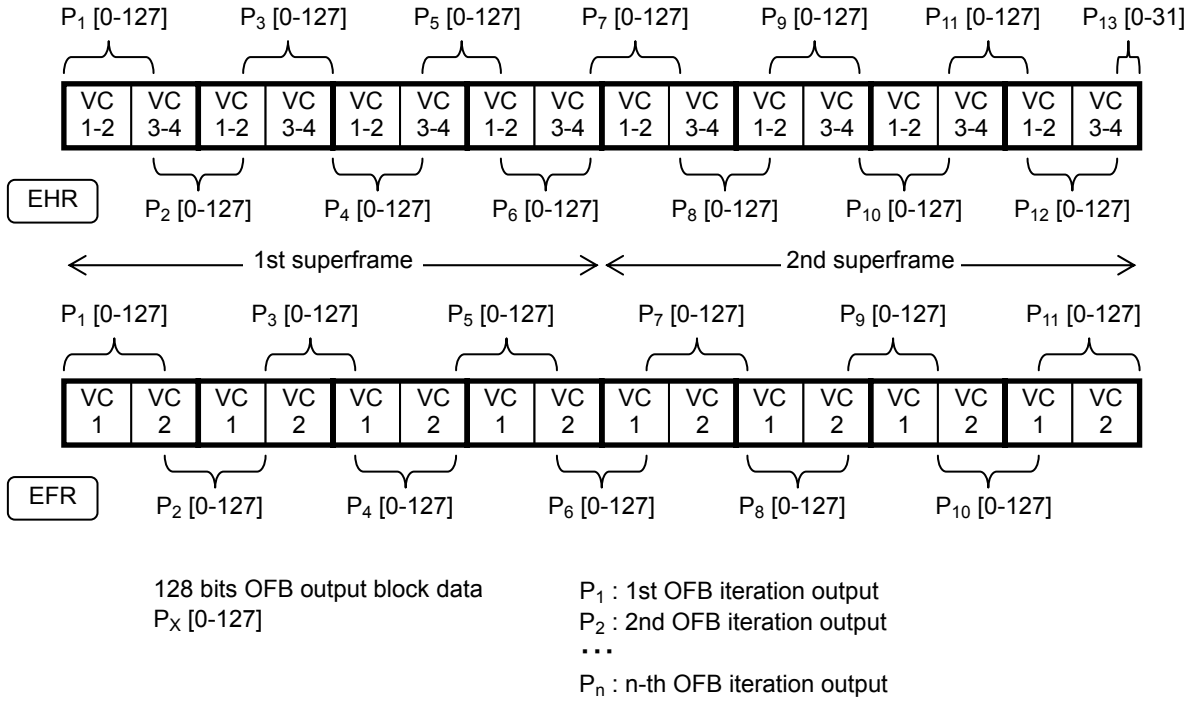


Figure 5.4-4 Relationship between AES OFB output data and VCH

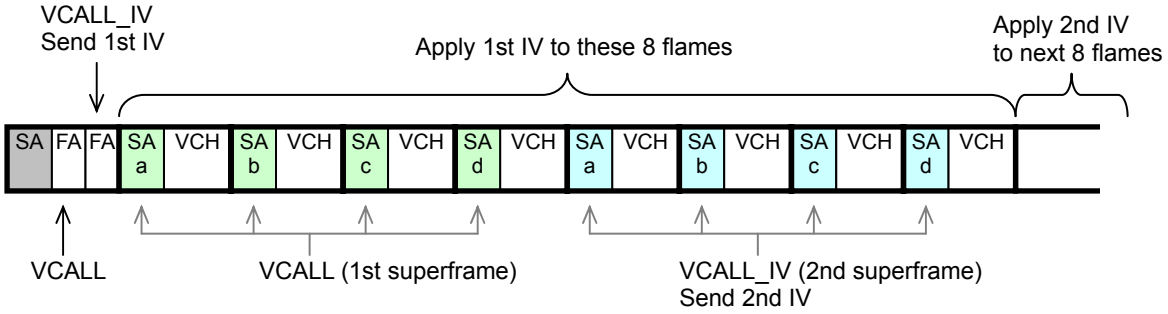


Figure 5.4-5 Usage of Initialization Vector

### 5.4.5. Encrypted Data Calls

This section describes how to apply the encryption processing to a data call.

#### 5.4.5.1. Data Calls with Scramble Encryption

Since data calls do not support the Late Entry function of voice calls, the encryption session is not based on frames like a scramble encrypted voice call, but one packet is handled as one encryption session. UDCH in the first frame has a DCALL in Header format which Cipher Type and other information elements are embedded in. Figure 5.4-6 shows the relationship between PN sequence and User Data field in the DCALL sent on UDCH.

As shown here, 32,767-bit PN sequence is sequentially applied to the User Data field in a DCALL in User Data format sent on UDCH until the last frame is sent.

Additionally, in the case of a SDCALL_REQ sent by UPCH on RCCH or a SDCALL_REQ sent by FACCH1 on RTCH/RDCH, the above procedure is exactly applied to those. However the User Data field has a different amount of data between DCALL and SDCALL_REQ, thus the PN sequence must be modified to the proper position as shown in Figure 5.4-7. SDCALL_REQ is sent by Long CAC on an inbount RCCH or by Single Message format of CAC on an outbound RCCH. The octet sizes of Long CAC and a SDCALL_REQ (User Data) are the same, while the octet size of a Single Message format of an outbound CAC is two octets larger than that of SDCALL_REQ (User Data). The area of two octets is outside of a SDCALL_REQ and does not apply an encryption. Thus the relationship described in Figure 5.4-7 applies both of an inbound and outbound RCCH.

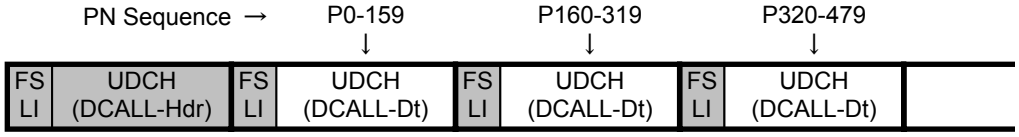


Figure 5.4-6 Relationship between PN sequence and UDCH in Scramble Encryption

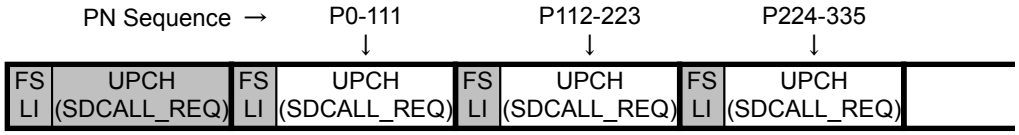


Figure 5.4-7 Relationship between PN sequence and UPCH in Scramble Encryption

#### 5.4.5.2. Data Calls with DES/AES Encryption

In the DES and AES encryptions, one encryption session consists of one packet as well as the scramble encryption.

UDCH in the first frame has a DCALL(Header) message which Cipher Type and other information elements are embedded in. Figure 5.4-8 shows the relationship between the DES

 

OFB output data sequence and the User Data field in the DCALL (User Data) message sent on UDCH. Also Figure 5.4-9 shows the relationship for the AES OFB output data sequence.
As shown here, Initialization Vector is generated at the time of a transmitting start, and the OFB output data with 64-bit length (DES) or 128-bit length (AES) is repeatedly applied to the User Data field in DCALL (User Data) message. In the case where two or more packets are sent out continuously, Initialization Vector for the next packet, i.e., next encryption session, is generated by the procedure shown in Section 5.3.2.2 and Section 5.3.3.2, and is sent by DCALL (Header) message of the next packet.

Additionally, the procedure of sending SDCALL_REQ message on RCCH using UPCH and the procedure of sending SDCALL_REQ message on RTCH/RDCH using FACCH1 are the same as the above. However, since the SDCALL_IV message is used in order to send the Initialization Vector, the number of frames which constitutes the one packet is one more frame than the scramble encryption. And since the information capacities of User Data Area contained in DCALL message and SDCALL_REQ message differ, it is necessary to correct appropriately the place which applies the OFB output data sequence as shown in Figure 5.4-10 and Figure 5.4-11. Refer to Section 5.4.5.1 for notes concerning the difference of the size of Long CAC and Outbound CAC.

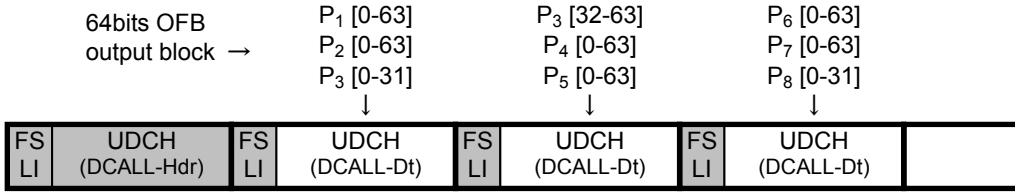


Figure 5.4-8 Relationship between DES OFB output data and UDCH

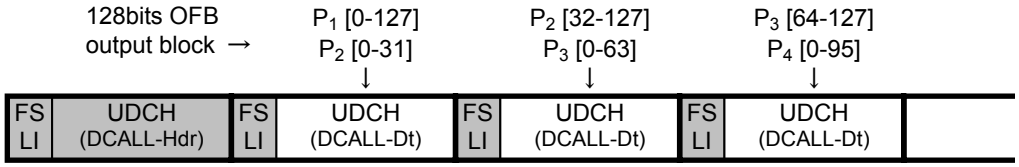


Figure 5.4-9 Relationship between AES OFB output data and UDCH

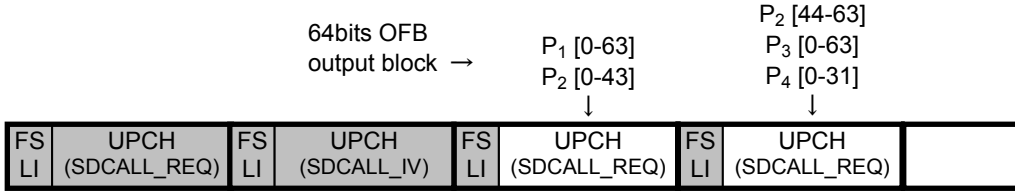


Figure 5.4-10 Relationship between DES OFB output data and UPCH

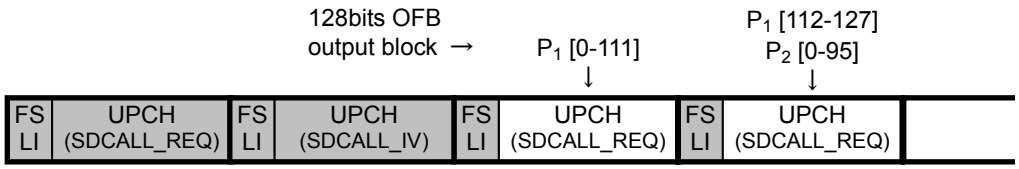


Figure 5.4-11 Relationship between AES OFB output data and UPCH

### 5.4.6. Simultaneous Data Call

While making a voice call, a data call can be simultaneously carried out at low speed by using FACCH1. In this case, encryption is completely independent in voice calls and data calls. Different encryption algorithms, including non-encryption, can be used between a voice call and a data call.

The procedure to apply scramble encryption and DES/AES encryption to voice calls is described in Section 5.4.4.1 and Section 5.4.4.2 respectively. Even if a VCH is replaced to a FACCH1, the positional relationship between the VCH and the encryption bit sequence is unchanged, and the encryption bit sequence corresponding to the VCH replaced to FACCH1 is unused and discarded by both senders and recipients. Since this rerationship is the same as the one between superframe and encryption bit sequence in 9600bps/EHR as described in Section 5.4.4.1, the encryption bit sequence corresponding to the frame for FACCH1 in the 9600bps/EHR superframe is unused and discarded by senders and recipients.

The procedure to apply scramble encryption and DES/AES encryption to data calls is described in Section 5.4.5.1 and Section 5.4.5.2 respectively. The Information elements for encryption in data calls are contained in a SDCALL_REQ (Header) message and are also contained in a SDCALL_IV message in DES/AES encryption optionally, therefore the algorithm used in a data call can be specified independently of that of a voice call.

The encryption session for voice call is integer multiples of superframes, but VCH of any frames can be replaced to FACCH1 without considering the superframe structure.

 

# 6. Authentication Procedure

This section outlines the authentication procedure and its operational procedure.

## 6.1. Layer for Authentication

Authentication is a function to confirm the validity of a SU that is requesting to connect to the system. An authenticating side equipped with an authentication facility checks for a response from the SU as an authenticated side and confirms the validity. The authentication facility may be installed in a SU, CRS, TRS or a console connected to these and is not specified in this document.

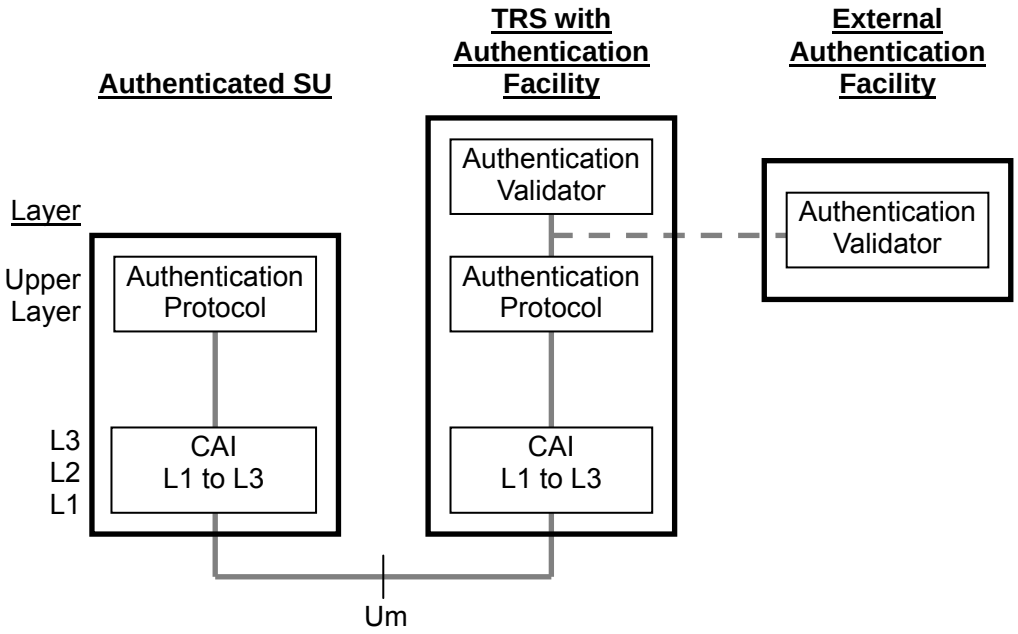


Figure 6.1-1 Layer for Authentication Protocol

## 6.2. Authentication Protocol

A block diagram of the authentication process is presented in Figure 6.2-1. The following information elements are used in the authentication process:

- ESN (48-bit length)
- Authentication Parameter (16-bit length)
- Authentication Value (56-bit length)

The authentication facility generates the Authentication Parameter using the Authentication Parameter Generator and sends it to an authenticated SU. The Authentication Parameter is also stored in the authentication facility. An authenticated SU calculates the Authentication Value using the Authentication Value Encoder from the received Authentication Parameter and the prestored ESN, and sends back the value to the authentication facility. The authentication facility extracts an ESN of the authenticated SU from the received Authentication Value and the stored

 

Authentication Parameter, and confirms the validity of the authenticated SU based on whether the extracted ESN exists in the valid ESN database stored in the authentication facility.

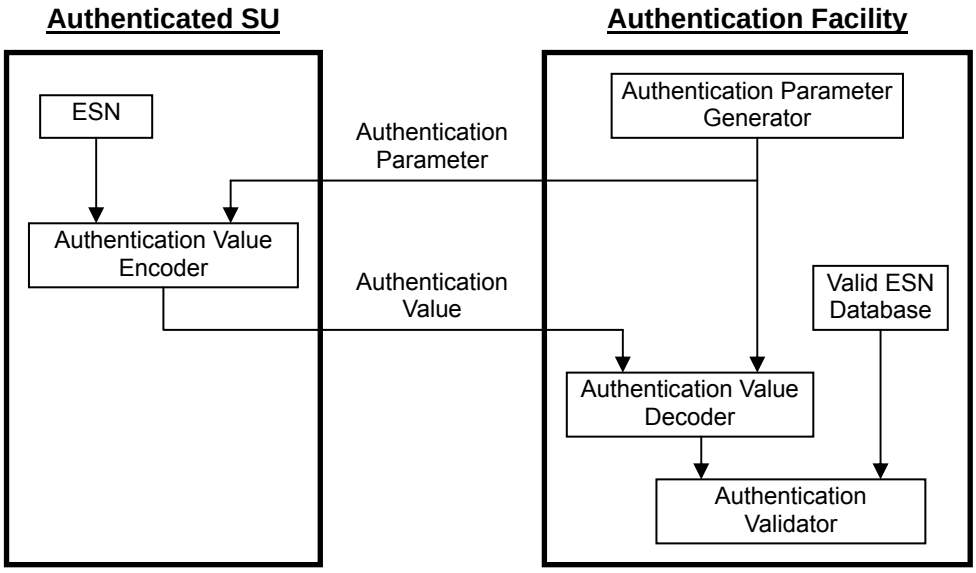


Figure 6.2-1 Block Diagram for Authentication

## 6.3. Cryptography Technique

This section describes the cryptography technique used in the authentication process.

### 6.3.1. Authentication Parameter Generator

Authentication Parameter is 16-bit length, and the generation method of its bit sequence is not specified in this document. Any bit sequence other than all zero can be used, and the bit sequence shall be random and different each time it is generated.

### 6.3.2. Authentication Value Encoder/Decoder

This section describes the encoding procedure to obtain the Authentication Value. 8-bit parity is calculated using the polynomial $G(x) = X^8 + X^5 + X^4 + 1$ from a 48-bit ESN to build a 56-bit data sequence. Figure 6.3-1 shows the parity calculation using the shift register and the bit allocation. The default values of the all shift register are set to 1.

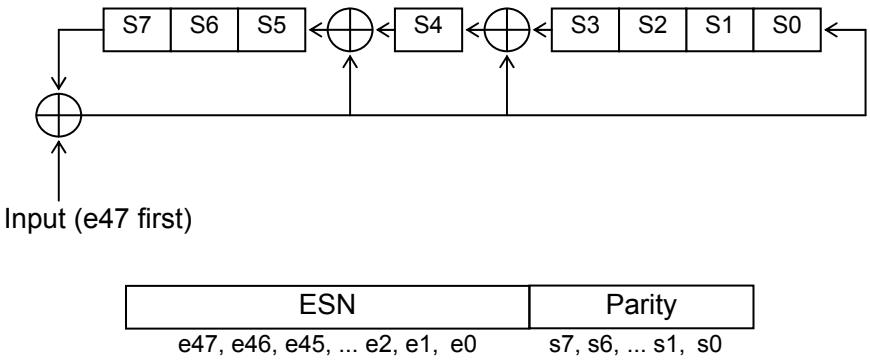


Figure 6.3-1 Parity Calculation and Bit Allocation

The Authentication Value is obtained by a bit scramble processing using an exclusive-or operation between the 56-bit data sequence and the data sequence of a polynomial P(x). Figure 6.3-2 shows the bit scramble processing.

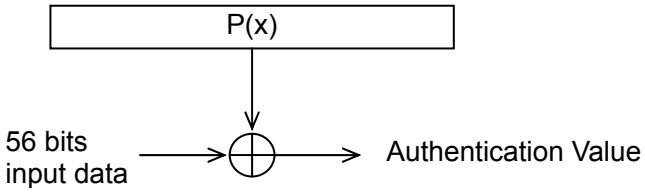


Figure 6.3-2 Scramble Method

# 7. Appendices

## 7.1. ESN

### 7.1.1. ESN Format

This section describes the 48-bit ESN format. ESN is comprised of the following 3 information elements:

Manufacturers Number (MFN)

This 8-bit information element is a unique number assigned to each manufacturer.

Manufacturer Definable Field (MFF)

This 16-bit information element is a field that can be freely defined by the manufacturer.

Serial Number

This 24-bit information element is a serial number to identify a SU uniquely in combination with Manufacturer Definable Field. This can identify 1 to 999,999 in the BCD format.

An ESN is externally expressed in a 12-character hexadecimal format.

For example, ESN of a SU of "MFN = 68 hex, MFF = 12AB hex, 100,000th unit" is expressed as following characters.

ESN = 6812AB100000

It is possible to identify up to 999,999 units in each MFF. In order to avoid the ESN overlapping, each manufacturer shall configure the MFF information element properly.

| Bit / Octet | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Manufacturer Number | | | | | | | |
| 1 | Manufacturer Definable Field | | | | | | | |
| 2 | | | | | | | | |
| 3 | Serial Number | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | | |

Figure 7.1-1 ESN Format

## 7.2. Test Vectors

### 7.2.1. Voice Channel Test Vectors

This section provides test vectors per superframe which represent an encrypted Tone Pattern.

For an input vector of EFR, there are 2 types of voice coding data having different LSBs as shown in Voice Coding Method of REF [1], and VCH test vectors are configured using them alternately.

#### 7.2.1.1. Voice Channel with Scrambled Encryption

This section presents the test vector for a voice call using scramble encryption. Key information below is written in binary number.

| Rate | Input Vector | VCH Test Vector for 1 Superframe |
|---|---|---|
| EHR | Data: 1031 Hz Tone Pattern<br>Key: 000 0000 0000 0001 | 2AA CF8 C7C CA0 847 464<br>EC8 EFF D12 758 56A 48A<br>A08 0D2 9DA CE0 81E C62<br>9F1 4FD C32 D06 B16 44D<br><br>B77 DA5 30F 731 390 CC8<br>ACC C99 F75 320 B0A 406<br>C1D 05F 618 884 016 5EF<br>F50 AFD C53 C4A 556 850<br><br>907 962 D49 035 326 3B8<br>957 B84 047 8E8 76C E6C<br>E28 6A4 E7C EA4 812 02E<br>128 CBD 810 B03 F4A 7E8<br><br>F55 FF2 1A1 99D 644 020<br>4F4 89C 91B 0A7 882 FDA<br>DE2 756 A9C 28A 23C B8D<br>0E8 CBD 835 C2F 1C3 664 |
| EFR | Data: 1011 Hz Tone Pattern<br>Key: 000 0000 0000 0001 | C641 6DDE 325E B6DB F514 B6CC 9DC3 1CB2 2FC4<br>797A A29E 6055 90CE 0D83 0753 6CB5 DB47 8A3F<br><br>6D8C 9E54 F589 ECCB 46BE 7EA3 2606 BDB3 8905<br>7A22 5894 CCBD 8299 9CE3 30B6 97B5 3873 19B9<br><br>118C 912D C8DC 7CAD 754B 5DFA 7424 CC12 415F<br>2E18 3B98 886F 93D9 30FA AFD2 6397 2CC1 B892<br><br>2866 A59D A89A FAFA 8C91 F9F2 3255 B358 3322<br>3DEF BF21 587E F9C7 E20C 2794 98E2 2201 1456 |

#### 7.2.1.2. Voice Channel with DES Encryption

This section presents the test vector for a voice call using DES encryption.

| Rate | Input Vector | VCH Test Vector for 2 super frames |
|---|---|---|
| EHR | Data: 1031Hz Tone Pattern<br>Key: ABCD EF01 2345 6789<br>IV: ABCD EF12 3456 7890 | 5A1 511 7BC 1A2 267 FA5<br>FDB 29C 8B1 6A4 3BF C9B<br>121 19A 8C7 ED5 F47 12D<br>5CD 929 438 403 89B 3E7<br><br>667 F48 309 3AE 6E8 012<br>2C7 778 404 A1A C12 99B<br>EA0 ABC 843 5BC 294 3A1<br>403 684 D93 6BA 5D8 B82<br><br>904 6C2 09B 217 777 E56<br>45C CC2 60D 1E3 FE9 8CA<br>46A F37 612 AF2 F3C 919<br>85B 75E 23B 79D 259 597<br><br>AE7 A6D 578 4C2 2E5 3D7<br>D65 63E 77F 865 2DF 7D4<br>989 42B DC8 857 F4F 52E<br>959 829 B3C 31A 908 6D9<br><br>7F9 929 683 AC1 75A 994<br>8A5 F51 2DD 6F9 04E 6D2<br>BC1 679 CE4 45E E2B 490<br>4B7 E31 5EE C9B 1C5 34E<br><br>F27 899 AB1 451 775 575<br>0D5 D22 14E 38A FC1 CC4<br>EAF 96D A3D 359 FA4 4FB<br>BC6 59A 00C E74 428 F3E<br><br>86A 31B AC5 280 74E 403<br>0B6 21F 540 51A 139 94D<br>31C 5FB 690 EAB A11 D71<br>3D2 F3C FB6 AD5 A2C 740<br><br>57D E83 20B E6E 3F6 6E6<br>ACF E50 291 649 E11 832<br>3F8 846 5CD 4C8 440 FF4<br>D82 771 894 8CC 6A0 4B1 |

| Rate | Input Vector | VCH Test Vector for 2 super frames |
|---|---|---|
| EFR | Data: 1011Hz Tone Pattern<br>Key: ABCD EF01 2345 6789<br>IV: ABCD EF12 3456 7890 | D4DE 9003 7927 EFEB B3EC 8533 61AF 75F0 1130<br>C4B5 2F9B BC98 3325 49F3 C699 1864 8AF2 CC99<br><br>E2AF 2143 A86C 826C 7751 44C7 4FA7 968A B890<br>63B9 D2DD 20F3 A49D BBFB C85B 91E6 D14E 86C1<br><br>E21C 0250 39DC BD88 3F3E 87C6 68BB F026 7A22<br>5DFB ADFC ED6A CE22 5988 74DE 7359 09D6 51CB<br><br>7262 DF90 FB78 2121 7F5E 4912 D2FE 1BC6 76EF<br>8E00 0DC9 36A8 690C DDC0 64DE 3F1F 1C35 B58D<br><br>F310 CD1A E8CB BE89 81AA C18A 5DCC B9C6 CFDA<br>ACC0 EB94 EBB2 60AA E54A 8DD4 165E FA9C 9A2A<br><br>FC45 8421 5E5D 95D0 A42A 5C1C 5407 3E75 5FA3<br>E3C5 4C23 D355 0B0C 749D CC31 EBEE F50E 695E<br><br>A627 5FDF 6CAD 1C33 094C B8FA 3DE0 8ED1 AF11<br>9358 9B0B 76F8 D501 E8F0 3C9F 766C 93E7 F735<br><br>394F 5D4B 6016 0AD9 301A 179E 95D7 84AD F5F6<br>E5EB 53DD 9046 8E72 AA36 339D 1A77 B775 AABC |

#### 7.2.1.3. Voice Channel with AES Encryption

This section presents the test vector for a voice call using AES encryption.

| Rate | Input Vector | VCH Test Vector for 2 super frame |
|---|---|---|
| EHR | Data: 1031Hz Tone Pattern<br>Key: ABCD EF01 2345 6789<br>CDEF 0123 4567 89AB<br>EF01 2345 6789 ABCD<br>0123 4567 89AB CDEF<br>IV: ABCD EF12 3456 7890 | BD4 503 BDC 7F1 87A F31<br>72F FC7 506 DB5 833 0D0<br>5C0 BC6 F14 71D DE5 72B<br>17D 720 2AF 5EA 342 472<br><br>4AE F65 28D A81 45E 9BD<br>916 D09 EC7 0C2 D7E 5FC<br>15A 25D 968 FD1 A7F 14E<br>DB6 E47 165 5BB A93 502<br><br>657 929 676 3DC F3F 8CD<br>4E9 7E1 AB7 7B9 C8E 5C8<br>A13 90C 285 695 DB3 667<br>BBA E0D 4F6 9A2 FC8 AFF<br><br>2D6 070 1DF 171 C73 5B7<br>F40 E8F B1B B3F 9A5 58D<br>748 575 1FE 484 A65 3BB<br>644 E70 5B9 16A 310 BCF<br><br>50D D37 B40 C5E BD3 638<br>6CB 7C1 8B1 79D B8F 2CB<br>492 45F CEB 8D0 507 35F<br>7E2 D9F 89A 6ED 1FA 25F<br><br>E89 4C5 3A4 D7B 5AF BBF<br>2CB 7A7 B4B 827 B6B D42<br>D1F 438 BC5 8B9 304 32B<br>664 A0B 015 5CB E78 BBC<br><br>C68 3C9 BAE 600 2F0 D04<br>E8D 8D9 ACD A07 ADE 722<br>79B 5E8 7D9 E41 418 C13<br>E22 0F3 E27 131 907 C6B<br><br>294 8D6 C1C 2EF A23 0C0<br>73B 3D3 BCF A3E 3BB 0C7<br>00D 78E 911 60F D8F A75<br>0FA 9A6 E8D 516 0D1 F3C |

| Rate | Input Vector | VCH Test Vector for 2 super frame |
|---|---|---|
| EFR | Data: 1011Hz Tone Pattern<br>Key: ABCD EF01 2345 6789<br>CDEF 0123 4567 89AB<br>EF01 2345 6789 ABCD<br>0123 4567 89AB CDEF<br>IV: ABCD EF12 3456 7890 | 14FD 7827 25C8 FBFB 330A 7D3D B987 6604 E57E<br>1F7B 9535 BF9D 3E64 F6A1 9E46 14DA 13FF 1923<br><br>94F8 311F 6ED0 B525 0084 DDBA DB24 2BAE 7A41<br>BEE8 86F3 D285 94C4 0A12 530A D433 B686 E625<br><br>FCCA 171F BFDD C5CE 6E10 A9ED D73B 3D3D F79D<br>9341 840E 8C77 7373 F5F5 8330 94D5 FFF6 34B6<br><br>E102 7876 AEA2 E805 B68F DBCB 5424 FAAA 4FFD<br>67E4 2B2A B0D3 E329 C390 5B71 12DA 5108 0B07<br><br>7E06 7F77 23C2 1156 CD34 F548 CCFC 19E4 3890<br>A2AC 7DCE 00E3 6A61 9D9B 0EFD DEB5 D6A0 81F2<br><br>0778 E80F 7F5B FF18 7C0D A917 D30C EC09 20AB<br>1A72 8783 E40B C4E6 6E20 9F70 8242 2A3F 25DD<br><br>1423 23CC 288B 5F73 88EF 2B14 B406 4A2E 6799<br>A9C0 045A 2EA5 3D16 6C59 8E8C 56C0 9419 83B6<br><br>BAD6 7486 40EC FA86 0633 A27E D371 188D 30D9<br>471B 0A43 AB7E 72F9 8498 D6E4 2717 DCFF 5A8B |

### 7.2.2. User Data Channel Test Vectors

This section provides test vectors of the case that the first DCALL (User Data) message with 20 characters is encrypted. Those test vectors indicate only User Data field of Octets 2 to 21 of DCALL, and a UDCH is constructed by adding Octets 0 to 1 information to the test vector and performing channel coding.

The test vectors encrypting the first SDCALL_REQ (User Data) message with 14 or 8 characters are constructed by extracting data corresponding to the “A-N” or “A-H” string of the test vectors for UDCH.

#### 7.2.2.1. User Data Channel with Scramble Encryption

This section presents the test vector for a data call using scramble encryption. Key information below is written in binary number.

| Input Vector | UDCH Test Vector |
|---|---|
| User Data: “ABCDEFGHIJKLMNOPQRST”<br>Key: 000 0000 0000 0001 | C143 4342 4552 4730 485A 4D2C 580E<br>30D1 5054 5540 |

#### 7.2.2.2. User Data Channel with DES Encryption

This section presents the test vector for a data call using DES encryption.

| Input Vector | UDCH Test Vector |
|---|---|
| User Data: “ABCDEFGHIJKLMNOPQRST”<br>Key: ABCD EF01 2345 6789<br>IV: ABCD EF12 3456 7890 | FE04 A2AB 29D0 CFAE 27FA ABCA B12D<br>F741 898B E218 |

#### 7.2.2.3. User Data Channel with AES Encryption

This section presents the test vector for a data call using AES encryption.

| Input Vector | UDCH Test Vector |
|---|---|
| User Data: “ABCDEFGHIJKLMNOPQRST”<br>Key: ABCD EF01 2345 6789<br>CDEF 0123 4567 89AB<br>EF01 2345 6789 ABCD<br>0123 4567 89AB CDEF<br>IV: ABCD EF12 3456 7890 | 7C06 FE6B F9BD A40E 9ADB F432 E471<br>E71F 1C6B E8BF |

### 7.2.3. Initializtion Vector Generation Value

This section presents the test vector for Initilization Vector Generation.

| | 64-bit IV Generation | 128-bit IV Generation |
|---|---|---|
| Input (seed) | ABCDEF1234567890 | ABCDEF1234567890 |
| 1st Output | ABCDEF1234567890 | ABCDEF1234567890 8B8DDEEB890F4CF1 |
| 2nd Output | 8B8DDEEB890F4CF1 | 8B8DDEEB890F4CF1 C83AB2073A8D80E8 |
| 3rd Output | C83AB2073A8D80E8 | C83AB2073A8D80E8 6C4663AF3D39244B |
| 4th Output | 6C4663AF3D39244B | 6C4663AF3D39244B 00A5754C8AC49E3F |
| 5th Output | 00A5754C8AC49E3F | 00A5754C8AC49E3F 81C55D6EA55913C7 |
| 6th Output | 81C55D6EA55913C7 | 81C55D6EA55913C7 C5522BA39D354F27 |
| 7th Output | C5522BA39D354F27 | C5522BA39D354F27 C8065031EAF69931 |
| 8th Output | C8065031EAF69931 | C8065031EAF69931 03C240EC1877D693 |

## 8. Revision History

| Version | Date | Revised Contents |
|---|---|---|
| 1.0 | Oct 26 2007 | Version 1.0 release |
| 1.1 | Dec 12 2008 | Section 3: Added the item .<br>Section 5.4: Added the description for 9600bps/EHR to Encryption Call Procedure. |
| 1.2 | Jul 7 2008 | Copyright added.<br>Section 5.4.3: Added the description for encryption of short data call. |
| 1.3 | Nov 11 2011 | Section 1; Add the description of Type-C.<br>Section 3: Delete unused abbreviations.<br>Section 5: Add the section for scope of encryption, Modify the number of Key ID in Table 5.4-1.<br>Descriptions of DES and AES are added in the related sections. (Section 2, 3, 4.2, 5.3.2, 5.3.3, 5.4.1, 5.4.3, 5.4.3.2, 5.4.4.2, 5.4.5.2, 5.4.6, 7.2.1.2, 7.2.1.3, 7.2.2.2, 7.2.2.3, 7.2.3) |
| | | |