通信情報保護學會論文誌
第6卷 第4號, 1996. 12

CDMA Digital Mobile Communications and Message Security

Man Young Rhee

Professor Emeritus, Hanyang University
President of Korea Institute of Information Security and Cryptology, Seoul, Korea
Chairman, Board of Directors, Korea Information Security Agency, Republic of Korea
kiiscedc@soback.kornet.nm.kr

Abstract

The mobile station shall convolutionally encode the data transmitted on the reverse traffic channel and the access channel prior to interleaving. Code symbols output from the convolutional encoder are repeated before being interleaved except the 9600 bps data rate. All the symbols are then interleaved, 64-ary orthogonal modulation, direct-sequence spreading, quadrature spreading, baseband filtering and QPSK transmission.

The sync, paging, and forward traffic channel except the pilot channel in the forward CDMA channel are convolutionally encoded, block interleaved, spread with Walsh function at a fixed chip rate of 1.2288 Mcps to provide orthogonal channelization among all code channels. Following the spreading operation, the I and Q impulses are applied to respective baseband filters. After that, these impulses shall be transmitted by QPSK.

Authentication in the CDMA system is the process for confirming the identity of the mobile station by exchanging information between a mobile station and the base station. The authentication scheme is to generate a 18-bit hash code from the 152-bit message length appended with 24-bit or 40-bit padding. Several techniques are proposed for the authentication data computation in this paper. To protect sensitive subscriber information, it shall be required enciphering ceratin fields of selected traffic channel signaling messages. The message encryption can be accomplished in two ways, i.e., external encryption and internal encryption.

1. Introduction

This paper mainly covers topics that can be applied to Code Division Multiple Access (CDMA) which is receiving a great deal of attention as promising technology for future generations of mobile communications system.

For the cellular industry, selection of the most appropriate access method is a challenging task. CDMA is an attractive technique for wireless access to broadband services.

The CDMA concept is explained simply in terms of modulation and multiple access schemes based on spread spectrum communication. Techniques involving spread spectrum(SS) modulation have been evolving over the last 40 years. Spread spectrum techniques were well established for anti-jam and multipath applications as well as for accurate ranging and tracking. It is also proposed to employing SS techniques for CDMA to support simultaneous services for digital communication among a large community users.

In the lase six years, the wireless (or radio) communications field has changed very rapidly. This paper is intended to motivate the reader to further explore this challenging area.

In early 1990, QUALCOMM Incorporated of San

臺灣高雄市 國立中山大學에서 開催된 ICS'96 國際學術大會(1996.12.19-21)에 招請받아 2時間 10分間 主題發表한 論文

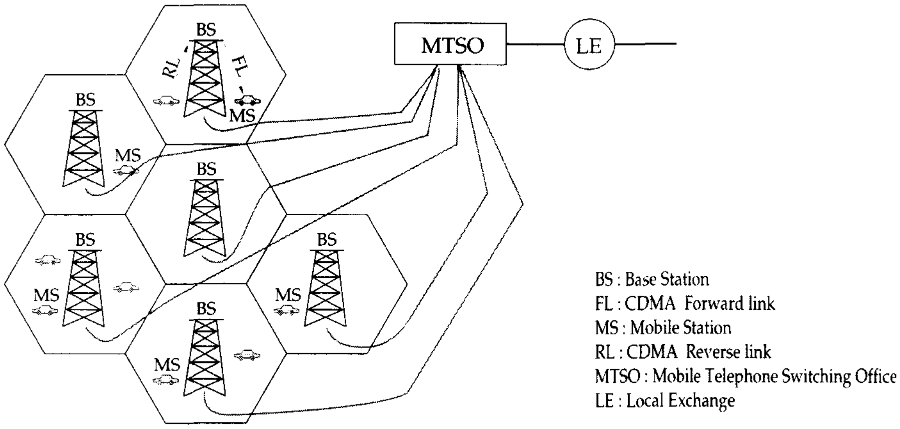


Figure 1 CDMA forward/reverse cellular link geome in hexagonal cell coverage area

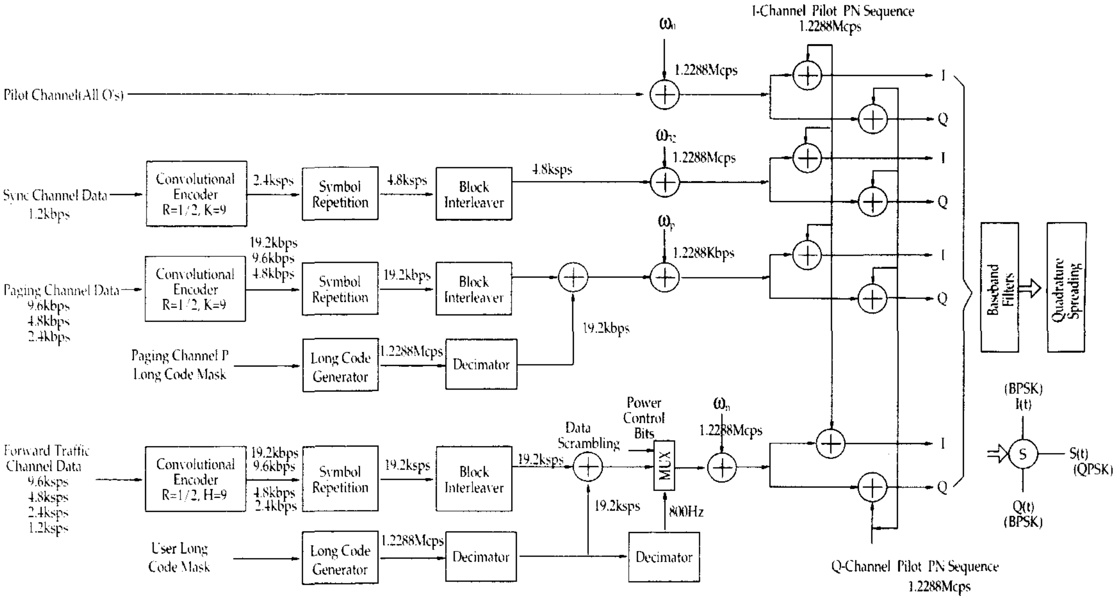


Figure 2 Forward CDMA Code Channel Structure

Diego, California pioneered to introduce the intensive system concepts and the innovative implementation approaches on CDMA spread spectrum digital cellular systems. This CDMA system was standardized and is known as the IS-95 standard of the Telecommunications Industry Association (TIA) and the Electronic Industries Association (TIA/EIA/IS-95).

Typical digital cellular systems can be listed such as GSM (European scheme, 1990), NA-TDMA (North American IS-54 scheme, 1990), PDC (Japanese standard scheme, 1991), and CDMA (US IS-95 scheme, 1993).

The Global System for Mobile Communications (GSM) TDMA system was developed in Western Europe starting in June 1982. GSM offers the capability of extending through diverse telecommunication networks (i.e. ISDN) and compatibility throughout the European continent. In 1992, the first commercial GSM system was devised in Germany. GSM is based on a combination of FDMA and TDMA.

The NA-TDMA system is similar to GSM scheme. The only difference is that there is only one common radio interface in this system. Personal Digital Cellular (PDC) is Japanese TDMA cellular system operating at 800 MHz and 1.5 GHz. This system provides nine interfaces among the digital cellular networks and 1.5 GHz PDC was put in service in 1994.

2. Structual Layout of CDMA Channels

For CDMA cellular systems, the service area is divided into hexagonal cells as shown in Fig. 1. Each cell contains a base station which is connected to the Mobile Telephone Switching Office (MTSO) prior to vocoding. In each cell there are two links consisting of the forward and reverse channels between the base station and each mobile station in the cell. The forward channel designates the forward link from a base station to a mobile station in the cell. The reverse CDMA channel denotes the reverse link from the mobile station to the base station.

CDMA does reuse the cellular ratio frequency excellently but also controls system capacity effectively because CDMA is inherently an excellent anti-interference. The forward CDMA channel is orthogonally spread by the appropriate Walsh function and it then spread by a quadrature pair of pilot PN sequences at a Walsh function and is then spread by quadrature pair of pilot PN sequences at a fixed ship rate of 1.2288 Mcps. All data to be transmitted on the reverse CDMA channel are convolutionally encoded, block interleaved, modulated by the 64-ary orthgonal modulation, and direct sequence spread by the long code prior to transmission. Of course, a mobile station can obtain service by communicating with wither an analog FM base station or with a CDMA base station.

The overall structures of forward/reverse CDMA channels are shown in Figs. 2 and 3, respectively. The forward CDMA channel consists of the pilot channel, sync channel, paging channels, and a number of forward traffic channels. Each of these code channels is orthogonally spread by the appropriate Walsh function and are then spread by a quadrature function and are then spread by a quadrature pair of PN sequences at a fixed chip rate of 1.2288 Mcps. Data rates at the channel input are as follows:

(1) the pilot channel sends all 0's at a 19.2 kbps rate, (2) the sync channel operates at a fixed rate of 1200 bps, (3) the paging channel supports the fixed data rate operation at 9600, 4800, or 2400 bps, and (4) the forward traffic channel supports variable data rate at 9600, 4800, 2400, or 1200 bps.

The sync channel, paging channel, and forward traffic channel are convolutionally encoded for error-correction prior to transmission, but the pilot channel does not use convolutional encoding. For all code channels except the pilot channel, each convolutionally encoded symbolis repeated prior to block interleaving.

For paging and forward traffic channels, repetition depends on the data rate of each channel. For a sync channel, each encoded symbol is repeated once (two of each symbol) and the modulation symbol rate is 4800 sps.

All symbols after repetition on the sync channel, paging channel, and forward traffic channel are block interleaved. The purpose of using block interleaving is to protect from burst errors while sending them through a multipath fading environment. Each of the forward CDMA channel is orthogonally spread

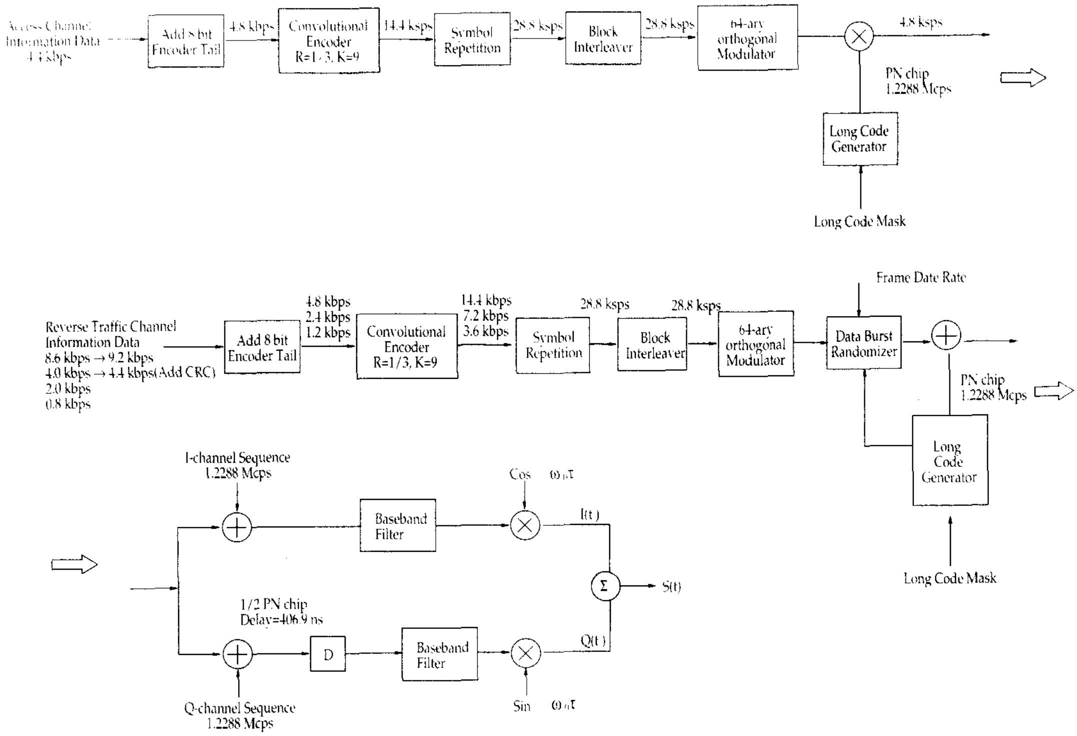
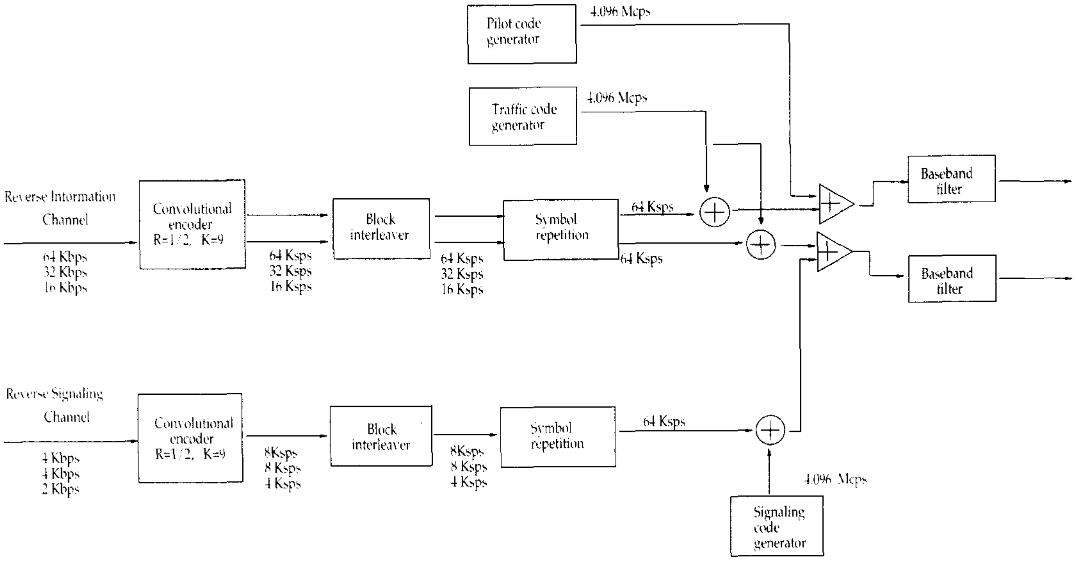


Figure 3 Reverse CDMA code channel structure



Figrue 4 Reverse W-CDMA channel structure for reverse traffic channel

by one of 64 Walsh functions and is then spread by a quadrature pair of pilot PN sequences at a fixed chip rate of 1.2288 Mcps.

The reverse CDMA channel is composed of access channel and reverse traffic channels.

Figure 3 shows the overall structure of the reverse CDMA channel. Data transmitted on the reverse CDMA channel is grouped into 20 ms frames. All data transmitted on the reverse CDMA channel is convolutionally encoded for random-error correction, block interleaved for burst error correction, modulated by the 64-ary Walsh codes, consisting of each 64 chips long, and direct-sequence spread by the long code of period $2^{42}-1$ chips prior to transmission.

The data burst randomizer is not used when the mobile station transmits on the access channel. But in the reverse traffic channel, the data burst randomizer generates a masking pattern of 0s and 1s that randomly masks out the redundant data generated by the code repetition. The reverse traffic channel and the access channel are direct-sequence spread by the long code. This spreading operation involves modulo-2 addition of the output stream from the data burst randomizer and the long code. Following the direct sequence spreading, the reverse traffic channel and access channel are spread in quadrature as shown in Fig. 3.

3. *Characteristics and Functions of CDMA Channel Link*

Since CDMA uses duplexing channels, there exists functional relationships between the base station and the mobile station in a cell coverage. The following summary explains the functions of the base/mobile stations and the characteristics of all the code channels in the CDMA channel. This brief survey includes system aquisition, timing, synchronization, interleaving, orthogonal channelization, spreading techniques, power control, call processing, handoff procedures, authentication, and message privacy.

CDMA Cellular System

Four overhead messages are conveyed between the base station and mobile station in order to meet the required functions in the CDMA cellular system.

System parameter message: contains paging channel, registration parameters, parameter to aid pilot acquisition, etc.

Access parameter message: contains access channel and control parameters. Some of these control parameters provides a dynamic feedback to the mobile station to control its transmit rate, and thus serves to stabilize the access channel.

Neighbor list message: contains information to speed handoff to a neighbor base station, including the time offset of the pilot PN and the basic neighbor configuration.

CDMA channel list message: lists CDMA frequency assignments that contain paging channels. This allows the mobile station to correctly determine where to find its paging channel.

Mobile Station	Channels	Base Station
	Pilot Channel	
○ The mobile station moninitors the pilot channel at all times except when not receiving in the slotted mode. ○ A mobile station uses the pilot channel for synchronization. ○ The mobile station acquires the pilot channel of a CDMA system within 20 ms.	○ The pilot channel is a reference channel which the mobile station uses for acquisition, timing, and as a phase reference for coherent demodulation. ○ A pilot is transmitted at all times by the base station on each active forward CDMA	○ The base station shall continually transmit a pilot channel for every CDMA channel supported by the base station. ○ During pilot and sync channel processing, the base station transmits the pilot channel and sync channel which the mobile

	channel. The pilot channel is an unmodulated spread spectrum signal that is used for synchronization by a mobile station operating within the coverage area of the base station.	station uses to acquire and synchronize to the CDMA system.

Sync Channel

○ Receives sync channel message. ○ Adjusts its timing to normal system timing. ○ Determines and begins monitoring its paging channel. ○ Sync channel is an encoded, interleaved, spread, and modulated spread spectrum signal that is used by mobile stations to acquire initial time synchronization.	○ Sync channel is used for the system acquisition. ○ Once the mobile station acquires the system, it will not normally reuse the sync channel. ○ Since the pilot PN sequences are offset differently for each base station, the flaming of the sync channel is different for every base station. ○ Sync channel transports the synchronization message to the mobile station.	○ Sends sync channel message. ○ Only one message is sent on the sync channel, i.e., sync channel message. ○ Sync channel frame is the length of pilot PN sequence. ○ Transmits the sync channel frame time-aligned with the pilot PN sequence.

Paging Channel

○ Monitors only a single paging channel. ○ Orders (a broad class of messages) are used to control a particular mobile station. ○ Orders are used for everything from acknowledging registration to locking or preventing an errant mobile station from transmitting. ○ The channel assignment message allows the base station to assign a mobile station to the traffic channel, change its paging channel assignment, or direct the mobile station to use the analog FM system. ○ Data scrambling applies to the paging channel by performing the modulo-2 addition of interleaver output with the long code.	○ A paging channel is determined by hashing over all the available paging channels. ○ Data rates: 2.4, 4.8, 9.6 kbps. (paging channels on different CDMA frequency assignment) ○ CDMA frequency assignments listed in CDMA channel list message allows the mobile station to correctly determine where to find its paging channel. ○ Paging channel messages convey information from the base station to the mobile station. ○ For major types of messages: Overhead, paging, order and channel assignment.	○ The base station may also assign a mobile station to a particular paging channel. ○ Page message contains pages to one or more mobile stations. ○ Sends pages when the base station receives a call for the mobile station and pages are usually sent by several different base stations. ○ The paging channel is a encoded, interleaved, spread and modulated spread spectrum signal used for transmission of control information and pages from a base station to a mobile station.

Access Channel

- Responds to a paging channel message by transmitting on one of the associated access channels.
- Chooses randomly both on access channel from the set of available access channels and a PN time alignment from the set of PN time alignments.
- Provides communications to the base station when a traffic channel is not used at the mobile station.
- Distinguished by a different long PN code.

- The access channel is a random access CDMA channel.
- Provides communications from the mobile station to the base station when the mobile station is not using a traffic channel.
- Uses only the special 4800 bps mode in our case.
- Provides for
  - Call originations
  - Responses to pages
  - Orders
  - Registrations
- One or more access channels is paired with every paging channel.
- The access channel transmission rate can be varied for different types of transmissions and for different classes of mobile stations. But one particular data rate should be chosen.

- Responds to transmissions on a particular access channel by a message on an associated paging channel.
- Unless two or more mobile stations choose the same access channel and the same PN time alignment, the base station is able to receive their simultaneous transmission.
- Controls the rate of access channel transmissions to prevent too many simultaneous transmissions by multiple mobile stations.
- Control of access channel transmissions is accomplished through the parameters contained in the access parameters message which is sent on the paging channel.

Traffic Channel

- When the mobile station has been assigned to a traffic channel, signaling (either blank-and-burst or dim-and-burst) occurs directly on the traffic channel.
- Blank-and-burst signaling is sent at 9.6 kbps and replaces one or more frames of primary traffic data, typically vocoded voice, with signaling data, as in the analog FM system.
- Dim-and-burst signaling sends both signaling and primary traffic data in a frame using the 9.6 kbps transmission rate. Dim-and-burst signaling has an immense advantage over blank-and-burst because in voice quality is essentially undetectable.

- Both the forward and reverse traffic channels use a similar control structure consisting of 20 ms frames. Frames can be sent at either 9.6, 4.8, 2.4, or 1.2 kbps.
- There are five types of control messages on the traffic channel :
  - Messages controlling the call itself
  - Messages controlling handoff
  - Messages controlling forward line power
  - Messages for security and authentication
  - Messages eliciting or supplying special information from or to the mobile station.

- During traffic channel processing, the base station uses the forward and reverse traffic channels to communicate with the mobile station while the mobile station is in the mobile station control on the traffic channel state.
- Traffic channel processing consists of the following substates :
  - Traffic channel initialization substate - In this substate, the base station begins transmitting on the forward traffic channel and receiving on the reverse traffic channel.
  - Waiting for order substate- In this substate, the base station sends the alert with

○ The mobile station supports the following handoff procedures:

- In case of commencing communications with a new base station without interrupting communications with the old base station (Soft handoff).
- In case of transitions between disjoint sets of base stations, different frequency assignments, or different frame offset (Hard handoff).

○ The forward traffic channel is used for transmission of user and signaling traffic from the base station to a sspecific mobile station during a call.

○ A reverse traffic channel is used to transport user and single traffic from a signaling mobile station to one or more base stations.

information message to the mobile station.

- Conversation substate - In this substate, the base station exchanges primary traffic bits with the mobile station's primary service option application.
- Release substate - In this substate, the base station disconnects the call.

○ Encryption and Authentication : Authentication is achieved only when the base station posseses identical sets of shared secret data (SSD) with the mobile station.

A 128-bit SSD is stored in the mobile station and it is partitioned into distinct subsets - SSD-A and SSD-B.

- The 64-bit SSD-A is used to support the authentication.
- The 64-bit SSD-B is used for CDMA voice privacy and data confidentiality.

4. Wideband CDMA Channels

Sections 2 and 3 have covered CDMA links based on TIA/EIA/IS-95. Modulation parameters of both the reverse and forward channels aewe shown as listed in Table.1

Table 1 Reverse and Forward Traffic Channel Modulation Parameters

Channel Parameters	Reverse Traffic Channel	Forward Traffic Channel
Data rates (bps)	9600, 4800, 2400, 1200	9600, 4800, 2400, 1200
PN chip rate(Mcps)	1.2288	1.2288
Code symbols rate (sps)	28,800	19,200

On the other hand, JTC (AIR)/95 · 01 · 30 titled "proposed Wideband CDMA PCS standard' was distributed by TAG-7 in 1995. Modulation parameters of both the reverse and forward information channels of respective traffic channel are shown in Table 2.

Table 2 Modulation Parameters for Reverse and Forward Information Channels of Respective Traffic Channel

Channel Parameters	Reverse Information Channel	Forward Information Channel
Data rates (bps)	64000, 32000, 16000	64000, 32000, 16000
PN chip rate(Mcps)	4.096	4.096
symbols rate (sps)	64,000	64000

4.1 Reverse W-CDMA Channel

The reverse W-CDMA channel is the communication link from the personal station to the base station. The reverse W-CDMA channel contains an access channel and a traffic channel. Each channel includes a pilot channel. The reverse traffic channel also includes a signaling channel. A personal station transmits a reverse pilot channel, which is synchronized to the pilot channel from the base station.

The reverse W-CDMA channel is composed of access and reverse traffic channels. The access channel contains of two channels i.e., the reverse pilot channel and reverse access channel. The reverse traffic channel consists of three channels whose types are the reverse pilot channel, reverse information channel, and reverse signaling channel as shown below.

- Reverse W-CDMA Channel
  - Access channel
    - Reverse pilot channel
    - Reverse access channel
  - Reverse traffic channel
    - Reverse pilot channel
    - Reverse information channel
    - Reverse signaling channel

These channels shall share the same CDMA frequency assignment using direct-sequence CDMA techniques. Each channel is identified by a distinct user long code sequence.
All data transmitted on the reverse traffic channel is convolutionally encoded, interleaved, and modulated by direct-sequence spreading prior to transmission. The reverse CDMA channel structure for the reverse traffic channel is shown in Fig. 4.

4.2 Forward W-CDMA Channel

The forward W-CDMA channel consists of the following code channels : (1) one pilot channel, (2) one sync channel, (3) up to eight paging channels, and (4) a number of forward traffic. channels. The forward traffic channel consists of a forward information channel and a forward signaling channel as shown below.

- Forward W-CDMA Channel
  - Pilot channel
  - Sync channel
  - Paging channel
  - Forward traffic channel
    - Forward information channel
    - Forward signaling channel

Each of these code channels is orthogonally spread by the appropriate Walsh code and is then spread by a PN sequence at a fixed chip rate of 4.096 Mcps. The forward traffic channel structure for the forward information channel and forward signaling channel is shown in Fig. 5

- A pilot channel is transmitted at all times by the base station when it is active. The pilot channel is an unmodulated spread spectrum signal that is used for synchronization by a personal station operating within the coverage area of the base station. The pilot channel shall be orthogonal spread with Walsh code index zero prior to transmission.
- The sync channel is convolutionally encoded, interleaved, spread, and modulated to become a spread spectrum signal that is used by personal stations operating within the coverage area of the base station to acquire initial frame synchronization. The bit rate for the sync channel is 16 kbps. The I and Q channel pilot PN sequences for the sync channel use the same pilot PN sequence offset as the pilot channel for a given station.
- The paging channel is encoded, interleaved spread, and modulated to become a spread spectrum signal that is monitored by personal stations operating within the coverage area of the base station. The base station uses multiple paging channels to transmit system information and personal station specific messages.
- The forward traffic channel is used for the transmission of user and signaling information to a specific personal station during a call. The maximum number of forward traffic channels that can be simultaneously supported by a given forward channel is equal to 64 for 64 kbps, 128 for 32 kbps, and 256 for 16 kbps. The number of pilot

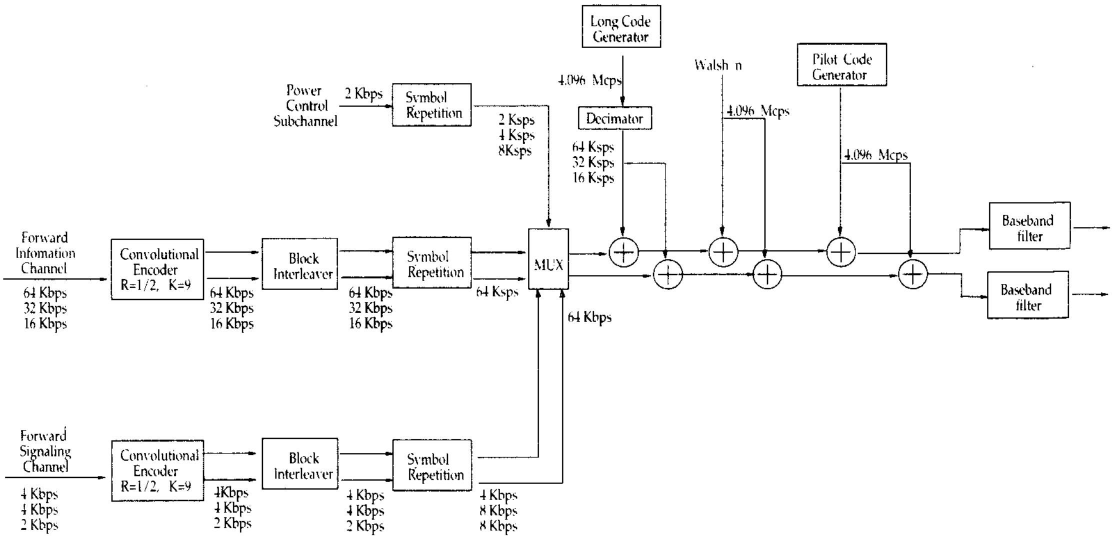


Figure 5 Forward traffic channel structure

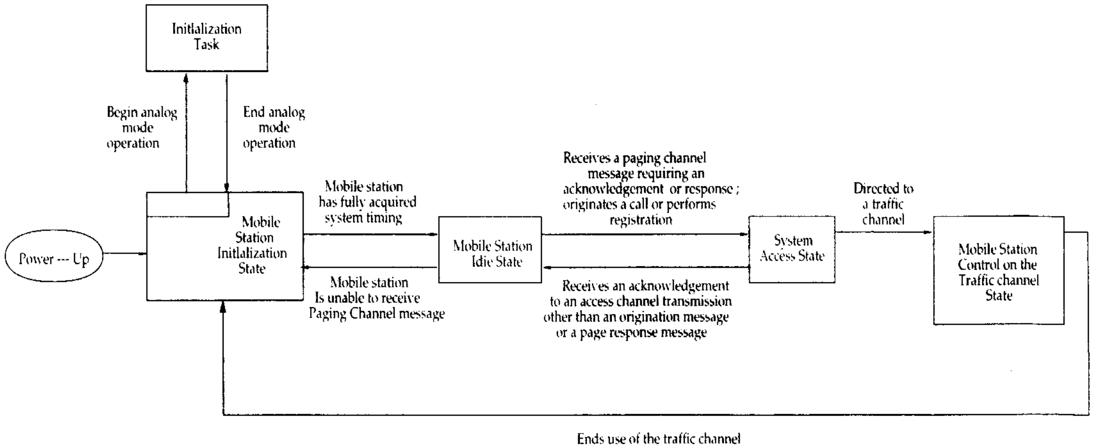

Initialization Task
Begin analog mode operation
End analog mode operation
Power --- Up
Mobile Station Initialization State
Mobile station has fully acquired system timing
Mobile station Is unable to receive Paging Channel message
Mobile Station Idle State
Receives a paging channel message requiring an acknowledgement or response ; originates a call or performs registration
Receives an acknowledgement to an access channel transmission other than an origination message or a page response message
System Access State
Directed to a traffic channel
Mobile Station Control on the Traffic channel State
Ends use of the traffic channel

Figure 6 Mobile station call processing states

channel, paging channel, and sync channel operating on the same forward channel must be subtracted from these numbers.

Forward information channel of the forward traffic channel frames sent at the 64, 32, and 16 kbps data rate shall consist of 320, 160, and 80 bits, respectively. The forward traffic channel consists of a forward information channel and a forward signaling channel. The forward signaling channel of the forward traffic channel shall be convolutionally encoded, block interleaved, orthogonal spread with a Walsh function, quadrature modulated by a pilot PN sequence at a fixed chip rate of 4.096 Mcps, baseband filtered, and finally transmitted by QPSK waveform.

5. Call Processing

Call processing can be seperated into two parts, mobile station call processing and base station call processing. Call processing refers to the technique of technique of message flow protocols between the mobile station and the base station.

5.1 Mobile Station Call Processing

As illustrated in Fig. 6, mobile station call processing consists of the following four states.

① Mobile station initialization state

In this state, the mobile station must :

- o Select which system to use (Analog or CDMA operation).
- o Acquire the pilot channel of the selected CDMA system within 20 ms.
- o Receive and process the sync channel message to obtain system configuration and timing information.
- o Synchronize its long code timing and system timing to those of the CDMA system.

②. Mobile station idle state

In this state, the mobile station monitors messages on the paging channel.

The mobile station can receive messages, receive an incoming call, initiate an originating call, initiate a registration, or initiate a message transmission.

③. System access state

In this state, the mobile station sends messages to the base station on the access channel and receives messages from the base station on the paging channel

The system access state consists of the following substates and the mobile station must :

- o Monitor the paging channel until it has received a current set of configuration messages.
- o Send an origination message to the base station.
- o Send a paging response message to the base station.
- o Send a response to a message received from the base station.
- o Send a registration message to the base station.
- o Send a data burst message to the base station.

The mobile station transmits on the access channel using a random access procedure.

The entire process of sending one message and receiving an acknowledgement for that message is called an access attempt. Each transmission in the access attempt is called an access probe. The mobile station transmits the same message in each access probe in an access attempt. Each access probe consists of an access channel preamble and an access channel message capsule.

④ Mobile station control on the traffic channel state

In this state, the mobile station communicates with the base station using the forward and reverse traffic channels.

This state consists of the following substates and the mobile station must :

- o Verify that it can receive the forward traffic channel and begins transmitting on the reverse traffic channel.
- o Wait for an order on an alert with information message.
- o Wait for the user to answer the call.
- o Its primary service option application exchanges primary traffic packets with the base station.
- o Disconnect the call in this release substate.

5.2 Base station call processing

Base station call processing refers to the method relating to the message flow between the base station and the mobile station.

Base station call processing consists of the following types of processing.

① Pilot and sync channel processing

During pilot and sync channel processing, the

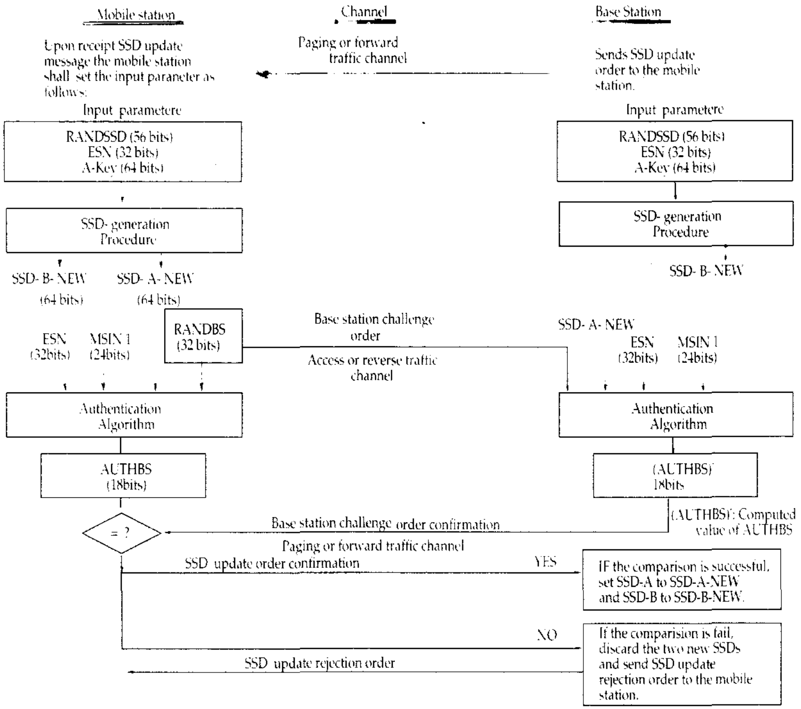

Mobile station
Channel
Base Station
Upon receipt SSD update message the mobile station shall set the input paraneter as follows:
Paging or forward traffic channel
Sends SSD update order to the mobile station.
Input parametere
Input parametere
RANDSSD (56 bits)
ESN (32 bits)
A-Key (64 bits)
RANDSSD (56 bits)
ESN (32 bits)
A-Key (64 bits)
SSD- generation Procedure
SSD- generation Procedure
SSD- B- NEW (64 bits)
SSD- A- NEW (64 bits)
SSD- B- NEW
ESN (32bits)
MSIN 1 (24bits)
RANDBS (32 bits)
Base station challenge order
Access or reverse traffic channel
SSD- A- NEW
ESN (32bits)
MSIN 1 (24bits)
Authentication Algorithm
Authentication Algorithm
AUTHBS (18bits)
(AUTHBS)' 18bits
(AUTHBS)': Computed value of AUTHBS
= ?
Base station challenge order confirmation
Paging or forward traffic channel
SSD update order confirmation
YES
IF the comparison is successful, set SSD-A to SSD-A-NEW and SSD-B to SSD-B-NEW.
NO
If the comparision is fail, discard the two new SSDs and send SSD update rejection order to the mobile station.
SSD update rejection order

Figure 7　SSD update message flow

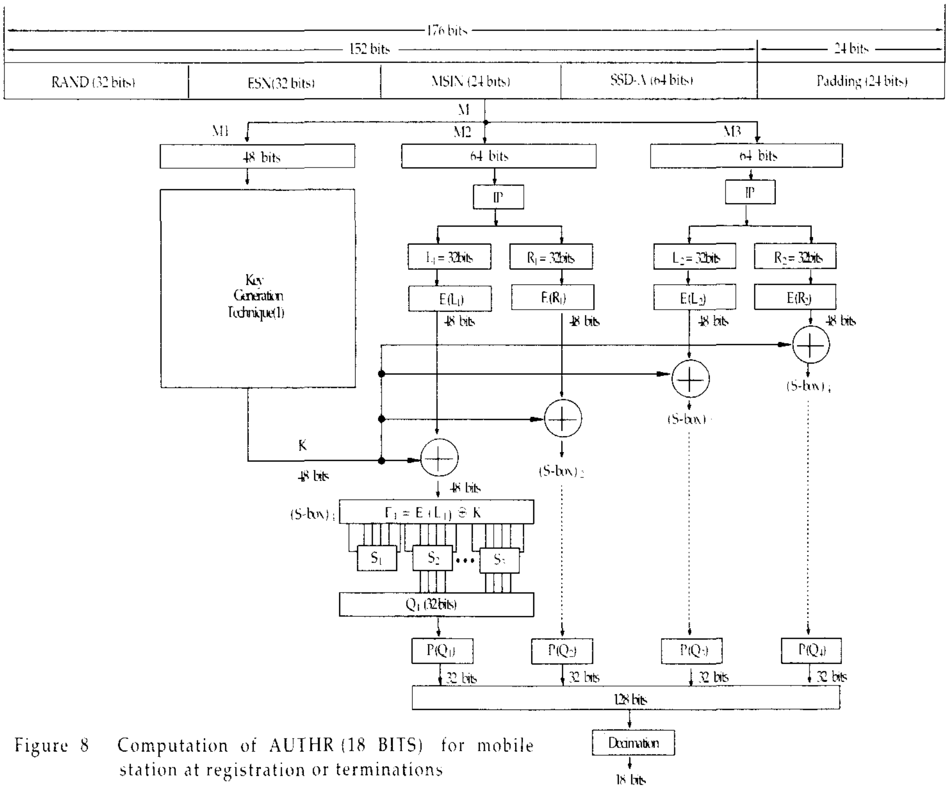


Figure 8　Computation of AUTHR (18 BITS) for mobile station at registration or terminations

base station transmits the pilot channel and sync channel which the mobile station uses to acquire and synchronize to the CDMA system while the mobile station is in the mobile station initialization state.

② Paging channel processing
During paging channel processing, the base station transmits the paging channel which the mobile station monitors to receive messages while the mobile station is in the mobile station idle state and system access state.

③ Access channel processing
During access channel processing, the base station monitors the access channel to receive messages which the mobile station sends while the mobile station is in the system access state.

④ Traffic channel processing
During traffic channel processing, the base station uses the forward and reverse traffic channels to communicate with the mobile station while the mobile station is in the mobile station control on the traffic channel state.

6. Authentication and Message Confidentiality

Authentication refers to the process by which the base station confirms the identity of the mobile station. The successful authentication can be achieved only when the base station possesses identical sets of Shared Secret Data (SSD) with the mobile station. SSD is a 128-bit pattern stored in the mobile station and it is partitioned into two distinct subsets, that is, SSD-A and SSD-B. The 64-bit SSD-A is used to support the authentication and the 64-bit SSD-B is used for CDMA voice privacy and data confidentiality.

The base station is equipped with a database that includes unique mobile station authentication keys (A-keys) and shared secret data (SSD) for each registered mobile station in the cell system. This database is used for authentication of mobile stations. If the base station supports mobile station authentication, the base station may send and receive authentication messages and perform the authentication calculations. SSD is updated using the SSD-generation procedure initialized with the mobile station specific information (ESN), random data (RANDSSD), and the mobile station's A-key. This 64-bit A-key is stored in the mobile station's permanent security and identification memory and is known only to the mobile station and to its associated Home Location Register/Authentication Center (HLR/AC).

The SSD update procedure is described as shown in Fig. 7 and performed as follows:

The base station sends an SSD update message order on either the paging channel or the forward traffic channel. Upon receipt of the SSD update message, the mobile station sets the input parameters (RANDSSD, ESN, A-key) to the SSD-generation algorithm. The mobile station then selects a 32-bit random number (RANDBS), and sends it to the base station in a base station challenge order on the access channel or reverse traffic channel.

Both the mobile station and the base station then set the input parameters (RANDBS, ESN, MIN1, SSD-A-NEW) of the Auth-Signature procedure as illustrated in Fig. 7 and execute the Auth-Signature procedure. AUTHBS is set to the 18-bit result AUTH-SIGNATURE. The base station sends its computed value of AUTHBS to the mobile station in a base station challenge confirmation order on the paging channel or the forward traffic channel. Upon receipt of the base station challenge confirmation order, the mobile station compares the received value of AUTHBS to its internally compared value.

If the comparison is successful, the mobile station executes the SSD-Update procedure to set SSD-A and SSD-B to SSD-A-NEW and SSD-B-NEW, respectively. The mobile station shall then send an SSD update confirmation order to the base station, indicating successful completion of the SSD update.

Upon receipt of the SSD update confirmation order, the base station sets SSD-A and SSD-B to the values computed by the HLR/AC.

In an effort to protect sensitive subscriber information, a method is devised to encrypt certain fields of selected traffic channel signaling messages. If the base station supports mobile station authentication, it may also support message encryption by providing the capability to send encryption control messages and to perform the operations of encryption and decryption.

However, the encryption algorithm is not available in TIA/EIA/IS-95 because the algorithm is governed and regulated under the U.S. International Traffic and Arms Regulation (ITAR) for a description of how the algorithm is initialized and applied. Signaling shall not be encrypted if authentication is not performed. Signaling message encryption is controlled for each call individually.

Voice privacy is provided in the CDMA system by means of the private long code mask used for PN spreading. Voice privacy can be applied on the traffic channels only. All calls are initiated using the public long code mask for PN spreading. The mobile station user may request voice privacy during call setup using the origination message or page response message, and during traffic channel operation using the long code transition request order.

The transition to private long code mask shall not be performed if authentication is not performed. To initiate a transition to the private or public long code mask, either the base station or the mobile station sends a long code transition request order on the traffic channel.

7. *Message Digest Algorithms for Authentication Data*

This chapter shall provide a variety of hash functions which are suitable for security techniques, particularly applicable to the provision of authentication.

A hash function is a one-way function. A one-way hash function is also known a message digest function. Therefore, we may use the terms hash code and message digest interchangeably. A hash function accepts a variable-size message M as input and converts it to a fixed-size message digest H (M) as output. In general, H (M) is much smaller than M ; for example H (M) might be 18, 64, or 128 bits, whereas M might be a megabyte or more.

For message authentication, the message digest function often plays an important role, whence it is useful to understand the properties described in what follows :

① A hash function H is applicable to any data block of a variable-length message M.

② A hash function H produces a fixed-size hash code H(M).

③ H(M) should be computed for any given M for the user's specific security requirements.

④ It must be computationally infeasible to find a message M which has a given prespecified message digest H(M).

⑤ For any given message M, it must be computationally infeasible to find another message M′ even though H(M) = H(M′).

⑥ It is computationally infeasible to find any two distinct messages M and M′, M ≠M′, which map to the same message digests H(M) = H(M′). A hash function satisfying this property is called collision-resistant hash function.

The following specified hash functions are applicable to the provision of authentication.

① MD2 Algorithm

The input to MD2 is a message whose length is an integral number of 8-bit bytes and produces a 128-bit message digest.

The message is padded to be a multiple of the 16-byte checksum which is appended to the end. The message is processed with 16 bytes at a time to produce an intermediate message digest. Each intermediate value of the digest depends on the previous intermediate value and the 16 bytes of the original message being processed. The structure of MD2 is similar to MD4 and MD5, but it is slower and less secure.

② MD4 Algorithm

MD4 was designed, by Rivest in 1990, to be the 32-bit-word oriented scheme. MD2 requires the message to be an integral number of bytes, while MD4 can handle messages with an arbitrary number of bits. Hence, MD4 shall be computed faster on 32-bit machines than a byte-oriented MD2 scheme. MD4 takes an input message of arbitrary length and produces an output 128-bit message digest, in such a way that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest.

The MD4 algorithm is intended for digital signature applications, where a large file must be compressed in a secure manner before being encrypted with a secret key under a public-key cryptosystem.

③ MD5 Algorithm

The MD4 message digest algorithm is a one-way hash function which produces a 128-bit hash. Since the MD4 algorithm was introduced, several researchers attacked the first few rounds of MD4. Rivest subsequently strengthened the algorithm which is called MD5. But MD5 is slightly slower than MD4. MD5 is actually an improved version of MD4, but it also produces a 128-bit hash value for the 512-bit input blocks, divided into sixteen 32-bit sub-blocks. The output of MD5 is a set of four blocks which concatenate to form a single 128-bit message digest.

④ Secure Hash Algorithm(SHA)

SHA like MD5 is a variation of MD4, jointly designed by NIST and NSA. FIPS proposed Secure Hash Standard(SHS) specified as a Secure Hash Algorithm(SHA), which ensures the security of the Digital Signature Algorithm(DSA). However, the SHA is able to be used whenever a secure hash function is required.

⑤ Snefru

Snefru designed by Merkle is a one-way hash function which hashes arbitrary-length messages into either 128-bit or 256-bit values. The message is broken into a 512-m in length. The algorithm hashes a 512-bit value into an m-bit value. A hash function H was designed so that it was easy to compute the hash of an input but computationally infeasible to compute an input that generates a specific hash value. Biham and Shamir demonstrated the insecurity of two-pass Snefru by using differential cryptanalysis.

⑥ Kerberos

Kerberos originally developed at MIT in 1988 for an authentication service is to be able to restrict access to authorized users and to be able to authenticate requests for service.

Kerberos relies exclusively on conventional encryption, making no use of public key encryption. Version 4 and Version 5 of Kerberos are conceptually similar, but Version 4 (1988) is the most widely used version and works only with TCP/IP networks, while Version 5 (1994) corrects some of the security deficiencies of Version 4 and has been issued as a draft Internet Standard (RFC 510). An implementation of Kerberos consists of a Key Distribution Center (KDC) and a library of subroutines.

⑦ X. 509

X. 509 was initially issued in 1988 and is based on the public-key cryptography and digital signature. The digital signature scheme is to require the use of a hash function. X. 509 is an important standard because it is expected that X. 509 directory service will become widely used.

The X. 509 standard does not dictate a specific hash algorithm but recommends RSA. However, the standard was subsequently revised to address some of the security concerns, and a revised recommendation was issued in 1993.

⑧ DM Algorithm

Hash code computation based on DM algorithm was proposed independently by Davies and Meyer in 1985. The message to be hashed is first divided into fixed length blocks : M_1, M_2, $\cdots$, M_t. The message block M_i, $1 \le i \le t$, shall be used as the encryption key. The previous message block is encrypted using that key and then EX-ORed with itself. The result becomes the input to the next round. Thus, at the end, the length of message shall become the last M_i. Let H_i $(1 \le i \le t)$ be defined as $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$, $1 \le i \le t$. Using DES algorithm, the message M is first partitioned into 56-bit DES key blocks. The first block M_1 shall become the 56-bit DES key. The input to the DES algorithm is an initializing vector H_0. The ciphertext output of the DES algorithm is H_1 which is EX-ORed $E_{M_1}(H_0)$ with H_0 and becomes the input to the hash of the second block. The next 56-bit message block M_2 becomes the new DES key. H_1 is the plaintext input to the algorithm and H_2 shall become the ciphertext output. $H_2 = E_{M_2}(H_1) \oplus H_1$ is the input to the hash of the third block, and so on.

Message digest algorithms are much more useful for designing either public key or secret key cryptosystems. Computation of a signature on a long message makes often impractical because of slow speed and low performance. The message can then be compressed into a message digest in a small size. Therefore, instead of computing a signature over a entire long message, a signature over the compressed digest could be used.

8. Authentication Data Computation

In computer-communication networks, it is often necessary for communication parties to verify one another's identity. One practical way is the use of cryptographic authentication protocols employing a one-way hash function. In order to authenticate the identity of the mobile station, the mobile station should have operated in conjunction with the base station. Authentication in the CDMA system is the process for confirming the identity of the mobile station by exchanging information between a mobile station and base station.

One possible authentication scheme may be considered for the case of any block cipher. It can be possible to use a symmetric block cipher algorithm (such as DES) in order to compute the 18-bit hash code. If the block algorithm is secure, then it shall be thought as the one-way hash function is also secure.

The 152-bit message block M is inputted into the authentication algorithm device. Using DES, M is first broken down into the 64-bit blocks as such M = M_1 M_2 M_3 · · ·. The first message block M_1 shall become the DES key. Division into 64-bit blocks can be accomplished by mapping the 152-bit message value onto the 192-bit value by padding with the 40-bit zeros. Appropriate padding shall be needed for devising the message to conveniently

divide into certain fixed lengths. Our authentication scheme is to generate a 18-bit authentication data from a message length of 192 bits. Several techniques shall be proposed for the computation of authentication data (AUTHR) in what follows.

8.1 AUTHR (18 bits) Computation I

Suppose M_1, M_2, and M_3 are the decomposition of a 176-bit value into 48 bits, 64 bits, and 64 bits, respectively. The 176-bit value is composed of 152-bit message length and 24-bit padding. M_1=48 bits shall be used as input to the key generation scheme, (i.e., foggy block in Fig. 8..

Let us arrange the 48-bit input into a 6×8 array, as shown below :

Table 3 A 6×8 Array for Key Computation

Input (column by column)

⇩

1	7	13	19	25	31	37	43
2	8	14	20	26	32	38	44
3	9	15	21	27	33	39	45
4	10	16	22	28	34	40	46
5	11	17	23	29	35	41	47
6	12	18	24	30	36	42	48

Column-wise permutation

7	31	1	43	13	37	25	19
8	32	2	44	14	38	26	20
9	33	3	45	15	39	27	21
10	34	4	46	16	40	28	22
11	35	5	47	17	41	29	23
12	36	6	48	18	42	30	24

Row-wise permutation

11	35	5	47	17	41	29	23
7	31	1	43	13	37	25	19
9	33	3	45	15	39	27	21
12	36	6	48	18	42	30	24
8	32	2	44	14	38	26	20
10	34	4	46	16	40	28	22

⇨ Output (row by row)

A 48-bit key generation from M_1 is computed as follows :

11	35	5	47	17	41	29	23	7	31	1	43	13	37	25	19
9	33	3	45	15	39	27	21	12	36	6	48	18	42	30	24
8	32	2	44	14	38	26	20	10	34	4	46	16	40	28	22

Example 1 Assume that the key input be 16c27a415f39 (hexadecimal) = 0001 0110 1100 0010 0111 1010 0100 0001 0101 1111 0011 1001 (binary).

Using Table 3, the output data can be computed as 01959d7857e4 (hexadecimal).

The 48-bit key output shall be obtained in binary notation as follows :
0000 0001 1001 0101 1001 1101 0111 1000 0101 0111 1110 0100 (binary) = 01959d7857e4 (hexadecimal)

This 48-bit key K was computed from a fitting to the foggy block in Fig. 8.

Example 2 Now, using Fig. 8, the 18-bit message digest for AUTHR shall be derived by hashing the 176-bit padded message. Assume that the M_2 data block (64 bits) is 17b439a12f51c5a8. This M_2 data is first subjected to an initial permutation (IP) to make it split into two blocks L_1 (left) and R_1(right) where each of them consists of 32 bits as indicated in Table 4.

Table 4 Initial Permutation (IP)

L_1	58	50	42	34	26	18	10	2
	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8
R_1	57	49	41	33	25	17	9	1
	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

When M_2 hexadecimal data convert into the binary values first and arrange them according to Table 4, the initial permuted sequence can be found as follows :

L_1 : 0110 0000 0010 0111 0101 0011 0111 1101
(32 bits)
6 0 2 7 5 3 7 d

R_1 : 1100 1010 1001 1110 1001 0100 0001 0001
(32 bits)
c a 9 e 9 4 1 1

Thus, the initial permuted sequence shall express as

6027537d (L_1) ; ca9e9411 (R_1)

These L_1 and R_1 are expanded from 32 bits to 48 bits, respectively, according to Table 5.

Table 5 Bit Expansion Table

32	1	2	3	4	5
4	5	6	7	8	9
5	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

$E(L_1)$ denotes a function which takes a block of 32 bit as its input and yields a block of 48 bits as its output as shown below.

$E(L_1)$ = b0010eaa6bfa

Once $E(L_1)$ is computed, it is added bit by bit to the key K such that

$\Gamma_1 = E(L_1) \oplus K$
= (b0010eaa6bfa)$\oplus$(01959d7857e4)
= (b19493d23c1e)

This 48-bit input Γ_1 to the (S-box)$_1$ shall be passed through a nonlinear substitution to form the 32-bit output. The 48-bit vector Γ_1 is used as argument in the substitution operation of (S-box)$_1$ from S_1 through S_8. Each S_i, $1 \le i \le 8$, is a matrix of four rows and 16 columns as shown Table 8.8. The input to each S_i is composed of 6 bits. It's first and last bits represent the row number of S_i and the middle 4 bits produce a column number. For example, for an input (010011) to S_i, denoted as $S_i^{01}(1001)$, the row number is 01, i.e., row 1, and the column number is determined by 1001, i.e., column 9.

Table 8.8 Primitive S-box Functions

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3.	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7															
4	11	2	14	15	0	8	13	2	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

If the S-box input, Γ_1 = b19493d23c1e, is expressed in the binary notation, it gives
1011000110010100100100111101001000111
10000011110

Grouping this 48-bit Γ_1 into sets of 6 bits leads to easy computation of substitution operations from S_1 through S_8 as follows :

$$S_1^{10}(0110) = S_1^2(6) = 2 = 0010$$
$$S_2^{01}(1100) = S_2^1(12) = 6 = 0110$$
$$S_3^{00}(1001) = S_3^0(9) = d = 1101$$
$$S_4^{01}(1001) = S_4^1(9) = 7 = 0111$$
$$S_5^{10}(1010) = S_5^2(10) = c = 1100$$
$$S_6^{11}(0001) = S_6^3(1) = 3 = 0011$$
$$S_7^{10}(1000) = S_7^2(8) = a = 1010$$
$$S_8^{00}(1111) = S_8^0(15) = 7 = 0111$$

Concatenating all of these 4-bit S_i, $1 \le i \le 8$, shall yield the 32-bit Ω_1 :

Ω_1 = 0010 0110 1101 0111 1100 0011 1010 0111
= 26d7c3a7

This is the output of $(\text{S-box})_1$.

Using Table 7, the 32-bit permuted output $P(\Omega_1)$ shall be produced by permuting the bits of Ω_1 by taking the 16th bit of Ω_1 as the first bit of $P(\Omega_1)$, the seventh bit of Ω_1 as the second bit of $P(\Omega_1)$, and so on until the 25th bit of Ω_1 is taken as the 32nd bit of $P(\Omega_1)$.

Table 7 Permutation Function P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Thus the result of $P(\Omega_1)$ shall be represented as

$P(\Omega_1)$ = 1100 0101 0110 0111 0011 1111 0011 0001
= c 5 6 7 3 f 3 1

Consider, next, R_1 = 32 bits of M_2. If R_1 expands to 48 bits by using Table 5, we have

$E(R_1)$ = e 5 5 4 f d 4 a 8 0 a 3

EX-ORing $E(R_1)$ with K, the 48-bit input Γ_2 to $(\text{S-box})_2$ shall be computed as

$\Gamma_2 = E(R_1) \oplus K$
= e4c16032d747
= 1110 0100 1100 0001 0110 0000 0011 0010 1101 0111 0100 0111

Division of Γ_2 into 8 sets of 6 bits shall yield $(\text{S-box})_2$ as shown below :

I	Input to S_i, Γ_2	S_i, $1 \le i \le 8$		Output of S_i Ω_2
		Row	Column	
1	111001	3	12	a
2	001100	0	6	3
3	000101	1	2	0
4	100000	2	0	a
5	001100	0	6	b
6	101101	3	6	f
7	011101	1	14	8
8	000111	1	3	8

Thus, $(\text{S-box})_2$ output shall be

Ω_2 = a 3 0 a b f 8 8

Using Table 7, the data sequence by permutation shall become

$P(\Omega_2)$ = 7 9 c 0 6 2 c 9

Thus far, two permutation data $P(\Omega_1)$ and $P(\Omega_2)$ corresponding to the 64-bit M_2 block have been completely computed.

Suppose the third data block M_3 (64 bits) is 51cb36af43000000. Using Table 4, the data sequence by initial permutation of M_3 shall become 13050c1ba0c0a1e, where L_2 = 13050c1b and R_2 = a0c0a1e. Either L_2 (left half) or R_2(right half) of M_3 shall be expanded, respectively, from 32 bits to 48 bits according to Table 8.7 as shown below :

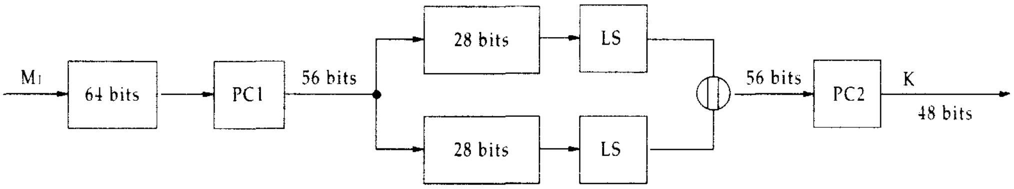
$E(L_2)$ = 8a680a, $E(R_2)$ = 8580f6

As you see, the 32-bit L_2 or R_2 can be spread out and scrambled into 48 bits with the E-table for expansion.

The expanded data $E(L_2)$ or $E(R_2)$ shall be EX-ORing with the key data K such that

$\Gamma_3 = E(L_2) \oplus K$
= (8a680a8580f6) $\oplus$ (01959d7857e4)
= (8bfd97fdd712)
= (1000 1011 1111 1101 1001 0111
1111 1101 1101 0111 0001 0010)

This 48-bit Γ_3 is the input data to $(\text{S-box})_3$. The $(\text{S-box})_3$ operation from S_1 through S_8 shall be executed as follows :



PC1:Permuted choice 1
PC2:Permuted choice 2
⦶ :Concatenation
LS: Left shift once

Figure 9 Key generation based on DES key schedule

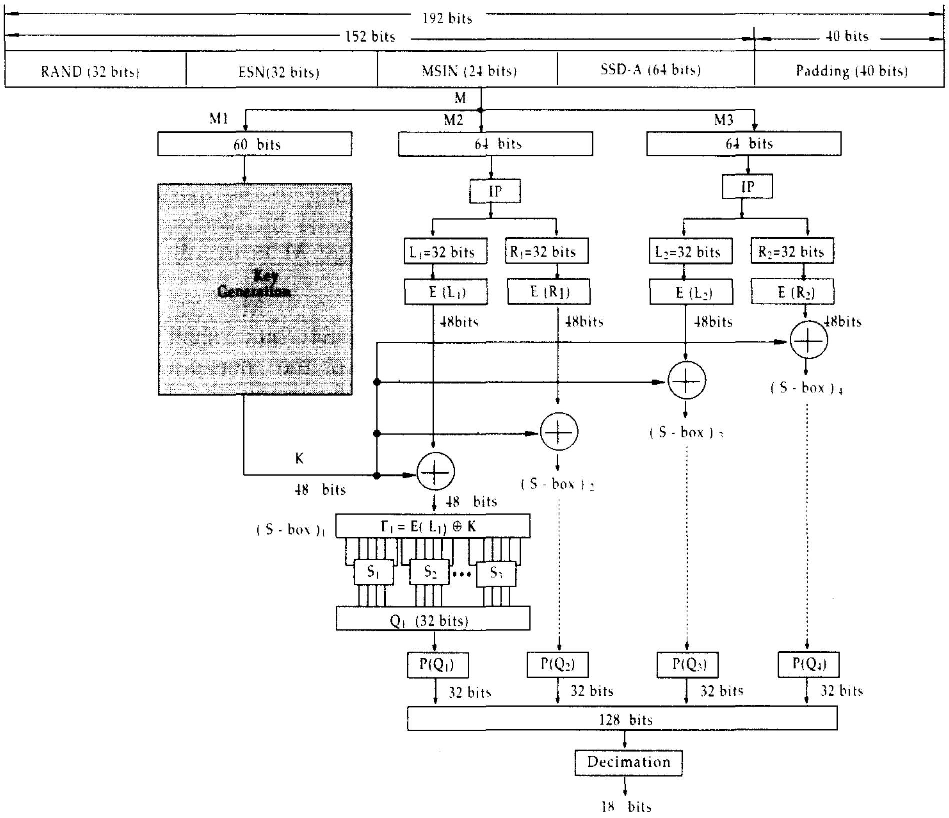


Figure 10 Computation of authentications of mobile station at registrations, unique challenge - response originations, and terminations.
Authentication input parameters:

- Registration and terminations are identical.
- Unique challenge : RAND (32 bits) = RANDU (24 bits) + MIN2 (8 LSBS)
  Originations : Auth - Data (24 bits) = Digits (Last 6 digits transmitted by the mobile station)

Input to S_i,	S_i,		Output from S_i
	Row	Column	
S_1^{10}	2	1	1
S_2^{11}	3	15	9
S_3^{10}	2	11	12
S_4^{01}	1	11	12
S_5^{11}	3	15	3
S_6^{01}	1	14	3
S_7^{00}	0	14	6
S_8^{00}	0	9	9

The output data from (S-box)$_3$ shall be written as Ω_3 = 1 9 c c 3 3 6 9

The permutation function P(Ω_3) yields a 32-bit output from a 32-bit input by permuting the bits of Ω_3 according to Table 7.

Hence P(Ω_3) resulting from permutation shall be

P(Ω_3) = 2 8 3 9 7 d c 2

Finally, consider the right half R_2 = 0a0c0a1e which was resulted from the initial permutation (Table 8.6) of the 64-bit M_3. The 48-bit expanded data, E(R_2), from the 32-bit R_2 is

E(R_2) = 5 4 0 5 8 5 4 0 f c

The (S-box)$_4$ input data Γ_4 shall be computed from EX-ORing E(R_2) with K such that

Γ_4 = E(R_2) $\oplus$ K
= 0 4 d 5 c 5 7 d 1 7 1 8

where K is the key data sequence.

The (S-box)$_4$ substitution operation from S_1 through S_8 shall be executed as shown in the following table.

Input to S_i,	S_i,		Output from S_i
	Row	Column	
S_1^{10}	1	0	0
S_2^{01}	1	6	8
S_3^{01}	1	11	14
S_4^{01}	1	2	11
S_5^{01}	1	15	6
S_6^{01}	1	8	6
S_7^{00}	0	14	6
S_8^{00}	0	12	5

Thus, the output data Ω_4 from (S-box)$_4$ shall be computed as

Ω_4 = 0 8 e b 6 6 6 5

Using Table 7, the permutation of Ω_4 shall be shown as

P(Ω_4) = 8 0 7 d 0 d e c

Thus, we have completely computed all those four permutations, i.e., P(Ω_1), P(Ω_2), P(Ω_3), and P(Ω_4). The 128-bit final output data P(Ω) can be obtained from concatenation process such that

P(Ω) = P(Ω_1) || P(Ω_2) || P(Ω_3) || P(Ω_4).
P(Ω) = (c5673f31, 79e062c9, 28397dc2,
807d0dec)
= (1100 0101 0110 0111 0011 1111 0011 0001
0111 1001 1110 0000 0111 0010 1100 1001
0010 1000 0011 1001 0111 1101 1100 0010
1000 0000 0111 1101 0000 1101 1110 1100)

Finally, the 18-bit authentication data (AUTHR) shall be computed by taking the decimation process from the 128-bit P(Ω) by picking 1 bit per every 7 bits :

AUTHR : 011111000011000101

8.2 AUTHR(18 bits) Computation II

The 152-bit message length is expanded into the 192-bit padded message value by adding a 40-bit padding to the message in order to divide it into three 64-bit blocks, M_i = 64 bits, $1 \leq i \leq 3$. The message M is evenly partitioned into 64-bit blocks

M_1, M_2, M_3

where M_1=64 bits shall be hashed for the generation of the encryption key of 48 bits.

Figure 9 illustrates the key generation scheme applicable to the foggy block in Fig. 10.

Utilizing Fig. 9 for the DES key computation, we should provide Tables 8., 9, and 10 for the permuted choices 1 and 2 and for the number of left shifts.

Table 8. Permuted Choice 1 (PC-1)

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Table 9 Permuted Choice 2 (PC-2)

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	51	51	45	33	48
44	49	39	56	34	53	46	50	50	36	29	32

Table 8.12 Number of Left Shift

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of Left Shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Example 3 Let the DES input key be 7a138b2524af17c3 Using Table 8, the PC-1 output data shall become

L_0 = a481394 (left half)
R_0 = e778253 (right half)

From Table 10, shifting one place to the left (LS) shall yield LS data as

4902729 cef04a˙7

These two LS data shall be concatenated and transformed them into the 48-bit key data according to Table 9. Then the 48-bit DES key shall be

K = 058c4517a7a2

Example 4 Given that M_2=17b439a12f51c5a8 Using Table 4, an initial permutation(IP) shall split into two blocks L_1(left) and R_1(right) where each of them consists of 32 bits. Hence the initial permuted data shall be

IP(L_1) = 6027537d IP(R_1) = ca9e9411

(1) Using Table 5, the data expansion E(L_1) shall become

E(L_1) = b0010eaa6bfa

EX-ORing E(L_1) with K, we have

Γ_1 = E(L_1)⊕K
= (b0010eaa6bfa) ⊕ (058c4517a7a2)
= b58a4bbacc58

This is the input data to (S-box)$_1$, as shown in Fig. 10.

(S-box)$_1$ Operation :

Input	S_1^{11}	S_2^{00}	S_3^{11}	S_4^{01}	S_5^{11}	S_6^{00}	S_7^{11}	S_8^{00}
Row	3	0	3	1	3	0	3	0
Column	6	12	10	5	7	14	8	12
Output	1	12	14	15	13	5	9	5

Thus, (S-box)$_1$ output shall be Ω_1 = 1cefd595. Using Table 7, the permutation function P(Ω_1) shall become

P(Ω_1) = a34d397f

(2) Using Table 5, the data expansion of R_1 yields

E(R_1) = e554fd4a80a3

EX-ORing E(R_1) with K shall become the input Γ_2 to (S-box)$_2$:

Γ_2 = E(R_1)⊕K
= (e554fd4a80a3) ⊕ (058c4517a7a2)
= e0d8b85d2701

(S-box)$_2$ Operation :

Input	S_1^{10}	S_2^{01}	S_3^{10}	S_4^{10}	S_5^{01}	S_6^{00}	S_7^{00}	S_8^{01}
Row	2	1	2	2	1	0	0	1
Column	12	6	1	12	11	9	14	0
Output	3	8	6	5	10	13	6	1

The output data of (S-box)$_2$ shall be Ω_2 = 3865ad61.

Using Table 7, the permutation function P(Ω_2) shall become

P(Ω_2) = 91193e8e

(3) Given that M_3 = 51cb360000000000. Using Table 4, an initial permutation(IP) of M_3 can be obtained as follows :

Left block L_2 = 03050403
right block R_2 = 02040206

These two blocks shall be expanded into 48 bits each according to Table 5.

E(L_2) = 80680a808006

EX-ORing E(L_2) with K, we shall yield the input data to (S-box)$_3$:

Γ_2 = E(L_2)⊕K
= (80680a808006) ⊕ (058c4517a7a2)
= 85c44f9727a4

(S-box)$_3$ Operation :

Input	S_1^{11}	S_2^{00}	S_3^{01}	S_4^{01}	S_5^{11}	S_{61}^{10}	S_7^{00}	S_8^{10}
Row	3	0	1	1	3	2	0	2
Column	0	15	8	7	2	9	15	2
Output	15	10	2	3	12	0	1	4

Thus, the output data from (S-box)$_3$ can be obtained as

Ω_3 = fa23c914

The permutation function P(Ω_3) shall be computed according to Table 7 :

P(Ω_3) = c3cc8226

(4) Next, consider the case for R_2 = 02040206 The bit expansion of R_2 using Table 5 shall become

E(R_2) = 00400800400c

The input Γ_4 to (S-box)$_4$ shall be computed as follows :

Γ_4 = E(R_2) ⊕ K_1
= (00400800400c) ⊕ (058c4517a7a2)
= 05cc4d17e7ae

(S-box)$_4$ Operation :

Input	S_1^{01}	S_2^{00}	S_3^{11}	S_4^{01}	S_5^{01}	S_6^{10}	S_7^{00}	S_8^{10}
Row	1	0	3	1	1	2	0	2
Column	0	14	8	6	2	15	15	7
Output	0	5	4	0	2	6	1	2

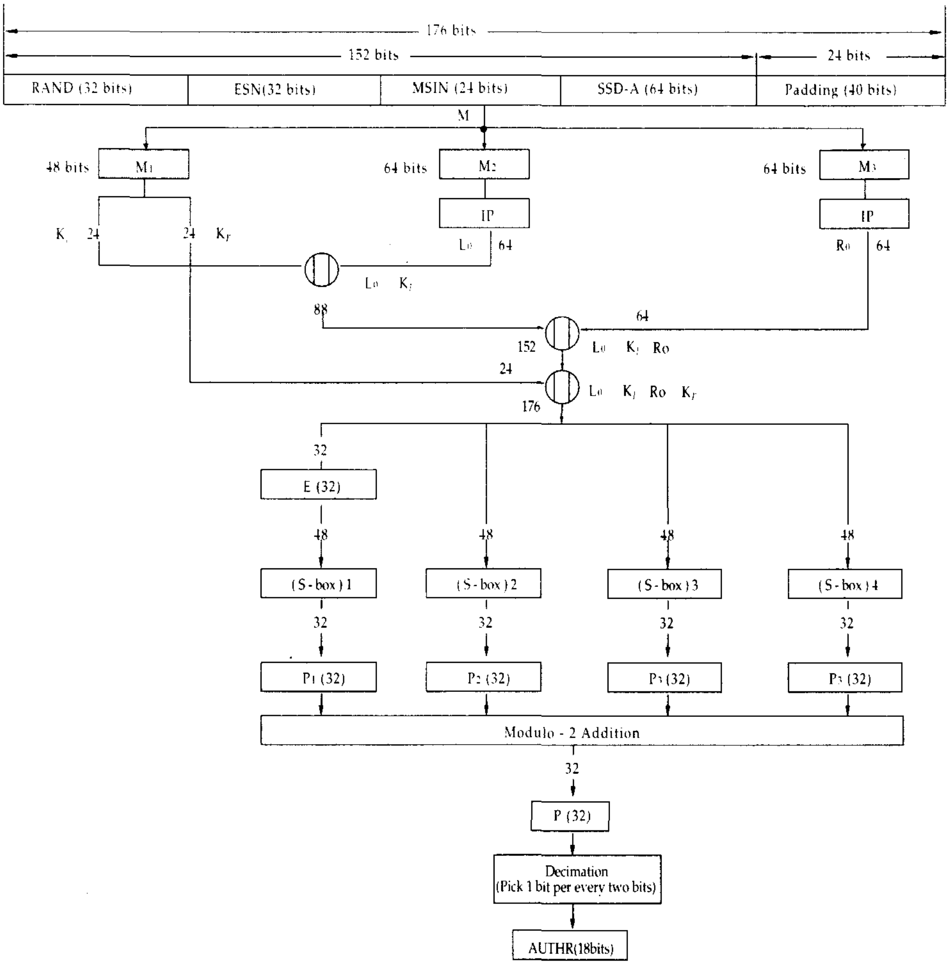


Figure 11 Computation of AUTHR for authentication of mobile station's registrations or terminations.

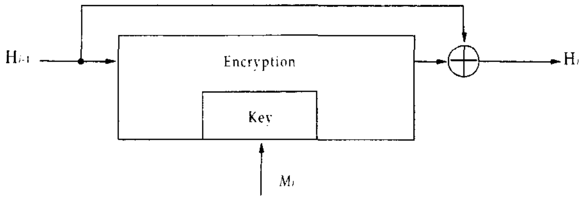


Figure 12 Davies-Meyer scheme

Hence, the output data from (S-box)$_4$ shall be

Ω_4 = 05402612

Using Table 7, the permutation of Ω_4 becomes

P(Ω_4) = 02234098

Thus, the output of S-box is a set of four blocks which concatenate to form a single 18-bit hash. Final data output from S-box is

P(Ω) = P(Ω_1)||P(Ω_2)||P(Ω_3)||P(Ω_4)

= a34d397f 91193e8e c3cc8226 02234098

Finally, the 18-bit authentication data (AUTHR) shall be computed by making use of decimation process from the 128-bit P(Ω) by picking 1 bit per every 7 bits :

AUTHR = 111100001010100100

8.3 AUTHR (18 bits) Concatenation III

Figure 11 illustrates another authentication data computation. The computation procedure for authentication data is described as shown below:

1. Enlarge the 152-bit message into 176 bits by adding the 24-bit padding.
2. Divide this enlarged message of 176 bits into three blocks : M_1 = 48 bits, M_2 = 64 bits, and M_3 = 64 bits. Use M_1 as the 48-bit key.
3. Divide the 48-bit key into two halves : K_l = 24 bits and K_r = 24 bits.
4. Divide the 128-bit block into two halves : L_0 = 64 bits and R_0 = 64 bits.
5. Concatenate K_l with L_0 to produce 88 bits : $L_0||K_l$ → 88 bits
6. Concatenate the result of $L_0||K_l$ with R_0 to produce $L_0||K_l||R_0$ → 152 bits
7. Concatenate the 152-bit result at step 6 with K_r such that

   $L_0||K_l||R_0||K_r$ → 176 bits
8. Divide this 176-bit message into four parts : N_1=32 bits, N_2=48 bits, N_3=48 bits, N_4=48 bits
9. Expand N_1 = 32 bits into E(N_1) = 48 bits using Table 5.
10. Thus, it can be accomplished into four evenly partitioned 48 bits each.
11. These four 48-bit blocks are subject to input to four S-boxes.
12. The sum of these four S-box outputs are modulo-2 additioned, and shall be resulted in 32 bits.
13. Finally, these 32 bits are decimated into AUTHR (18 bits).

Example 5 Referring to Fig. 11 the 176-bit padded message M is divided into three blocks, i.e., M_1=48 bits, M_2=64 bits, and M_3= 64 bits where M_1 shall be used as the key such that K_l = 16c27a (24 bits) and K_r = 415f39 (24 bits). Assume that

M_2 = 17b439a12f51c5a8 (64 bits)

M_3 = 51cb36af43000000 (64 bits)

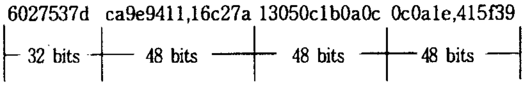
Using Table 4, the initial permutations of M_2 and M_3 shall become, respectively,

IP(M_2) = 6027537dcage9411 = L_0

IP(M_3) = 13050c1b0a0c0a1e = R_0

Data concatenation : $L_0||K_l||R_0||K_r$ (176 bits)

Divide the concatenation data into 4 blocks, i.e., 32 bits and three blocks of 48 bits each as follow :



(1) Using Table 5, the 32-bit block is expanded into the 48-bit block such that

E(6027537d) = b0010eaa6bfa (48 bits)

This is the input data Γ_1 to (S-box)$_1$ from S_1 through S_8 as shown below.

Input to S_i	S_i		Output from S_i
(Γ_1)	Row	Column	(Ω_1)
S_1^{10}	2	6	2
S_2^{00}	0	0	15
S_3^{00}	0	2	9
S_4^{00}	0	7	10
S_5^{10}	2	5	13
S_6^{10}	2	3	5
S_7^{11}	3	7	7
S_8^{10}	2	13	3

Thus, the output data from (S-box)$_1$ shall be

Ω_1 = 2f9ad573.

Using Table 7, the permutation function P(Ω_1) becomes

P(Ω_1) = 675e6f58

(2) Input data to (S-box)$_2$ is

Γ_2 = ca9e941116c2.

In binary notation,

Γ_2 = 110010101001111010010100000
100010001011011000010

Partitioning Γ_2 into every 6 bits and feeding them into S_1 through S_8, the output data from (S-box)$_2$ shall be computed as shown below :

Input to S_i	S_i		Output from S_i
(Γ_1)	Row	Column	(Ω_1)
S_1^{10}	2	6	2
S_2^{00}	0	0	15
S_3^{00}	0	2	9
S_4^{00}	0	7	10
S_5^{10}	2	5	13
S_6^{10}	2	3	5
S_7^{11}	3	7	7
S_8^{10}	2	13	3

Thus, the output Ω_2 of $(\text{S-box})_2$ shall become

Ω_2 = c3a846f2

Using Table 7, the permutation of Ω_2 yields

$P(\Omega_2)$ = 42b6c54d

(3) Input data to $(\text{S-box})_3$ is

Γ_3 = 7a13050c1b0a.

Feeding every 6 bits of Γ_3 into $(\text{S-box})_3$, the output data from $(\text{S-box})_3$ shall be found as

Ω_3 = 7dfbba7f

The permutation of Ω_3 shall yield

$P(\Omega_3)$ = bf7bcff6

(4) Input data to $(\text{S-box})_4$ is

Γ_4 = 0c0a1e415ff39

Express Γ_4 in binary notation first and break down Γ_4 into 8 blocks. Each block shall contain 6 bits, which shall become the input to S_i, $1 \le i \le 8$. Inputting each 6 bits to $(\text{S-box})_4$ shall be found the output from $(\text{S-box})_4$ as follows :

Ω_4 = ff858d93

and $P(\Omega_4)$ = d38afb1b

Thus, all output data from $(\text{S-box})_i$ $1 \le i \le 4$, becomes

675e6f58 42b6c54d bf7bcff6 d38afb1b

Modulo-2 addition of these four groups shall produce

49199ef8

and P(49199ef8) = bf38c449.

Using Table 5, this permuted data shall be expanded into 48 bits shown below :

dfe9f1608253

Converting these hexadecimal values to the binary values first and deleting the left 6 bits as well as the right 6 bits, the following 36-bit binary sequence shall be remained as

111110100111110001011000001000001001

If the decimation rule is applied to this sequence, the 18-bit message digest (AUTHR) can be obtained by picking 1 bit every two bits as shown below :

AUTHR = (110011101100000001)

8.4 *AUTHR (18 bits) Computation Using DM Scheme*

AUTHR computation based on the DM scheme shall be described using DES block cipher for obtaining a hash code. The DM scheme was proposed independently by Davies and Meyer in 1985.

The message to be hashed is first divided into fixed length blocks : $M_1, M_2, \cdots, M_t$. The message block M_i, $1 \le i \le t$, shall be used as the encryption key, as illustrated in Fig. 12. The previous message block is encrypted using that key and then EX-ORed with itself. The result becomes the input to the next round. Thus, at the end, the length of message shall become the last M_i. Let H_i $(1 \le i \le t)$ be defined as $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$, $1 \le i \le t$.

Using DES algorithm, the message M is first partitioned into 56-bit DES key blocks. The first block M_1 shall become the 56-bit DES key. The input to the DES algorithm is an initializing vector H_0. The ciphertext output of the DES algorithm is H_1 which is EX-ORed $E_{M_1}(H_0)$ with H_0 and becomes the input to the hash of the second block. The next 56-bit message block M_2 becomes the new DES key. H_1 is the plaintext input to the algorithm and H_2 shall become the ciphertext output. $H_2 = E_{M_2}(H_1) \oplus H_1$ is the input to the hash of the third block, and so on.

In the following, we attempt to design the authentication procedure to compute the 18-bit hash value, based on the DM scheme. The proposed authentication system is shown as illustrated in Fig. 12.

The message to be hashed is first partitioned evenly into a series of m-bit blocks.

Appropriate padding shall be needed to devise the padded message that divide into fixed length blocks. The n-bit message with k-bit padding shall be mapped n-bit values onto (n+k)-bit values so that (n+k)-bit messages are divided into a series of m-bit blocks of equal length.

The proposal authentication procedure based on Fig. 13 shall be described as shown below :

1. Transform the 152-bit message into the 192-bit padded message by appending 40-bit padding.

2. Divide the 192-bit padded message into three equal lengths of 64 bits. This 64-bit message block each shall be used as the DES key.
3. Choose 64 bits randomly for an initializing vector IV(or H_0), which shall act the plain-text input to the DES algorithm.
4. Compute the ciphertext output H_1 by EX-ORing with H_0 and it becomes the input to the hash of the second block.
5. M_2, the next 64-bit message block, becomes the new DES key.
6. H_2 shall become the ciphertext output, $H_2 = E_{M_2}(H_1) \oplus H_1$, that shall be the input to the third block.
7. H_3 shall become $H_3 = E_{M_3}(H_2) \oplus H_2$ of the 64-bit value from which the 18-bit message digest (AUTHR data) can be computed..

Example 6 The 192-bit padded message is divided into three 64-bit blocks, i.e., M_1, M_2, M_3. These three blocks shall be used as DES keys.
Letting Initializing Vector (IV) :
H_0 = 6 7 5 4 2 3 0 1 e f c d a b 8 9
DES Key Data :
M_1 = 7 a 1 3 8 b 2 5 2 4 a f 1 7 c 3
M_2 = 1 7 b 4 3 9 a 1 2 f 5 1 c 5 a 8
M_3 = 5 1 c b 3 6 0 0 0 0 0 0 0 0 0 0

I. First DES operation (see Fig. 12)
(1) Key schedule
Applying Table 8.(PC-1) to the key data M_1, PC-1 output shall become a481394 (left half) and e778253 (right half).
These two halves shall be shifted simultaneously one or two places to the left according to Table 10 and be concatenated them. The 48-bit key data from round 1 through round 16 can be computed as shown below :

K_1=058c457c2c45 K_2=d805187690e5
K_3=4368a26ecode K_4=44822dc2aceb
K_5=b8a5a061f48b K_6=8398062ebf19
K_7=d0060bae146b K_8=282ae27f5472
K_9=619214cadb66 K_{10}=b074284dc82a
K_{11}=0498c614cff8 K_{12}=c00750c47c5c
K_{13}=326072dd9c51 K_{14}=44da15e993fc
K_{15}=ac4520cbe738 K_{16}=07b14291dfab

(2) Encryption
H_0 shall be encrypted using the key M_1 as follows :

The 64-bit H_0 to be enciphered is
H_0 = 6 7 4 5 2 3 0 1 e f c d a b 8 9
Using Table 4, the initial permuted data is
IP = 3 3 0 0 3 3 f f f 0 5 5 f 0 5 5
where L_0 = 330033ff and R_0 = f055f055
Utilizing Table 5, the 48-bit expanded data $E(R_0)$ transformed from the initial permuted data R_0 shall be computed as
$E(R_0)$ = f a 0 2 a b f a 0 2 a b.
EX-ORing this expanded data with the key data K_1 shall yield as follows :
$\Gamma_1 = E(R_0) \oplus K_1 =$
= (fa02abfa02ab) ⊕ (058c457c2c45)
= ff8eee862eee
This data sequence is the input to $(\text{S-box})_1$. From Table 6, the output from $(\text{S-box})_1$ shall be
Ω_1 = d95dbe22
Using Table 7, the permutation function of $(\text{S-box})_1$ output shall produce as
$P(\Omega_1)$ = b5abd4ca
Finally, the 32-bit R_1 shall be obtained by the following formula :
$R_1 = P(\Omega_1) \oplus L_0$
= (65abd4ca) ⊕ (330033ff)
= 86abc735
Thus far, we have completed the computation of first round encryption of H_0.
Computation up to 16th round shall surely involve lengthy calculation. But the computing process of each round is exactly identical. Therefore, the results of all 16 rounds shall be summarized as shown below.
Consider the 16-round DES encryption using (H_0, M_1) :
H_0 = 67452301efcdab89
M_1 = 7a138b2524af17c3

Round i	R_i, $1 \le i \le 16$
1	86abe735
2	470cf9de
3	0ac41f33
⋮	⋮
15	9fb5a2bb
16	00f249d2

The preoutput block ($R_{16} \| R_{15}$) is the concatenation of R_{16} with R_{15} (=L_{16}). Using Table 11, the inverse permutation IP^{-1} applied to the preoutput block ($R_{16} \| R_{15}$) shall compute the output of first DES algorithm, $E_{M_1}(H_0)$.

$R_{16}||R_{15}$ = 00f249d29fb5a2bb

Thus, the data from IP^{-1} shall be

IP^{-1} = a69ba086b33a15bb

EX-ORing IP^{-1} data with H_0 shall become the input H_1 to the second DES algorithm such that

$H_1 = IP^{-1} \oplus H_0$
= (a696a086b33a15bb)⊕
(67452301efcdab89)
= c1de83875cf7be32

Table 11 Inverse of Initial Permutation, IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

II. Second DES operation (see Fig. 13)

Key M_2 : 17b439a12f51c5a8
Input H_1 : c1de83875cf7be32

(1) Key schedule

Applying Table 8 (PC-1) to the key data M_2, PC-1 output shall become ca609e2 (left half) and 1153947 (right half). These two halves shall be shifted according to Table 10 in order to generate the 48-bit key data from round 1 through round 16 as shown below :

K_1 = 0975a4d9e052	K_2 = 07a44cc18bad
K_3 = d005dd21e708	K_4 = 4a4cb2161f99
K_5 = 55c221b83446	K_6 = fca9285b1035
K_7 = 8399a6a8c3a6	K_8 = 82a60b0768a8
K_9 = b822c7106fc3	K_{10} = 691e16643815
K_{11} = 315628ba8051	K_{12} = 64b8c8e700be
K_{13} = c011f483e742	K_{14} = 12e472411b8b
K_{15} = 94ca553ca700	K_{16} = ec4d1216107d

(2) Encryption

The input H_1 shall be encrypted using the key M_2 as shown below :

Using Table 4, the initial permuted data shall be

IP = 33f27a2d6fc052ee

where L_0 = 33f27a2d and R_0 = 6fe052ee

Utilizing Table 5, the 48-bit $E(R_0)$ from the 32-bit R_0 shall be computed as

$E(R_0)$ = 35ff002a575c

EX-ORing $E(R_0)$ with the key data K_1 shall be computed as

$\Gamma_1 = E(R_0) \oplus K_1$
= (35ff002a575c) ⊕ (0975a4d9e052)
= 3c8aa4f3b70e

This data sequence is the input to $(\text{S-box})_1$. Using Table 8.8, the output Ω_1 from $(\text{S-box})_1$ shall be computed as

Ω_1 = 16f90061

From Table 7, the permuted data of Ω_1 becomes

$P(\Omega_1)$ = c4110d56

Thus, the 32-bit R_1 shall be obtained by

$R_1 = P(\Omega_1) \oplus L_0$
= (c4110d56) ⊕ (33f27a2d)
= f7e3777b

This is the result of first round encryption of H_1. The 16-round encryption process using (H_1, M_2) shall be summarized in the following.

Round i	R_i, $1 \le i \le 16$
1	f733777b
2	5c1083ed
3	da7e3b3b
⋮	⋮
15	12712707
16	76862d60

The preoutput block shall be the result of concatenation of R_{16} with R_{15} (= L_{16}), i.e., $R_{16}||R_{15}$. Using Table 11, the inverse of initial permutation IP^{-1} applicable to the preoutput block ($R_{16}||R_{15}$) shall lead to compute the output of second DES algorithm, algorithm, $E_{M_2}(H_1)$.

$R_{16}||R_{15}$ = 76862d6012712707

The data being resulted from IP^{-1} shall be

IP^{-1} = 2eda5e04e06d6110

EX-Oring IP^{-1} data with H_1 shall become the input H_2 to the third DES algorithm such that

$H_2 = IP^{-1} \oplus H_1$
= (2eda5e04e06d6110) ⊕
(c1de83875cf7be32)
= ef04dd83bc9adf22

III. Third DES operation (see Fig. 8.17)

Key M_3 : 51cb360000000000
Input H_2 : ef04dd83bc9adf22

(1) Key schedule

Applying Table 8. PC-1 to the data M3, the PC-1 output shall become 203040(left half) and 604025(right half). These two halves shall be shifted according to Table 10 in order to generate the 48-bit key data from round 1 through round 16 as shown below :

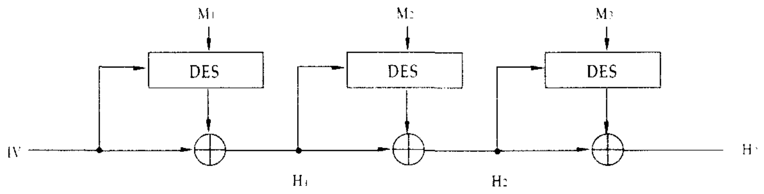


Figure 13 The authentication system based on DM scheme

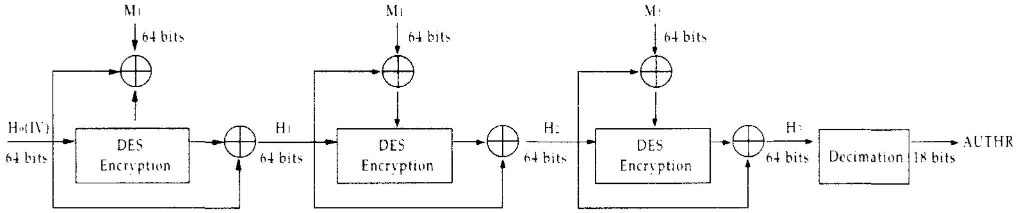


Figure 14 Modified CBC scheme for AUTHR computation

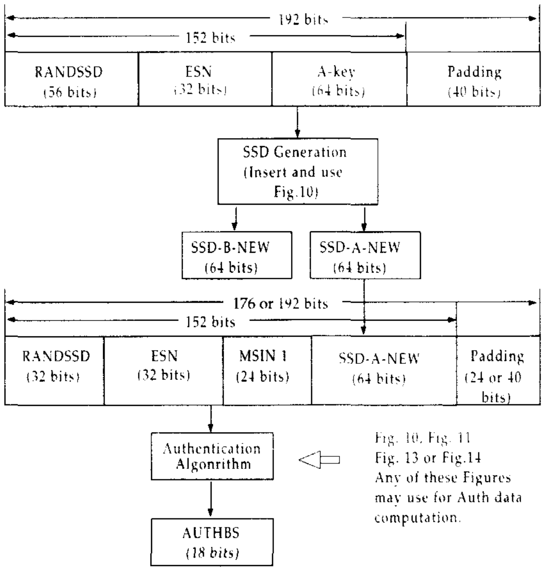


Figure 15 SSD generation and AUTHBS computation

K_1 = 80e000209341	K_2 = 8400066a0008
K_3 = 000702128422	K_4 = 22020000500e
K_5 = 601001080c00	K_6 = 0810200410a0
K_7 = 0180400c6050	K_8 = 800058c00861
K_9 = 00409261c000	K_{10} = 044200028b18
K_{11} = 34010080040a	K_{12} = 021900111410
K_{13} = 020001881304	K_{14} = 0820410d0020
K_{15} = 0900141043e0	K_{16} = 014408406804

(2) Encryption

The input H_2 shall be encrypted using the key M3 as shown below :

Using Table 4, the initial permuted data shall be

IP = 4574574d7d9175e9

Utilizing Table 5, the 48-bit $E(R_0)$ from the 32-bit R_0 can be found as follows :

$E(R_0)$ = bfbca2babf52

EX-ORing $E(R_0)$ with the key K_1 shall be computed as

$\Gamma_1 = E(R_0) \oplus K_1$

= (bfbca2babf52) ⊕ (80e000209341)

= 3f5ca29a2c13

This data sequence Γ_1 is the input to $(S\text{-box})_1$. Using Table 6, the output Ω_1 from $(S\text{-box})_1$ shall be computed as

Ω_1 = 1716ea5

Using Table 7, the permuted data of Ω_1 becomes

$P(\Omega_1)$ = 75605cbb

Thus, the 32-bit R_1 can be computed as

$R_1 = P(\Omega_1) \oplus L_0$

= (75605cbb) ⊕ (4574574d)

= 30140bf6

This is the result of first round encryption of H_2. The 16-round encryption process using (H_2, M_3) shall be summarized as follows :

Beginning with the 16-round DES encryption, the following data shall be used :

H_2 = ef04dd83bc9adf22

M_3 = 51cb36000000000

Round i	R_i, $1 \le i \le 16$
1	30140bf6
2	dcb14381
3	06720b32
⋮	⋮
15	e3fc056f
16	20a43d5c

Thus, the preoutput block shall be

$R_{16} \| R_{15}$ = 20a43d5ce3fc056f

Using Table 11, the data from IP^{-1} shall yield

IP^{-1} = 8e823f2725f6a3b0

EX-ORing this IP^{-1} data with H_2 shall become third DES output as follows :

$H_3 = IP^{-1} \oplus H_2$

=(8e823f2725f6a3b0)⊕(ef04dd83bc9adf22)

= 6186f2a4996c7e92

Transforming this hexadecimal number into the binary digit first and then chopping off 5 bits each from both ends of the binary sequence, we shall have the remainder as follows :

0011000011011111001010100100100110010110
110001111110100

Decimating this binary sequence in its size by picking 1 bit every three bits shall generate the required 18-bit authentication data :

AUTHR = 101111000000001100

8.5 *Modified CBC Scheme*

As seen so far, there are many ways to consider for computation of authentication data. Modified CBC mode shall be also one of them. Block diagram for this scheme is illustrated as shown in Fig. 14. The conceptual idea of this modified CBC scheme is somewhat similar to DM scheme discussed in 8.4.

The 192-bit padded message to be hashed is divided into three blocks, M_1, M_2, and M_3, where M_i, $1 \le i \le 3$, is 64 bits long each.

Set H_0 as an initializing vector(IV). Then the encryption steps shall be shown as follows :

$$H_1 = E_{H_0 \oplus M_1}(H_0) \oplus H_0,$$
$$H_2 = E_{H_1 \oplus M_2}(H_1) \oplus H_1,$$
$$H_3 = E_{H_2 \oplus M_3}(H_2) \oplus H_2,$$

where $H_i \oplus M_j$ denotes the enciphering key.

Since H_3 consists of 64 bits, chopping off 5 bits each from the left and right of H_3 shall be resulted in 54 bits as the reduced sequence of H_3. Decimating the chopped sequence by picking 1 bit every three bits shall form the 18-bit authentication data as expected.

Computation for the 18-bit AUTHR is exactly similar to that of DM scheme. Therefore, it shall be left to the reader for his/her exercise.

9. *SSD Update*

SSD is updated using the SSD generation procedure as illustrated in Fig. 7 or 15. The A-key assigned to the mobile station is 64 bits long and is known only to the mobile station and to its associated Home Location Register /Authentication Center (HLR/AC). SSD updates are carried out only in the mobile station and its associated HLR/AC, not in the serving base station.

The SSD update procedure shall be performed as follows :

- The base station shall send an SSD update order on either the paging channel or the forward traffic channel, with the RANDSSD field set to the same 56-bit random number used in the HLR/AC computations, to the mobile station via the SSD update message.
- Upon receipt of the SSD update message, the mobile station shall set the input parameters and execute the SSD-generation procedure. The mobile station shall compute SSD-A-NEW equal to the 64 most significant bits of SSD-generator output and SSD-B-NEW to the 64 least significant bits of the SSD-generation output.

Example 7 Upon receipt of the SSD update message, the mobile station shall set the following parameters (see Fig. 15) :
RANDSSD (56 bits), ESN (32 bits), A-key (64 bits), and Padding (40 bits).
The 152-bit message length is expanded into the 192-bit padded message by appending a 40-bit padding to the message.
Referring to Fig. 10, we set
M_1 (64-bit input key) : 7a138b2524af17c3
from which K(48-bit DES key) : 058c4517a7a2
M_2 (64 bits) : 17b439a12f51c5a8
M_3 (64 bits) : 51cb360000000000
The permuted output $P(\Omega_i)$, i = 1,2,3,4, of $(\text{S-box})_i$ are, respectively

$$P(\Omega_1) = \text{a34d397f}$$
$$P(\Omega_2) = \text{91193e8e}$$
$$P(\Omega_3) = \text{c3cc8226}$$
$$P(\Omega_4) = \text{02234098}$$

Final output data from S-box shall be a set of four blocks that shall be concatenated to form the 128-bit $P(\Omega)$:
$P(\Omega) = P(\Omega_1) \| P(\Omega_2) \| P(\Omega_3) \| P(\Omega_4)$
= (a34d397f91193e8ec3cc822602234098)
from which we obtain
SSD-B-NEW = a34d397f91193e8e
SSD-A-NEW = c3cc822602234098
Computation of the 18-bit message AUTHBS has been obtained by means of various authentication algorithms already presented in 8.
The AUTHBS computation in Fig. 15 should be done in conjunction with SSD-A-NEW.

10 Message Encryption and Security

To protect sensitive subscriber information, it shall be required to provide enciphering certain fields of selected traffic channel signalling messages. However, message should not be encrypted if authentication is not performed.
Every reverse traffic channel message contains an encryption field which identifies the message encryption mode active at the time the message was created.

10.1 Enciphering Key Generation by Nonlinear Combiners

The key-bit sequence generated from an m-stage LFSR with taps g_1 through g_m and m memory cells is not appropriate for use as a cryptographic key because only 2m bits of either plaintext or ciphertext shall be sufficient to break the key-bit stream by determining both taps and the initial contents of the shift register.
An m-stage LFSR with period $p=2^m-1$ is said to be a maximum-length shift register if it shall generate a PN sequence.
PN sequences generated from LFSRs combined by some nonlinear function have been proposed by several cryptologists for crypto-applications. Many proposed key-stream generators consist of a number of maximum-length shift registers combined by a nonlinear function. A nonlinear combination technique with LFSR sequence shall exhibit certain statistical properties which withstand any cryptanalytic attack.
Consider an m-stage LFSR generating a binary sequence which is applicable to a nonlinear filtering function f, as illustrated in Fig. 16.
A nonlinear keystream should be secure cryptanalytically. For achieving high unpredictability of the generated key sequence, we will develop a nonlinear theory of binary sequences which reflects directly the sum of k variable products in a Boolean function. Let f(*x*) be an arbitrary nonlinear function of variable x_i, $1 \le i \le n$, such that

$$\begin{aligned} f(x) = {} & a_1x_1 + a_2x_2 + \cdots + a_nx_n + a_{12}x_1x_2 \\ & + \cdots + a_{n-1,n}x_{n-1}x_n + a_{123}x_1x_2x_3 \\ & + \cdots + a_{n-2,n-1,n}x_{n-2}x_{n-1}x_n \\ & \cdots\cdots\cdots\cdots\cdots\cdots \\ & + a_{12\cdots n}x_1x_2\cdots x_n \end{aligned}$$

where a product of variables is called a *k*th order product. Therefore, the order of f(*x*) will be the maximum among the orders of its product terms.

Example 8. Figure 16 illustrates the 32-stage LFSR combined with a nonlinear filtering function $f(x)$ which shall be applicable to CDMA mobile communication system. Assume that the generator polynomial of maximum-length LFSR is $g(x) = 1+x+x^2+x^{22}+x^{32}$.

A 128-bit SSD, which is stored in the mobile station's semi-permanent memory, is partitioned into two distinct subsets : The 64-bit SSD-A is used for supporting the authentication procedure ; and the 64-bit SSD-B is for supporting the message confidentiality for CDMA.

The 64-bit SSD-B shall be divided into two halves (i.e., 32 bits each). The upper 32 bits shall be assigned to the 32-stage LFSR as the initial values ; while the lower 32 bits shall be assigned to the nonlinear function as shown in Fig. 7. The lower 32 bits shall again partition evenly into four 8 bits.

Assume that the 64-bit SSD-B is 30a7f415 83c7519a (in hexadecimal notation). Then, the upper 32-bit initial value is

30a7f415 (hexadecimal) =

00110000101001111111010000010101 (binary)

while the lower 32-bit 83c7519a (hexadecimal)

= 10000011110001110101000110011010 (binary).

The lower 32 bits shall be divided into four 8-bit identical lengths in order to utilize them as the coefficients of nonlinear function f(x) as follows :

First nonlinear combiner's coefficients :

$(a_1, a_2, a_3, \cdots)$

83 (hexadecimal)=10000011(binary) → $x_1+x_7+x_8$

Second nonlinear combiner's coefficients :

$(a_{12}, a_{13}, a_{14}, \cdots)$

c7 (hexadecimal) = 11000111 (binary) →

$x_1x_2 + x_1x_3 + x_1x_7 + x_1x_8 + x_1x_9$

Third nonlinear combiner's coefficients :

$(a_{123}, a_{124}, a_{125}, a_{126}, \cdots)$

51 (hexadecimal) = 01010001 (binary) →

$x_1x_2x_4 + x_1x_2x_6 + x_1x_2x_{10}$

Fourth nonlinear combiner's coefficients :

$(a_{1234}, a_{1235}, a_{1236}, \cdots)$

9a (hexadecimal) = 10011010 (binary) →

$x_1x_2x_3x_4 + x_1x_2x_3x_7 + x_1x_2x_3x_8 + x_1x_2x_3x_{10}$

Thus, the nonlinear function f(x) can be obtained as

$$\begin{aligned} f(x) = \; & x_1+x_7+x_8+x_1x_2+x_1x_3+x_1x_7+x_1x_8+x_1x_9 \\ & +x_1x_2x_4+x_1x_2x_6+x_1x_2x_{10}+x_1x_2x_3x_4 \\ & +x_1x_2x_3x_7+x_1x_2x_3x_8+x_1x_2x_3x_{10} \end{aligned}$$

In conjunction with the upper 32-bit initial values, the corresponding LFSR contents shall be given as

$$x_1=0,\ x_2=0,\ x_3=1,\ x_4=1,\ x_5=0$$
$$x_6=0,\ x_7=0,\ x_8=0,\ x_9=1,\ x_{10}=0$$

Using these LFSR contents, the nonlinear combiner's coefficients can be determined as follows :

1st coefficients : $x_1+x_7+x_8 = 0 \oplus 0 \oplus 0 = 0$

2nd coefficients : $x_1x_2+x_1x_3+x_1x_7+x_1x_8+x_1x_9$
$= 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$

3rd coefficients : $x_1x_2x_4+x_1x_2x_6+x_1x_2x_{10}$
$= 0 \oplus 0 \oplus 0 = 0$

4th coefficient : $x_1x_2x_3x_4+x_1x_2x_3x_7+x_1x_2x_3x_8$
$+x_1x_2x_3x_{10}$
$= 0 \oplus 0 \oplus 0 \oplus 0 = 0$

Thus, the combiner's output data is

$0 \oplus 0 \oplus 0 \oplus 0 = 0.$

Applying the 32-bit initial value to LFSR in Fig. 16 and shifting LFSR bitwise to the right, the PN sequence being outputted from the nonlinear combiner shall be computed as shown in Table 12

Table 12 Computation of Nonlinear Combiber's Output

Shift No.	LFSR Contents	Combiner's output (bits)
0	00110000101001111111010000010101	0
1	11111000010100111111100000001010	1
2	01111100001010011111110000000101	0
3	11011111000010100111111000000010	0
4	01101111100001010011111100000001	0
5	11010111110000101001111010000000	1
···	· · · · · · · · · · · ·	· · ·
14	11100000011010111100000010011110	1
15	01110000001101011110000001001111	0
···	· · · · · · · · · · · ·	· · ·
28	10101011110110011000001000100110	1
29	01010101111011001100000100010011	1
30	11001010111101100110001010001001	0
···	· · · · · · · · · · · ·	· · ·
49	11101110100011011110001101111101	0
50	10010111010001101111001110111110	1
···	· · · · · · · · · · · ·	· · ·

The linear combiner's output Z in Table 12, which shall be used as a key sequence, can be found as shown below :

Combiner's output sequence Z :
010001011110101011001010110011011
100010010101001101010010011110011
0101001110010011000000101001100101

This PN sequence Z shall be used for the enciphering key.

Example 7 The LFSR content corresponding to the shift number 14 in Table 12 is
11100000011010111100000010011110.

Using the contents at the shift No. 14, the *i*th stage content of LFSR shall be

$x_1=1$, $x_2=1$, $x_3=1$, $x_4=0$, $x_5=0$, $x_6=0$, $x_7=0$, $x_8=0$, $x_9=0$, and $x_{10}=0$. Hence, the nonlinear combiner's coefficients are computed as follows :

1st coefficient : $x_1+x_7+x_8 = 1 \oplus 0 \oplus 0 = 1$

2nd coefficient : $x_1x_2+x_1x_3+x_1x_7+x_1x_8+x_1x_9$
$= 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$

3rd coefficient : $x_1x_2x_4+x_1x_2x_6+x_1x_2x_{10}$
$= 0 \oplus 0 \oplus 1 = 1$

4th coefficient : $x_1x_2x_3x_4+x_1x_2x_3x_8+x_1x_2x_3x_{10}$
$= 0 \oplus 0 \oplus 0 \oplus 1 = 1$

Thus, the nonlinear combiner's output bit corresponding to the shift No. 14 is $1 \oplus 0 \oplus 1 \oplus 1 = 1$.

10..2 Encryption and Message Security

Data transmitted on the reverse traffic channel is grouped into 20 ms frames. Data frames may be transmitted on the reverse traffic channel at variable data rates of 9600, 4800, 2400, and 1200 bps. The reverse traffic channel is used for the transmission of user and signaling information to the base station during a call. The reverse traffic channel may use any of transmission rates of 9.6, 4.8, 2.4, and 1.2 kbps.

Each data frame is 20 ms in duration and consists of either information-CRC -tail bits or information and tail bits depending on transmission rates. For example, for 4800 bps, the frame bits(96 bits) consists of the 80-bit information, 8-bit CRC, and 8-bit encoder tail.

The message encryption can be considered in two ways-'external encryption' and 'internal encryption' as shown in Fig. 17.

The message information m is first enciphered with the key stream K generated by Fig. 16 and then encoded with the (3, 1, 8) convolutional encoder. We shall call this kind of encryption scheme 'external encryption'. The second scheme to be considered is illustrated as shown in Fig. 17(b). Encoding precedes encryption and deciphering shall be preceded decoding. We call this kind of encryption scheme 'internal encryption'.

Example 9 Consider the 'external encryption' case in this example. The 80-bit information data for the 4800-bps frame (20ms) is assumed as

1010110011 1001101111 0010010100
0100110011 0011001000 1110010110
1001100110 0110011110

PN key sequence produced from the nonlinear key generator was computed in 10.1 as shown below :

0100010111 1010101100 1010110011
0111000100 1010100110 1010010011
1100110101 0011100100 1100000010
1001100101

Using these information and key sequences, the encryption data sequence shall then become as follows :

1110100100 0011000011 1000100111
0011110111 1001101110 0100000101
0101010011 0101111010

Next, using the generator polynomial, $g(x)=1+x+x^3+x^4+x^7+x^8$, for computing CRC at 4800 bps, CRC computation shall be accomplished using $g(x)$ and the all-one initial contents of the register as shown below :

CRC = 11000001

Thus, concatenation of the 80-bit encryption data with the 8-bit CRC and the 8-bit all-zero encoder tail shall yield as

1110100100 0011000011 1000100111
0011110111 1001101110 0100000101
0101010011 0101111010 1100000100
000000

This encrypted ciphertext is subject to inputting to the (3, 1, 8) convolutional encoder. The encoder output shall be

c_0 : 1100101110 1110011111 0100110001
1101100101 1111010000 0110000011
1001101101 1100000010 1001101100 101111
c_1 : 1000111100 0000000111 0100000000
0101111000 1100010000 1111111100

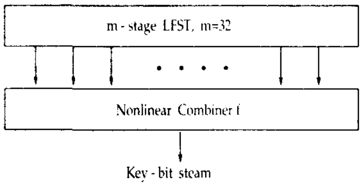
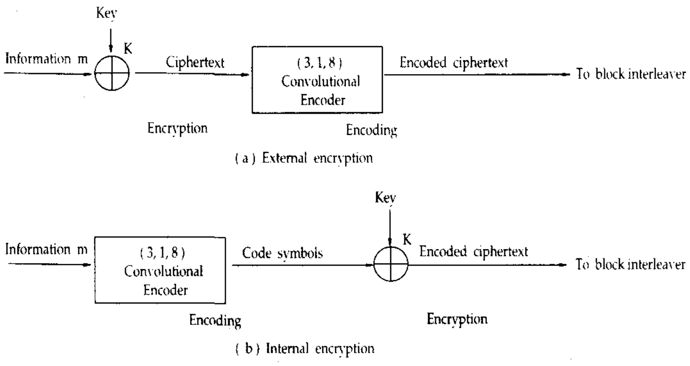


Figure 16 Key - bit generator filtered by the nonlinear combiner



(a) External encryption

(b) Internal encryption

Figure 17 Message encrtption Scheme

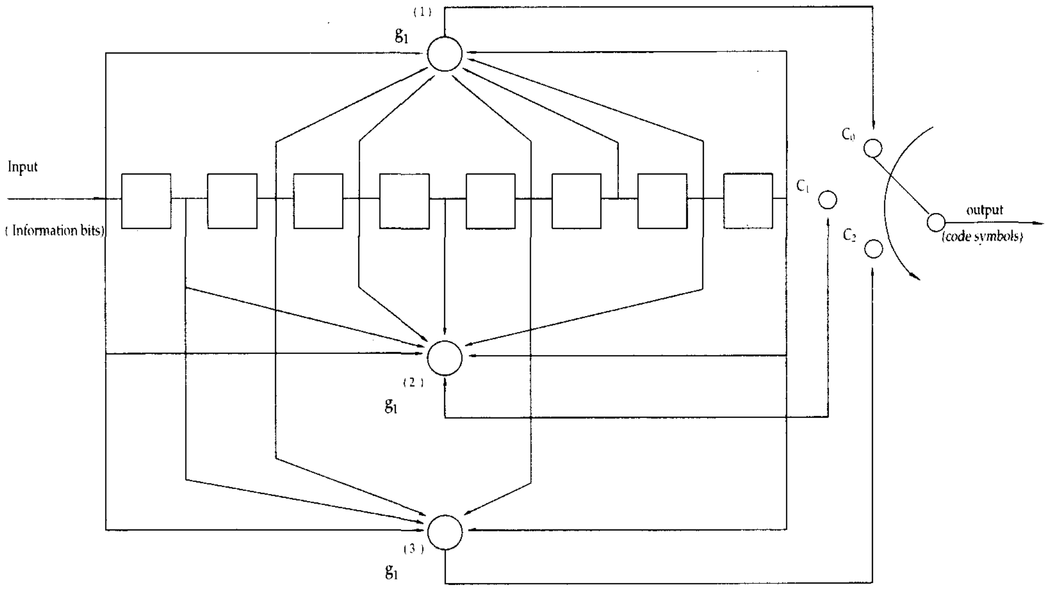


Figure 18 The (3, 1, 18) convolutional encoder

0000111010 0101101000 1101101011 110011
c_2 : 1010000001 1010000100 0111001010
1111101111 1111101111 1110110111
1000010001 1110111111 1011110100 001001

As a result, final encoded ciphertext symbols are obtained as

11110000100011001011011010000 1
1011001010000001001001111110110
0001110010011001000000000001100
1001110011111100100111000001100
1111111011010011100010010010 01
0111111110100110110100111011 01
1010000000100110011110100010101
1011110010100110010110011010 01
1110100011111110011101010100 10
11001010110011011 1

Example 10 Consider this time the enciphering problem of all 96-bit concatenated data per frame, i.e., information data at the transmission rate of 4800 bps was assumed to be

1010110011 1001101111 0010010100 0100110011
0011001000 1110010110 1001100110 0110011110

The CRC bits was computed as
10111001
Thus, the 96-bit frame data for the 4800 bps is simply computed by adding the encoder tail bits (extra all-zero bits) as follows :
1010110011 1001101111 0010010100 0100110011
0011001000 1110010110 1001100110 0110011110
1001110100 000000

This is the data sequence to be enciphered with the following key sequence which is generated from the nonlinear combiner discussed in 8.13.1. The key sequence is
0100010111 1010101100 1010110011 0111000100
1010100110 1010010011 1100110101 0011100100
1100000010 100110

EX-ORing the information frame data with the key sequence shall generate the following ciphertext data sequence :
1110100100 0011000011 1000100111 0011110111
1001101110 0100000101 0101010011 0101111010
0101110110 100110

This is the encrypted frame data sequence to be subjected as the (3, 1, 8) convolutional encoder input.
The mobile station shall convolutionally encode the encrypted data sequence transmitted on the reverse traffic channel prior to block interleaving. The encoder generates three code symbols (c_0, c_1, c_2) for each bit of encoder input. Three code symbols can be computed as shown below :

c_0 : 1100101110 1110011111 0100110001
1101100101 1111010000 0110000011
c_1 : 1000111100 0000000111 0100000000
0101111000 1100010000 1111111100
c_2 : 1010000001 1010000100 0111000010
0111001010 1111101111 1110110111

Therefore, the final encrypted code symbol shall be described as

1111000010001100101101101000001
1011001010000001001001111110110
0001110010011001000000000001100
1001110011111100100111000001100
1111111011010011100010010010 01
0111111110100110110100111011 01
1010000000100110011110100010101
1011110010100110010110011010 01
0000011001100011010100100110 10
0110100111111100010

So far, the case for the external message encryption has been discussed. The internal encryption for the message information shall be presented in the following. Data frames transmitted on the reverse traffic channel are grouped into 20 ms each. Each frame at the transmission rate of 4800 bps is 96 bits long which are composed of the 80-bit information, 8-bit CRC, and 8-bit encoder tail.
The internal encryption scheme is depicted as shown in Fig. 17(b). The 96-bit message frame shall be first encoded by the (3,1,8) convolutional encoder. As a result, the code symbols outputted from the convolutional encoder are 3 times longer than the message frame bits at the encoder input. These code symbols shall be encrypted with the key stream generated from the key-bit generator filtered by the nonlinear combiner being fully discussed in 10.1
The internal encryption method shall be demonstrated by means of the following example.

Example 11 Based on Fig. 17(b), the internal encryption problem at the 4.8 kbps rate shall be considered in this example.
Assumed information sequence :
1010110011 1001101111 0010010100 0100110011
0011001000 1110010110 1001100110 0110011110
Computed CRC bits : 10111001

Encoder tail bits : 00000000

Thus, the information data concatenated with CRC bits and tail bits is expressed by
1010110011 1001101111 0010010100 0100110011
0011001000 1110010110 1001100110 0110011110
1001110100 000000
This data sequence shall be input to the (3, 1, 8) convolutional encoder depicted as shown in Fig. 18.
These code symbols, corresponding to each input bit to encoder, are computed as

c_0 : 1001010001 1000100001 0111110000
1001100111 1001100011 1111110111
1011100111 0011001010 1100010011 011011
c_1 : 1110010000 1100100001 0001000000
0001101100 1010101100 0101110010
1101010001 0101010001 0100101001 010111
c_2 : 1101010001 1001111110 0110100011
0100111110 1100110100 0010000111
0101000101 1001100011 0001111010 110101

Concatenation of all three (c_0, c_1, c_2) symbols from the modulo-2 adders resulted in a single output sequence which shall be input to symbol repeater and block interleaver. Since the code rate is 1/3, the code symbol output shall be 288 bits as shown below :
1110110101 0100011100 0000000101 1110100000
0111100100 1001001110 0001011011 1010110000
0000001001 1000010001 1011100101 1111101100
1110010101 0011100101 0011100100 1001101011
1011011000 0101111101 1100111001 1110001000
0101100111 0010101001 1100101010 0000101011
1001100000 0101110101 1000101110 0011111000
11110111

Next, the key stream (288 bits only) generated from the key generator filtered by the nonlinear combiner shall be obtained as
0100010111 1010101100 1010110011 0111000100
1010100110 1010010011 1100110101 0011100100
1100000010 1001100101 0001010110 0001110010
1011111101 0001111101 0101000110 0100001110
0100010010 1110100101 1110010001 0101000010
1010110100 1100001101 0011111110 0001111110
0010101111 1000101000 0100101000 0110000001
1110000

Finally, the encrypted data sequence can be computed EX-ORing the code-symbol output with the key stream as follows :
1010100010 1110110000 1010110110 1001100100
1101000010 0011011101 1101101110 1001010100
1100001011 0001110100 1010110011 1110011110
0101101000 0010011000 0110100010 1101100101
1111001010 1011011000 0010101000 1011001010
1111010011 1110100100 1111010100 0001010101
1011001111 1101011101 1100000110 0101111001
00010111

11. Concluding Remarks

We have presented a brief survey on functional characteristics of all the code channels between the base/mobile stations in CDMA cellular system. Particular emphasis is placed on authentication and message privacy. The successful authentication cab be achieved only when the base station processes identical sets of shared secret data (SSD) with the mobile station. In this paper several techniques are proposed for the authentication data computation by employing a one-way hash function. To protect sensitive subscriber information, it shall be required to provide enciphering certain fields of selected traffic channel signaling message.
It is regrettable that SSD-A and SSD-B computation by MD 5 algorithm is impossible to include them in this paper due to the limitted space to be allowed.

12. References

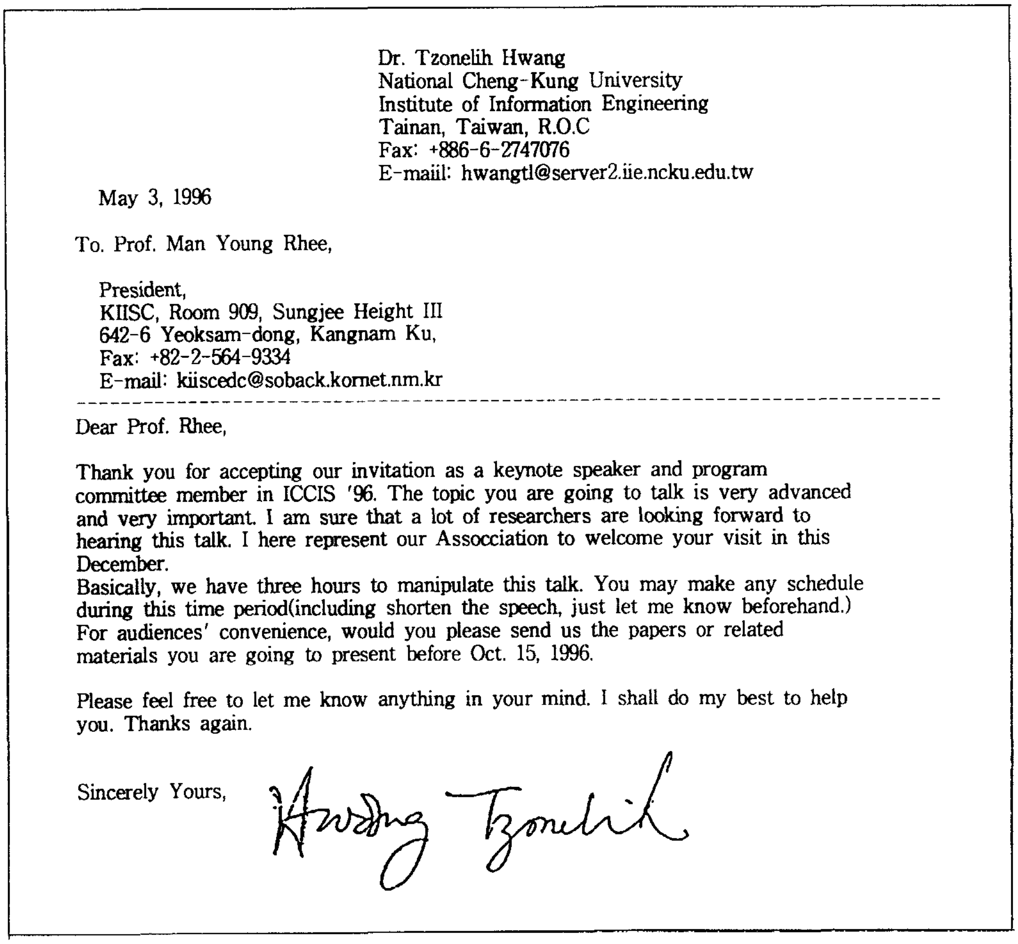
1. TIA/EIA/IS-95, Mobile Station - Base Station Compatibility Standard for Dual - Mode Wideband Spread Spectrum Cellular System, July 1993.
2. JTC(AIR)'95.02.02-037R1, W-CDMA Air Interface Compatibility Standard for 1.85 to 1.99 GHz PCS Applications, February 2, 1995.
3. Bruce Schneier : Applied Cryptography : Protocols, Algorithms, and Secure Code in C, Wiley, New York, 1994.
4. Gustavus, J. Simmons (Edited) : Contemporary Cryptology : The Science of Information Integrity, IEEE Press, New York, 1992.
5. Man Y. Rhee : Cryptography and Secure Communications, McGraw-Hill, New York, 1994.

ABOUT THE AUTHOR

Man Young Rhee is professor emeritus of electrical engineering at Hanyang University, Seoul, Korea. At this same University he also served as vice president. At present he is president of Korea Institute of Information Security and Cryptology; chairman of board of directors, Korea Information Security Agency, Republic of Korea. He received his B.S.E.E. degree from Seoul National University and his M.S.E.E. and Ph.D. degrees from the University of Colorado, Boulder, Colorado.

In the United States, Dr. Rhee taught at Virginia Polytechnic Institute and State University as a professor and was employed at the Jet Propulsion Laboratory, California Institute of Technology as a member of research staff. In Korea he was vice president of the Agency for Defence Development, Ministry of National Defence, Republic of Korea, president of the Korea Telecommunications Company (a government cooperation where the ESS telephone exchange system was first developed under contract with ITT/BTM); and president of Samsung Semiconductor and Telecommunications Company.

Dr. Rhee is the author of *Error Correcting Coding Theory* (1989), *Cryptography and Secure communications* (1994) published by McGraw-Hill in the United States and is a recipient of the academic achievement prize from the National Academy of Sciences, Republic of Korea. At present he is senior fellow, the Korean Academy of Science and Technology. His current interest is in the research area of CDMA digital cellular communications, error control coding, cryptography and its relevant applications.



Dr. Tzonelih Hwang
National Cheng-Kung University
Institute of Information Engineering
Tainan, Taiwan, R.O.C
Fax: +886-6-2747076
E-maiil: hwangtl@server2.iie.ncku.edu.tw

May 3, 1996

To. Prof. Man Young Rhee,

President,
KIISC, Room 909, Sungjee Height III
642-6 Yeoksam-dong, Kangnam Ku,
Fax: +82-2-564-9334
E-mail: kiiscedc@soback.kornet.nm.kr

Dear Prof. Rhee,

Thank you for accepting our invitation as a keynote speaker and program committee member in ICCIS '96. The topic you are going to talk is very advanced and very important. I am sure that a lot of researchers are looking forward to hearing this talk. I here represent our Assocciation to welcome your visit in this December.
Basically, we have three hours to manipulate this talk. You may make any schedule during this time period(including shorten the speech, just let me know beforehand.) For audiences' convenience, would you please send us the papers or related materials you are going to present before Oct. 15, 1996.

Please feel free to let me know anything in your mind. I shall do my best to help you. Thanks again.

Sincerely Yours, Hwang Tzonelih

□ 著者紹介

이 만 영(종신회원)

1924년 11월 30日生
서울大學校 電氣工學科 工學士(BSEE)
美國 Colorado 大學校 工學碩士(MSEE) 및 工學博士(Ph. D.)
美國 Virginia 州立大 工科大學 敎授
美國 California Institute of Technology, JPL 責任硏究員
國防科學硏究所 第1副所長/韓國電子通信 社長/三星半導體通信 社長/漢陽大副總長
現 : 漢陽大 名譽敎授/韓國通信情報保護學會 會長
著書 : Error Correcting Coding Theory, McGraw-Hill, New York, 1989.
Cryptography and Secure Communications, McGraw-Hill, New York, 1993.