Securing Against Fraud in Mobile Communications: Systems Design and Development in 3G Mobile Networks

by

Yujiro Mochizuki

M.S., Management of Technology, Massachusetts Institute of Technology, 2004
B.A., Economics, Keio University, 1996

Submitted to the Engineering Systems Division
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Technology and Policy
at the
Massachusetts Institute of Technology

February 2006

Signature of Author: ______________________
Technology and Policy Program, Engineering Systems Division
November 12, 2005

Certified by: ______________________
Michael A. Cusumano
Sloan Management Review Professor of Management
Sloan School of Management
Thesis Supervisor

Certified by: ______________________
Joel Cutcher-Gershenfeld
Executive Director, Engineering Systems Learning Center
Senior Research Scientist, Sloan School of Management
Thesis Reader

Accepted by: ______________________
Dava J. Newman
Professor of Aeronautics and Astronautics and Engineering Systems
Director, Technology and Policy Program

Securing Against Fraud in Mobile Communications: Systems Design and Development in 3G Mobile Networks

by

Yujiro Mochizuki

Submitted to the Engineering Systems Division on November 12, 2005
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Technology and Policy

ABSTRACT

Network security ensures the consistency, integrity, and reliability of telecommunications systems. Authorized network access prevents fraudulent communications and maintains the availability of the systems. However, limited development time, cost reduction pressure and requirement for high reliability in software development have forced mobile carriers to implement the insufficient and inflexible authentication mechanisms. Technical specifications including network architecture, network protocols, and security algorithm are widely available to the public. In addition, both secured and unsecured networks are interconnected by global roaming services. The inadequate system design will make the mobile systems vulnerable to unauthorized access to mobile communications.

Compared with GSM mobile systems, 3G mobile systems are equipped with more robust and flexible security mechanisms. The official position taken by mobile carriers, such as NTT DoCoMo, KDDI, and Vodafone, is that fraudulent communications, usually in the form of cloned mobile phones, are impossible with their 3G mobile systems. Examining the NTT DoCoMo's case, however, we find that this statement is based on weak security assumptions.

In order to avoid potential threats and to secure the 3G mobile systems, this thesis (1) explores the security architecture and mechanisms in 3G systems, (2) analyzes the current platform architecture and platform innovations of the network software, and (3) suggests a secure system design and development.

Thesis Supervisor: Michael A. Cusumano
Title: Sloan Management Review Professor of Management

Table of Contents

List of Figures

List of Tables

Acknowledgements

In every phase of this study, many people supported me with many suggestions and words of encouragement. I would never have been able to accomplish this work without these people.

First and foremost, I would like to extend my sincere gratitude to my thesis advisor, Professor Michael A. Cusumano, for providing me opportunities to work with him, as well as for his superb guidance, crucial insights, and invaluable support throughout the course of both the Management of Technology program and Technology and Policy Program.

I would like to thank my thesis reader, Dr. Joel Cutcher-Gershenfeld, for his constructive and valuable advice. His knowledge, perspectives and suggestions led me to accomplish this work. I could not have imagined having a better mentor for my thesis.

I wish to express my gratitude to Professor Yoshiko Takenaka of Keio University, who encouraged me to pursue further studies at MIT Sloan School of Management and Engineering Systems Division. I also would like to express my deep appreciation to Ms. Sydney E. Miller for her great support and encouragement.

Finally, I am deeply grateful to my parents, Takeo and Ikuko, for your constant support and love since my childhood. I will never forget the Zen comments you taught me.

Do not think
The moon appears when the clouds are gone.
All the time it has been there in the sky
So perfectly clear.

-Zen Comments on the Mumonkan by Zenkei Shibayama

Yujiro Mochizuki
Boston, Massachusetts
November 2005

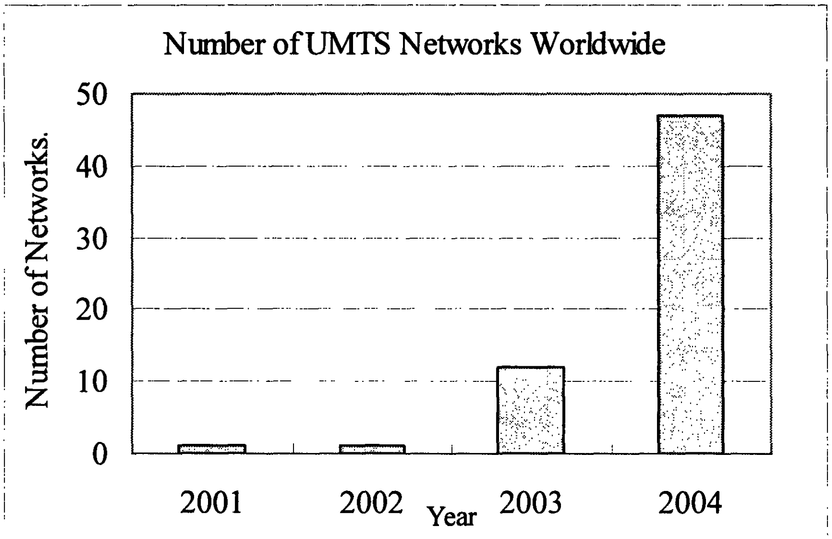
1 Introduction

Network security ensures the consistency, integrity, and reliability of telecommunications systems. Authorized network access prevents fraudulent communications and maintains the availability of the systems. Security is an essential part of network communications and has always been an issue for mobile systems. When first generation analog systems were designed, little attention was paid to network security [1][2], and insecure systems allowed eavesdropping in user traffic and mobile phone cloning [3][4]. Calls from cloned mobile phones were charged to the original subscriber's account. Against fraudulent communications, second generation digital systems were designed to apply digital ciphering mechanisms. Global System for Mobile communications, GSM, deployed in 1990, was the first public mobile communication system to implement integrated cryptographic mechanisms using a smart card or Subscriber Identity Module (SIM). According to the GSM Association [6], GSM technology is currently used by more than one-sixth of the world's population. It is estimated that over 1,296 million GSM subscribers existed across more than 210 countries/areas of the world at the end of December 2004.

Security features implemented in GSM have contributed to preventing fraudulent communications. However, some of the security mechanisms in GSM already have become insufficient and outdated. For example, COMP128, one of the main authentication algorithms in GSM, was broken in 1998 when Ian Goldberg and David Wagner of the University of California at Berkeley demonstrated a flaw in it [7]. Replay attacks on the security algorithm (called "A8") demonstrated by Goldberg and Wagner took just 2^19 queries, roughly 8 hours. This flaw allowed attackers to make a cloned mobile handset and then make fraudulent calls charged to the target user's account. Clearly, more advanced security mechanisms will be required for the next generation systems.

Security for the Universal Mobile Telecommunications System (UMTS) builds on the success and lessons learned in GSM systems. The basic authentication procedure is similar to GSM, but UMTS systems focus on mutual authentication between mobile handset and serving network to avoid fraudulent communications. Security algorithms have become more sophisticated, and new and longer security parameters have been applied. A new security message has also been implemented to detect potentially fraudulent communications.

As of January 2005, more than 60 3G/UMTS networks using W-CDMA (Wideband Code Division Multiple Access) technology are operating commercially in 29 countries (see Figure 1-1). In its initial phase, UMTS offers theoretical bit rates of up to 384 kbps in high-mobility situations, rising as high as 1.5 Mbps in stationary/nomadic user environments. High Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA) technologies are already standard, and HSDPA with around 14Mbps downlink speed will be released in 2006.



Source: Based on the UMTS Forum, 2005, "3G/UMTS Commercial Deployments."

Figure 1-1: Number of UMTS Networks Worldwide (Total as of January 2005)

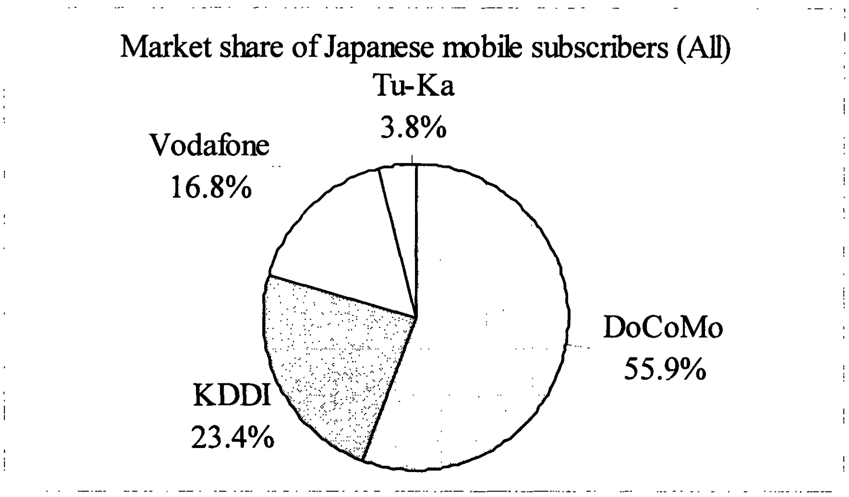
Every time subscribers use mobile handsets for such things as call setup, location update, web browsing, voice/movie message, and short message services, the security functions are executed. Enhanced security features employed in 3G mobile systems are expected to secure these true mobile broadband communications.

Previous research has focused on security mechanisms, specific security technologies and potential security holes ([1][2][3][4][5]), but no research has bridged the gap between security principals and actual implementation. Examining the NTT DoCoMo's case, this thesis will fill this gap and suggest a secured system design and development on the basis of platform thinking.

1.1 Operation of the 3G Systems

In order to investigate security issues in 3G mobile systems, Japan is the best example, for Japan has a longest history in the operation of 3G mobile systems and has implemented the most advanced 3G systems.

The first mobile carrier worldwide to initiate 3G services was NTT DoCoMo, the largest mobile communications carrier in Japan with nearly 50 million subscribers (see Figure 1-2). DoCoMo launched FOMA (Freedom of Mobile multimedia Access) 3G services based on W-CDMA on October 1, 2001. KDDI, Japan's second largest mobile carrier with 20 million subscribers, began its 3G services which uses CDMA2000 1X on April 1, 2002. Vodafone, Japan's third largest mobile telecommunications carrier with almost 15 million subscribers, initiated its Vodafone Global Standard 3G service, which uses W-CDMA and the latest version of 3GPP (Third Generation Partnership Project) standards, on December 20, 2002. In total, nearly 40 million subscribers enjoy 3G services in Japan (see Figure 1-3).

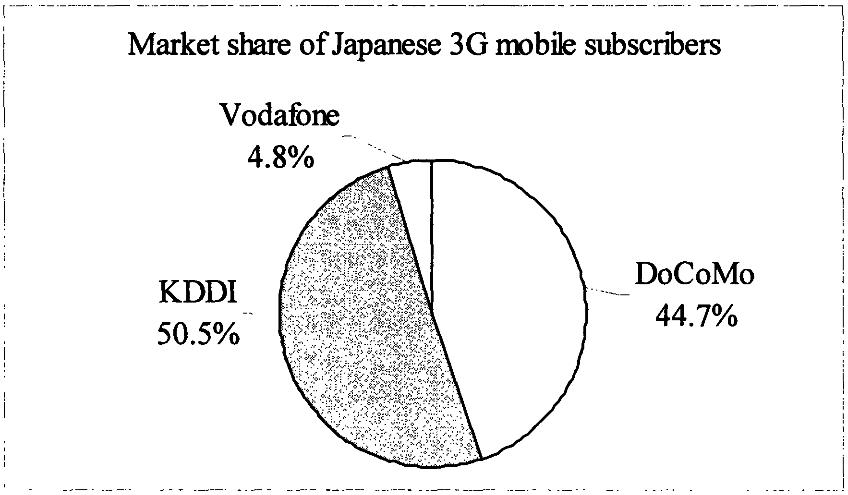


	DoCoMo	KDDI	Vodafone	Tu-Ka	Total
Number of subscribers	49,994,300	20,939,000	14,996,000	3,435,900	89,365,200
Market share	55.9%	23.4%	16.8%	3.8%	100%

Note: Tu-Ka is a subsidiary of the KDDI group.

Source: Telecommunications Carriers Association (TCA).

Figure 1-2: Market Share of Japanese Mobile Subscribers (Total as of October 2005)



	DoCoMo	KDDI	Vodafone	Total
Number of subscribers	17,584,400	19,849,500	1,894,900	39,328,800
Market share	44.7%	50.5%	4.8%	100%

Note: Tu-Ka is a subsidiary of the KDDI group.

Source: Telecommunications Carriers Association (TCA).

Figure 1-3: Market Share of Japanese 3G Mobile Subscribers (Total as of October 2005)

In November 2003, a Japanese mobile subscriber filed a suit against her mobile carrier to be reimbursed for huge packet communication charges that she had not used. The plaintiff was a junior high school student. She checked her usage logs after receiving huge bills and found that some calls were made during classroom hours. The plaintiff claimed that cloned mobile phones had charged huge amounts of communication fees to her account. A non-profit organization checked plaintiff's mobile phone, and then put it inside a safe for a month, but no fraudulent communications were found. Mobile carriers including DoCoMo, KDDI and Vodafone made an official announcement that such fraudulent communications, mainly using cloned mobile phones, are impossible in 3G mobile systems.

3G mobile systems were not stable in 2003, and billing systems sometimes had software defects. The junior high school student's huge bill was considered a software defect in the billing systems. However, the key issue was that all the mobile carriers officially denied the existence of this type of fraudulent communications, stating categorically that the network access security mechanism implemented in 3G systems is "perfect." After investigating the case of DoCoMo, I found that this statement needs to be reconsidered.

1.2 Purpose of This Thesis

This thesis aims to accomplish three goals: (1) explore the security architecture and mechanisms in 3G systems, (2) analyze the current platform architecture and platform innovations of the network software, and (3) suggest a secured system design and development.

Telecommunication today is a basic service for individuals and corporations, and security ensures the integrity, reliability, and consistency of the network. During the connection setup phase authentication procedure is always executed to provide session keys for confidentiality and integrity protection. Exploring the security mechanisms standardized in 3G systems, we can identify the strengths and weaknesses of the 3G network architecture.

Under short time-to-market and cost reduction pressure, network software must satisfy various requirements, such as high reliability, security, compatibility, real-time response and configurability. Platform architecture and platform thinking are fundamental factors to fulfill these requirements. Applying the DoCoMo's current platform architecture, I would like to show the innovation styles of network software. In particular, this thesis focuses on why (1) architectural innovations are difficult to accomplish, (2) the current platform has resulted in insufficient implementation and (3) platform renewal is crucial to break through the old constraints.

DoCoMo's case and interviews with network engineers[1] in other 3G systems suggest that huge differences exist between global standards and actual implementation. Even a single security breach can result in critical and costly failures. Thus, insufficient security features means more vulnerable against fraudulent communications. Examining the existing network software, this thesis clarifies the possible security threats and suggests a secured system design and development.

As of this writing, no cloned mobile handsets have been found in 3G mobile systems. However, it is crucial to design and develop flexible systems that prevent future fraud before we encounter unexpected problems. By addressing potential architectural and implementation problems beforehand, mobile carriers can manage many of tomorrow's security problems.

[1] These engineers are involved in Nokia, Ericsson, KDDI and Vodafone.

1.3 Scope of This Thesis

First, this thesis focuses on developing network software in mobile communications, especially "core network" software (Figure 1-4). The core network provides switching, routing, location management, and database functions. All data from handsets is transferred via the core network. Core networks are slightly different from UMTS (3GPP) and CDMA2000 (3GPP2) networks, but examining these differences is beyond the scope of this thesis, as is hardware (infrastructure) and software in handsets.

Second, the analysis herein is based on NTT DoCoMo's 3G mobile systems in Japan, for the following reasons: (1) DoCoMo is the dominant mobile carrier and its security systems have major impacts on the 3G networks, (2) the 3G systems in Japan is the most advanced and has the longest history worldwide and (3) cooperating with other manufacturers, such as NEC, NTT Comware, and Fujitsu, DoCoMo builds its network software in-house. Other mobile carriers (KDDI and Vodafone) outsource the network software development. Compared with its competitors, DoCoMo has considerable knowledge about network software and security issues.

Third, this thesis focuses on network access security. Five security feature groups are defined in 3GPP specifications: (1) network access security, (2) network domain security, (3) user domain security, (4) application domain security, and (5) visibility and configurability of security. Each of these feature groups meets certain threats and accomplishes certain security objectives. Among the security feature groups network access security, especially user authentication and network authentication, is directly related with fraudulent communications.

Finally, this thesis applies the underlying platform concepts and innovations in mobile network software. Meyer and Lehnerd define platform as "a set of subsystems and interfaces that form a common structure from which a stream of related products can be efficiently developed and produced" [8]. Therefore, in this thesis "platform" means the core network software that provides fundamental communication services for end users. Due to the limited scope of this study, operating system and contents-based services (e.g., web application services) are not included.

1.4 Overview of the Core Network

The core network is divided into two domains: (1) circuit switched (CS) domain and (2) packet switched (PS) domain. The circuit-switched elements are Mobile services Switching Center (MSC), Visitor Location Register (VLR), and Gateway MSC (GMSC). Packet switched elements are Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). Both CS/PS domains share some network elements such as Home Location Register (HLR) and Authentication Center (AuC) (see Table 1-1).

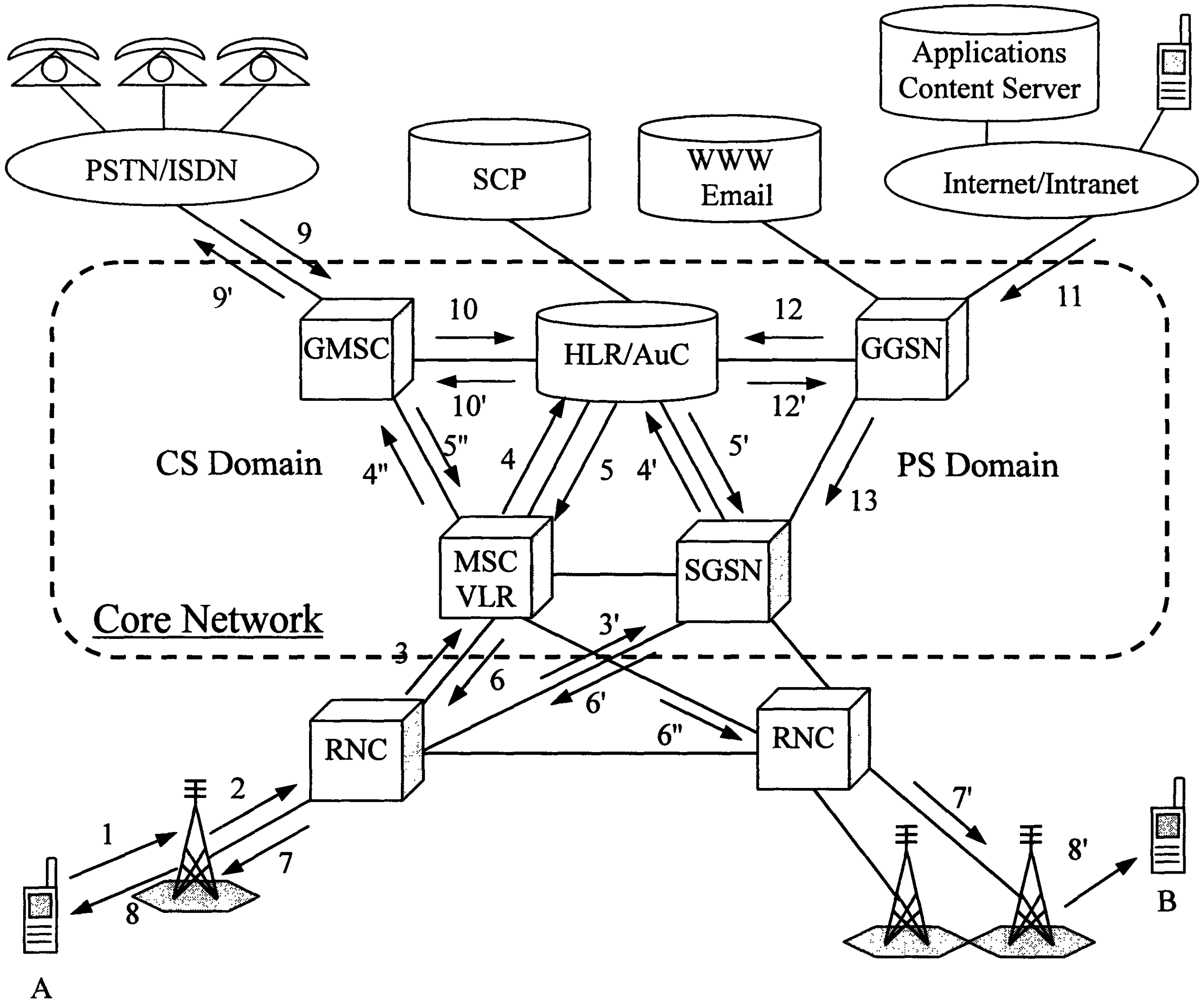
Table 1-1: Network Entities in 3G Systems

No.	Network Entity	Major Functions	Note
1	MSC	Switching services, controlling calls Mobility management for the subscribers	The VLR/MSC are usually implemented in the same node.
2	VLR	Temporary subscriber database	The VLR/MSC are usually implemented in the same node.
3	SGSN	Packet switching Mobility management for the subscribers	DoCoMo's network integrates the SGSN with the VLR/MSC
4	GMSC	Gateway of circuit switching services	Connects to the PSTN and ISDN
5	GGSN	Gateway of packet switching services	Connect to the Internet
6	HLR	Management of the subscriber database Location management Call handling	The HLR/AuC are usually implemented in the same node.
7	AuC	Authentication of the subscriber	The HLR/AuC are usually implemented in the same node.

Source: Based on 3GPP TS 23.002 V6.6.0 [9], 3GPP TS 09.02 V7.9.0 [10] and 3GPP TS 29.002 V6.8.0 [11].

Figure 1-4 illustrates the core network. The responsibilities of 3GPP core network are the followings:

- Mobility management
- Call connection control between user equipment and the core network
- Core network signaling among the core network nodes
- Inter-working functions between the core network and external networks
- Packet-related functions
- Operation and Maintenance (O&M) functions to maintain the network systems [9][10].



Note: 3GPP2 systems also have similar architecture

Source: This network architecture is based on 3GPP TS 23.002 V6.6.0 [9], 3GPP TS 09.02 V7.9.0 [10] and 3GPP TS 29.002 V6.8.0 [11].

Figure 1-4: Core Network in 3G (UMTS) Systems

- **Location Management (from handset A): 1, 2, 3, 4, 5, 6, 7, 8 (1, 2, 3', 4', 5', 6', 7, 8)**

The main task of location management is to keep track of the subscriber's current location. When mobile handsets connect to a wireless network, the VLR (SGSN) updates the location information in the HLR and stores the subscriber data sent by the HLR.

- **Authentication (from handset A): 1, 2, 3, 4, 5, 6, 7, 8 (1, 2, 3', 4', 5', 6', 7, 8)**

  3G mobile networks utilize a challenge-response mechanism to ensure that only authorized subscribers can access the network. Authentication information is made in the HLR/AuC and sent back to the VLR (SGSN). The VLR (SGSN) verifies the information and enables the service.

- **Call Handling (circuit switched): 9,10, 5-8, 1-4, 10', 9'**

  After receiving the message from the GMSC, the HLR searches the subscriber location in its database (DB) and requires the VLR to provide a roaming number. This routing information is sent back to the GMSC and the GMSC handles the routing.

- **Packet Handling (to handset A): 11, 12, 12', 13, 6', 7, 8**

  After receiving the message from the GGSN, the HLR searches for the subscriber location in its database. This information is sent back to the GGSN and the GGSN sends a confirmation message to the SGSN, which then connects the mobile handset via the RNC.

- **Operation and Maintenance (between VLR/MSC (SGSN) and HLR): 4,5 (4', 5')**

  Several operations are used to maintain consistency between network entities. If the VLR (SGSN) loses the subscriber data, it requests the latest subscriber information from the HLR.

1.5 Thesis Structure

Chapter 1 provides general information about mobile communications and network security (number of subscribers, market share, general security issues, and network architecture). Chapter 2 illustrates the network security mechanisms in 3G mobile systems. Focusing on the network software, Chapter 3 explores the unique characteristics of telecommunication services. This chapter also describes platform principles and provides a framework upon which to analyze platform innovation processes in network software. Chapter 4 analyzes the actual network software to investigate the security mechanism, and examines performance tradeoffs that mobile carriers face when implementing security features. Building upon Chapter 4, Chapter 5 suggests how mobile carriers can manage software development to secure their networks. Chapter 5 examines software development strategies that can reduce future threats. Figure 1-5 provides a graphic illustration of the thesis structure.

In this thesis the major data such as development size, development time, number of dynamic steps and CPU performance, is based on the actual project information.

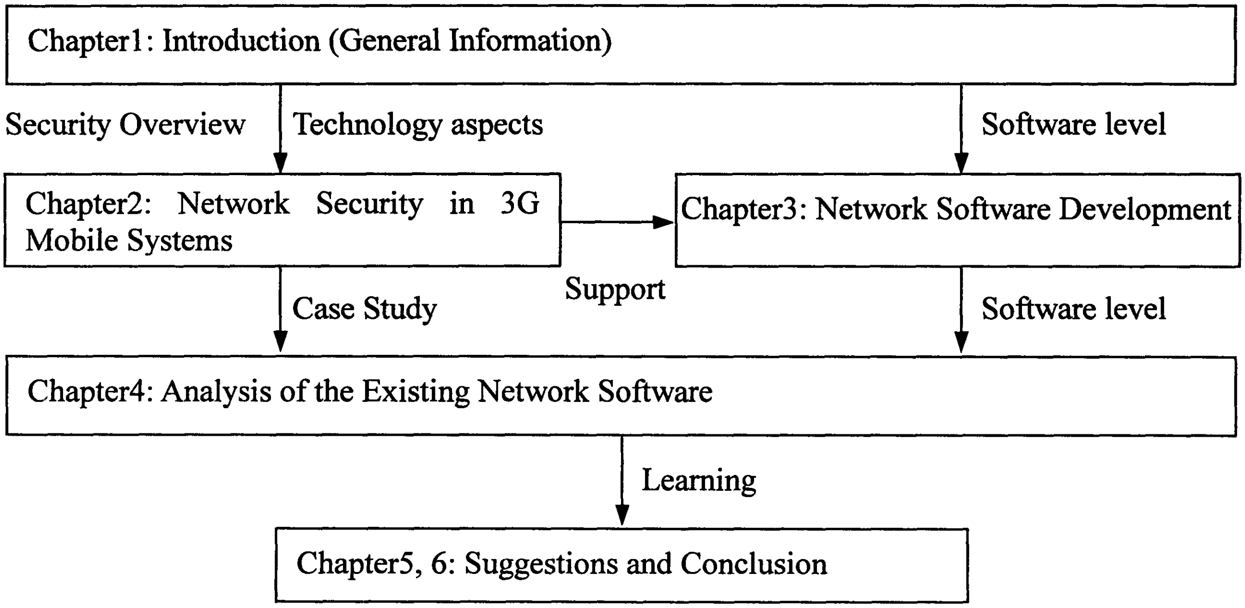

Chapter1: Introduction (General Information)
Security Overview
Technology aspects
Software level
Chapter2: Network Security in 3G Mobile Systems
Chapter3: Network Software Development
Support
Case Study
Software level
Chapter4: Analysis of the Existing Network Software
Learning
Chapter5, 6: Suggestions and Conclusion

Figure 1-5: Thesis Structure

2 Network Security in 3G Mobile Systems

Ideally, network security should prevent unauthorized network access and fraudulent communications. Since the beginning of the first generation systems, there have been discussions about security solutions. The central questions are: how is security defined; how should security architecture be designed; and how should security mechanisms work. As a first step to analyzing the security mechanisms in 3G mobile systems, it is essential to explore fundamental principles of security.

2.1 Security Principles

The fundamental principles of security have remained unchanged since the OECD proposed essential guidelines for security in 1992, which suggested, "the objective of security of information systems is the protection of the interests of those relying on information systems from harm resulting from failures of availability, confidentiality, and integrity [12]." Availability, confidentiality, and integrity comprise the three key security principles of information systems. In other words, a breach of any one of the three principles can have serious consequences for a system. 3G mobile systems define similar security features based on these OECD principles.[2] Although OECD's concepts have already become classic and traditional [13], network systems continue to follow this simple but strong security model.

[2] ISO [24] also defines security as follows: "The capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them."

1) Availability

The characteristic of data, information, and information systems being accessible and usable on a timely basis in the required manner

2) Confidentiality

The characteristic of data and information being disclosed only to authorized persons, entities, and processes at authorized times and in the authorized manner

3) Integrity

The characteristic of data and information being accurate and complete and the preservation of accuracy and completeness

2.2 3G Security Features

The first generation analog mobile systems had few security features to protect the systems and the users. The second generation digital systems incorporated improved security features and contained entity authentication and confidentiality protection. However, it has now been fifteen years since the first GSM system was deployed. Security features have become obsolete and need to be updated to prevent existing and potential security threats. With the advent of 3G mobile systems, a serious effort is underway to create a consistent security architecture based on the threats and risks that 3G systems face [5].

The Universal Mobile Telecommunications system (UMTS) is the newest evolution of the Global System for Mobile Communications (GSM). Therefore, the security of UMTS systems is built on security from GSM, making use of proven GSM security features. UMTS aims to maintain the robustness of existing GSM security features, enhance the existing security mechanisms, and correct perceived weaknesses in 2G systems (see Table 2-1).

Table 2-1: Comparison of 3G (UMTS) and 2G (GSM) Security

	3G Network Security	**GSM Network Security**
Authentication Procedure	Mutual authentication between a mobile handset and network	Unilateral authentication of user to network
Scope	Network to network, RNC	Mainly among base stations
Data Integrity	Explicit	Implicit
Key Length	128 bits	32 bits-64bits
Algorithm	KASUMI, MAC, MILENAGE	COMP 128 (already broken)
Design method	Open design	Closed design
Key Parameters Transmission	Mechanisms to secure key parameters within and between networks	No mechanisms to secure cipher keys and authentication values
Upgrading	Flexible	Inflexible
Fraud Detection	Explicit mechanisms	Implicit mechanisms

Source: Based on Howard, M Walker, T Wright, 2001 [14].

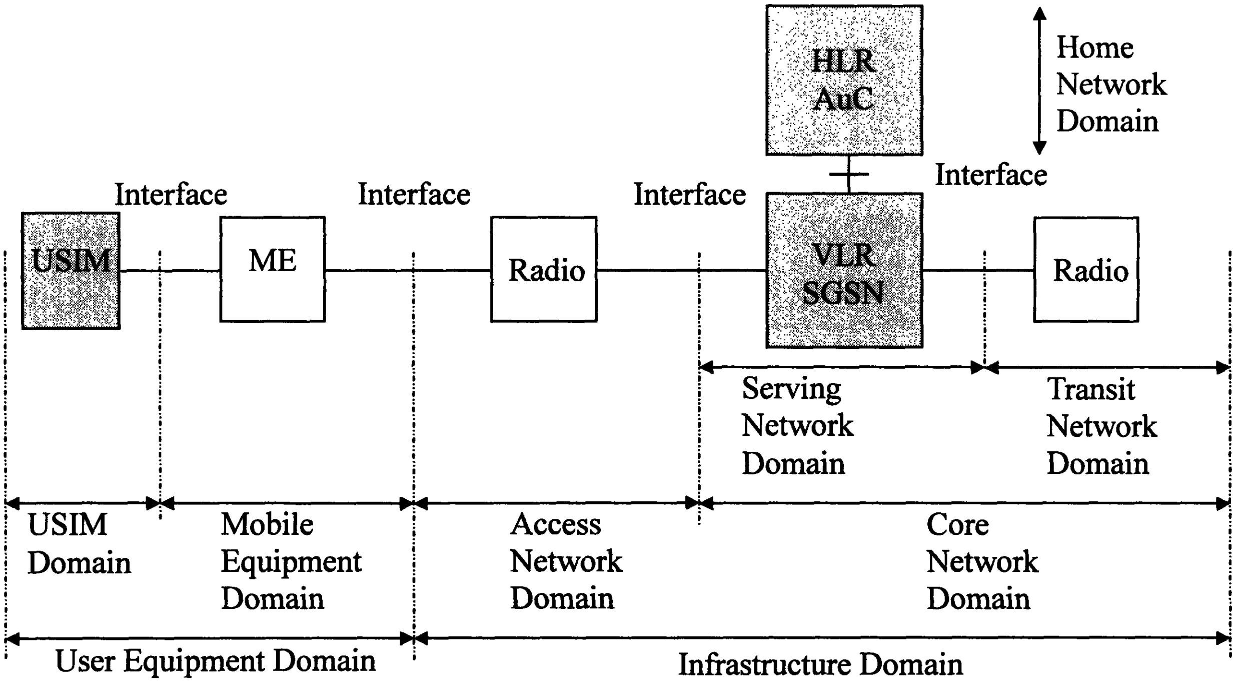
3GPP categorizes possible security threats to 3G systems: (1) unauthorized access to sensitive data (violation of confidentiality), (2) unauthorized manipulation of sensitive data (violation of integrity), (3) disturbing or misusing network services (leading to denial of service or reduced availability), (4) repudiation, and (5) unauthorized access to services. Fraudulent communications mainly result from "unauthorized access to services," but intruders[3] masquerading as users or network entities have the potential to trigger other threats.

2.3 Security Architecture

The UMTS network can be observed from physical entities and logical (protocol-related)

[3] 3GPP defines intruders as follows: a party who attempts to breach the confidentiality, integrity, or availability of 3G, or who otherwise attempts to abuse 3G in order to compromise services or defraud users, home environments, serving networks, or any other party. An intruder may, for example, attempt to eavesdrop on user traffic, signaling data and/or control data, or attempt to masquerade as a legitimate party in the use, provision, or management of 3G services.

aspects. Physical entities are modeled using the domain concept[4] and logical aspects are modeled using the stratum concept.[5] Each domain has its own functions and protocol interface (see Figure 2-1). Communication data is transferred between these domains and security features are defined to protect the data against attacks.



Source: Based on 3GPP TS 33.102 V.6.3.0 [15].

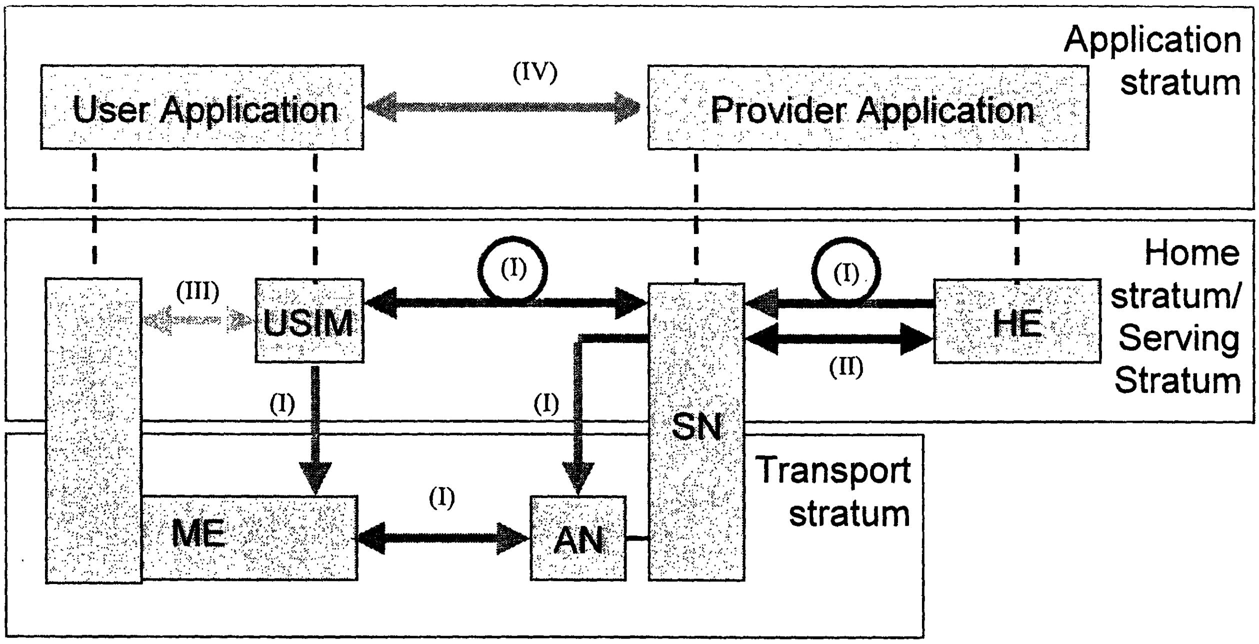
Figure 2-1: UMTS Domains

Five security feature groups are defined in UMTS and each of these feature groups meets certain threats and accomplishes certain security objectives [15]. An overview of the complete 3G security architecture is shown in Figure 2-2. Network access security ensures the user identity, user/network authentication, confidentiality, data integrity, and mobile equipment identification.

[4] Domain: The highest-level group of physical entities.

[5] Stratum: Grouping of protocols related to one aspect of the services provided by one or several domains.

This thesis will focus on network access security between Universal Subscriber Identity Module (USIM), Serving Network (SN), and Home Environment (HE).[6]



Key: AN=Access Network, HE=Home Environment, SN=Serving Network, USIM=UMTS Subscriber Identity Module, ME=Mobile Equipment
Source: 3GPP TS 33.102 V6.3.0 [15], p. 11, Figure 1.

Figure 2-2: Overview of the UMTS Security Architecture

1) **Network access security (I)**

   The set of security features that provide users with secure access to 3G services, and which protect against attacks on the (radio) access link.

2) **Network domain security (II)**

   The set of security features that enable nodes in the provider domain to securely exchange signaling data, and protect against attacks on the wire line network

[6] To simplify the discussion, we can consider that USIM is a mobile handset, SN is VLR/SGSN, and HE is HLR/AuC.

3) **User domain security (III)**

The set of security features that secure access to mobile stations

4) **Application domain security (IV)**

The set of security features that enable applications in the user and provider domains to securely exchange messages

5) **Visibility and configurability of security (V)**

The set of features that enables the user to inform himself whether or not a security feature is in operation, and whether the use and provision of services should depend on the security feature

2.4 Network Access Security Mechanisms

2.4.1 Authentication Procedure

The general procedure for authentication between a mobile handset and the network is illustrated in Figure 2-3. This security mechanism is designed to achieve mutual authentication between the mobile user (handset) and the network by showing knowledge of a pre-shared secret key[7] "K" (128 bits) shared in the USIM and the AuC. This two-way authentication procedure allows UMTS to increase network security compared with GSM by eliminating false base station problems.[8]

K is not transferred in the network between the mobile handset to HLR/AuC, but it is possible to attack the USIM in the mobile handset (especially when retailers register the subscriber information). If K is exposed to intruders, cloned mobile phones become quite possible.

[7] Definitions of these security parameters are given in Table 2-2.

[8] In GSM mobile handsets cannot reject the false base stations, which can unscramble mobile phone calls.

The authentication data contains sensitive information, such as cryptographic keys and random challenge-response codes. Thus, the transfer of authentication data between the HLR/AuC and the VLR/SGSN needs to be secured against eavesdropping and modification.[9]

Authentication Request (No. 1-4 in Figure 2-3)

The basic authentication procedure is carried out between the user equipment (USIM, MS) and the core network (VLR/SGSN and HLR/AuC). User-oriented transactions, such as call setup and location update, initiate the authentication procedure. Authentication data request (operation name: Send Authentication Info) is sent from VLR/SGSN[10] to HLR/AuC and AuC generates authentication vectors (AV). At the same time AuC increases the SQN_{HE} and stores the value in its database.[11]

Authentication Response (No. 5-8 in Figure 2-3)

The AV consists of five vectors: a random number (RAND), an expected user response (XRES), a cipher key (CK), an integrity key (IK), and an authentication token for network authentication (AUTN). After receiving these authentication vectors, VLR/SGSN sends two vectors (RAND and AUTN) to the mobile handset and verifies the response from the handset.

[9] In order to protect against fraud, MAPsec (Mobile Application Part security) has been developed by 3GPP

[10] If the call setup is made in the circuit domain, VLR sends the authentication request (Send Authentication Info) to HLR. If the call setup is made in the packet domain, SGSN sends the authentication request (Send Authentication Info) to HLR.

[11] This procedure is not implemented in the DoCoMo's 3G systems.

Verification and Failure Report (No. 9-12 in Figure 2-3)

The mobile handset also checks the MAC[12] and SQN in AUTN to verify the network. If authentication fails in the mobile handset, the error message (Authentication Failure Report) is generated and sent to HLR/AuC. This message is used to analyze fraudulent communications.

Re-synchronization (No. 13-20 in Figure 2-3)

If the mobile handset fails to verify[13] the SQN sent from HLR/AuC, a re-synchronization procedure[14] is initiated to match the counter between SQN_{HE} and SQN_{MS}. HLR/AuC checks the SQN and MAC in the AUTS and resets the SQN_{HE} value to SQN_{MS}. After re-synchronization, HLR/AuC sends "Send Authentication Info ack"[15] to VLR/SGSN, which completes the re-synchronization procedure.

Network Authentication (No. 21-28 in Figure 2-3)

After checking MAC and SQN, the mobile handset computes RES and sends the result to VLR/SGSN, which also compares the RES with the XRES received from the HLR/AuC and finishes the network authentication. If RES differs from XRES, VLR/SGSN sends "Authentication Failure Report" showing the cause of failure. If RES is equal to XRES, VLR/SGSN selects the CK and IK for connection setup. The mobile handset also computes CK and IK and stores the SQN.

[12] Given the real time constraints, 3GPP relies on conventional methods based on MAC.

[13] Unlike MAC failure, SQN failure is basically considered as lost synchronization not as an authentication error.

[14] This procedure is not implemented in DoCoMo's 3G systems.

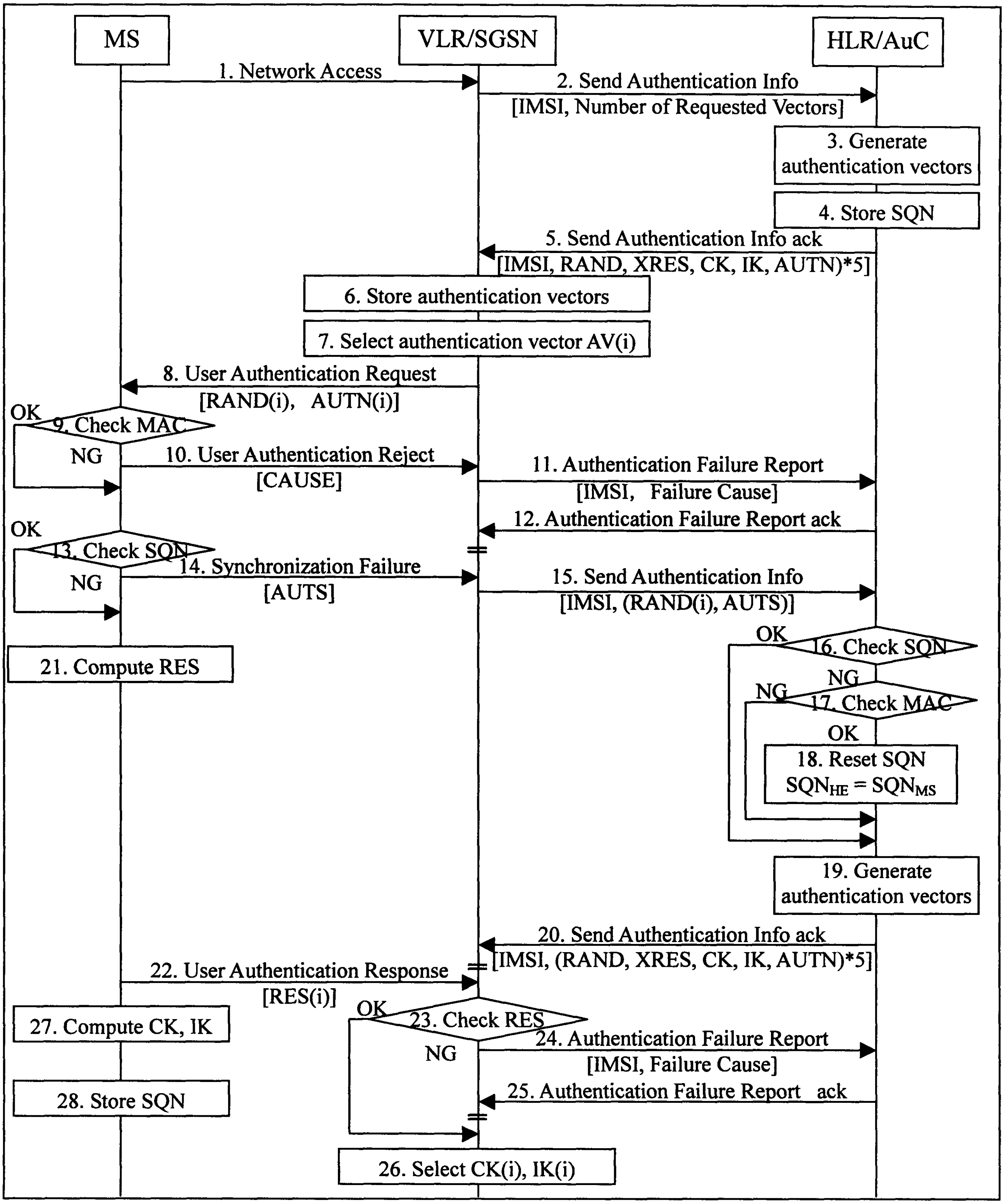
[15] "Ack" means "acknowledgement" and shows the response of the message.

Table 2-2: Definitions of Security Parameters in 3G Systems

	Stand for	Purpose	From/To	Message
K	Encryption key	Encrypt the authentication parameters	- (Only stored in the HLR/AuC and USIM)	- (Only stored in the HLR/AuC and USIM)
RAND	Random challenge	Random number to scramble the authentication message	HLR/AuC to VLR/SGSN VLR/SGSN to MS	Send Authentication Info User Authentication Request
XRES	Expected response	Authentication of the mobile handset	HLR/AuC to VLR/SGSN	Send Authentication Info
CK	Cipher Key	Cipher the data (confidentiality)	HLR/AuC to VLR/SGSN	Send Authentication Info
IK	Integrity Key	Protect the integrity of the control data	HLR/AuC to VLR/SGSN	Send Authentication Info
AUTN	Authentication Token	Authentication of the network	HLR/AuC to VLR/SGSN	Send Authentication Info
SQN	Sequence number	Protect against replay attacks	HLR/AuC to VLR/SGSN MS to HLR/AuC	Send Authentication Info Synchronization Failure
AK	Anonymity Key	Conceal the sequence number	HLR/AuC to VLR/SGSN MS to HLR/AuC	Send Authentication Info Synchronization Failure
AMF	Authentication management field	Support multiple authentication algorithms[16]	HLR/AuC to VLR/SGSN	Send Authentication Info
MAC	The message authentication code	Ensure the authenticity and integrity of the authentication token and the random challenge	HLR/AuC to VLR/SGSN MS to HLR/AuC	Send Authentication Info Synchronization Failure
AUTS	Authentication Token for Re-synchronization	Re-synchronization Authentication of the network	HLR/AuC to VLR/SGSN MS to HLR/AuC	Send Authentication Info Synchronization Failure

Source: Based on 3GPP TS 29.002 V6.8.0 [11], 33.102 V6.3.0 [15].

[16] Other use, such as changing sequence number verification parameters, is also possible.


MS
VLR/SGSN
HLR/AuC
1. Network Access
2. Send Authentication Info
[IMSI, Number of Requested Vectors]
3. Generate authentication vectors
4. Store SQN
5. Send Authentication Info ack
[IMSI, RAND, XRES, CK, IK, AUTN)*5]
6. Store authentication vectors
7. Select authentication vector AV(i)
8. User Authentication Request
[RAND(i), AUTN(i)]
OK
9. Check MAC
NG
10. User Authentication Reject
[CAUSE]
11. Authentication Failure Report
[IMSI, Failure Cause]
12. Authentication Failure Report ack
OK
13. Check SQN
NG
14. Synchronization Failure
[AUTS]
15. Send Authentication Info
[IMSI, (RAND(i), AUTS)]
OK
16. Check SQN
NG
NG
17. Check MAC
OK
18. Reset SQN
SQNHE = SQNMS
21. Compute RES
19. Generate authentication vectors
20. Send Authentication Info ack
[IMSI, (RAND, XRES, CK, IK, AUTN)*5]
22. User Authentication Response
[RES(i)]
OK
23. Check RES
NG
24. Authentication Failure Report
[IMSI, Failure Cause]
25. Authentication Failure Report ack
27. Compute CK, IK
28. Store SQN
26. Select CK(i), IK(i)

Source: Based on 3GPP TS 33.102 V6.3.0 [15].

Figure 2-3: Authentication Mechanisms (Key Agreement)

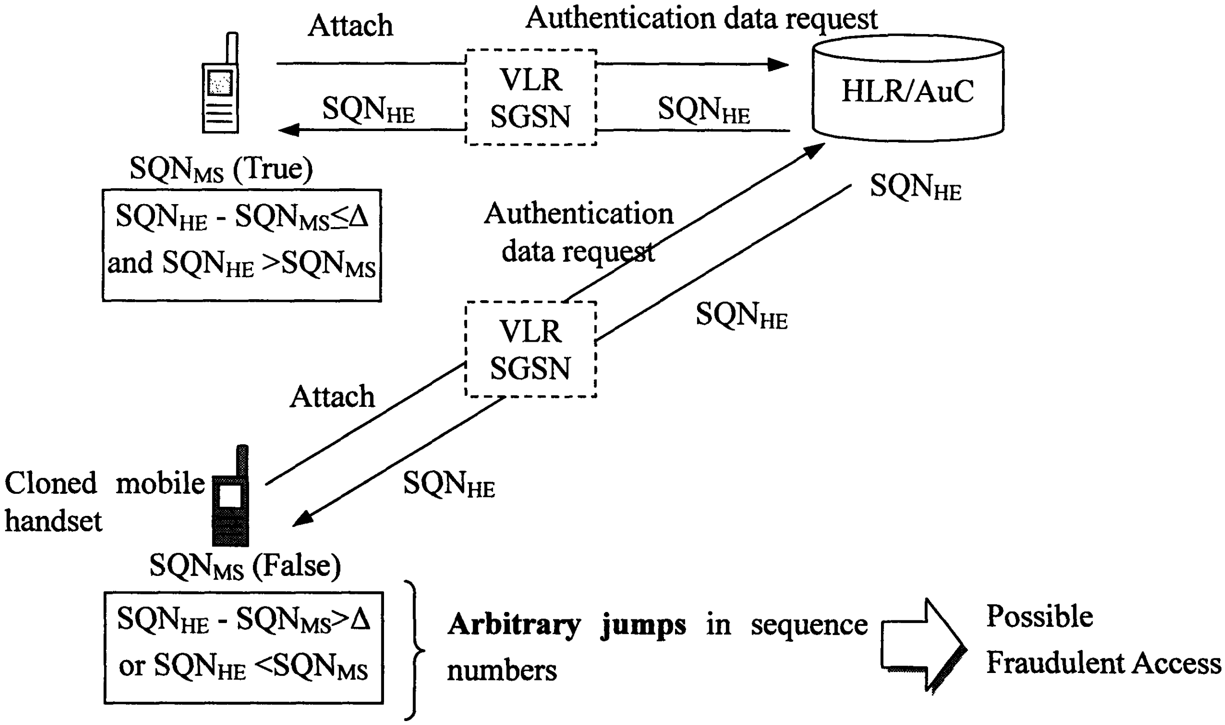
2.4.2 Verification of Sequence Number (SQN) against Replay Attacks

Replay attacks—attacks on the system where messages have been intercepted and then retransmitted (replayed) later—are fierce attacking mechanism that results in masquerading. To overcome this threat, 3GPP technical specifications describe the use of a Sequence Number (SQN). An SQN is the counter (48 bits) possessed by both USIM and AuC to ensure network authentication. The sequence number, SQN_{HE}, is an individual counter for each user stored in HLR/AuC, and the sequence number SQN_{MS} denotes the highest sequence number that the USIM has accepted. In the authentication process, the USIM and the HLR/AuC keep track of counters SQN_{MS} and SQN_{HE}, respectively, and compare the SQNs (SQN_{HE} - $SQN_{MS} \leq \Delta$[17] and $SQN_{HE} > SQN_{MS}$). If the received SQN is out of range, a re-synchronization procedure will be initiated to match the counter between SQN_{HE} and SQN_{MS}.

Verification of SQN is one of the essential mechanisms to maintaining network access security. The occurrence of a re-synchronization procedure is treated as lost synchronization and not as an authentication error, but it suggests possible fraudulent access.

Only the true USIM and HLR/AuC know the right SQN. SQN always changes with user transactions, and it is almost impossible for intruders to copy the SQN in real-time transactions. Any arbitrary jumps in sequence numbers can mean possible fraudulent access (see Figure 2-4).

[17] Mobile carriers can choose the appropriate value of Δ. However, it is recommended that the value of Δ uses 228 in the 3GPP TS 33.102 (Annex C) [15].



Source: Author, 2005.

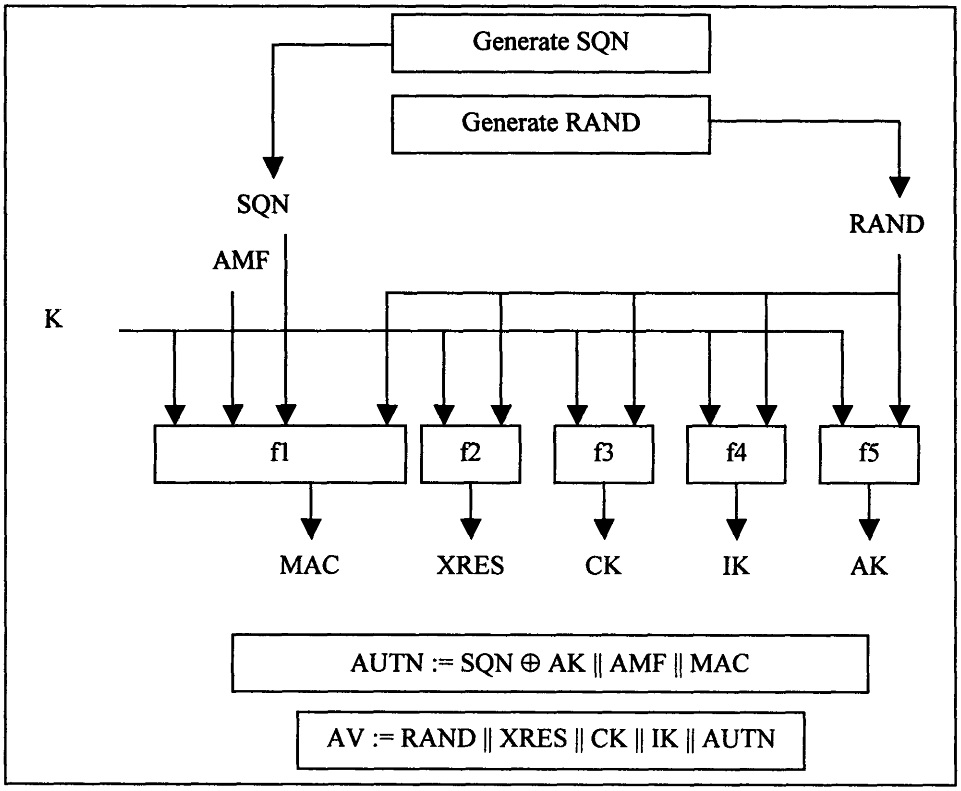
Figure 2-4: Verification of the SQN

The management of SQN is challenging for mobile carriers. Both the time-based and non-time-based SQN generation functions require complex software development. In terms of systems operation, operational difficulty exists in SQN management [16]. HLR/AuC has to store the latest SQNs for each subscriber and implement real-time backup functions for SQN. A crash in the database systems causes tremendous amount of re-synchronization procedures in the network, which results in severe network congestion.

In addition, mobile carriers have to execute the re-synchronization procedure when the USIM detects that the sequence number is not in the correct range. The re-synchronization procedure should not occur too frequently for performance reasons. Verification of the freshness of the sequence number helps mobile carriers detect potential fraud. However, it is costly and takes time to implement and manage this function. In fact, DoCoMo simplified this mechanism in order to reduce software complexity, development time, and development costs.

2.4.3 Generation of Authentication Vectors

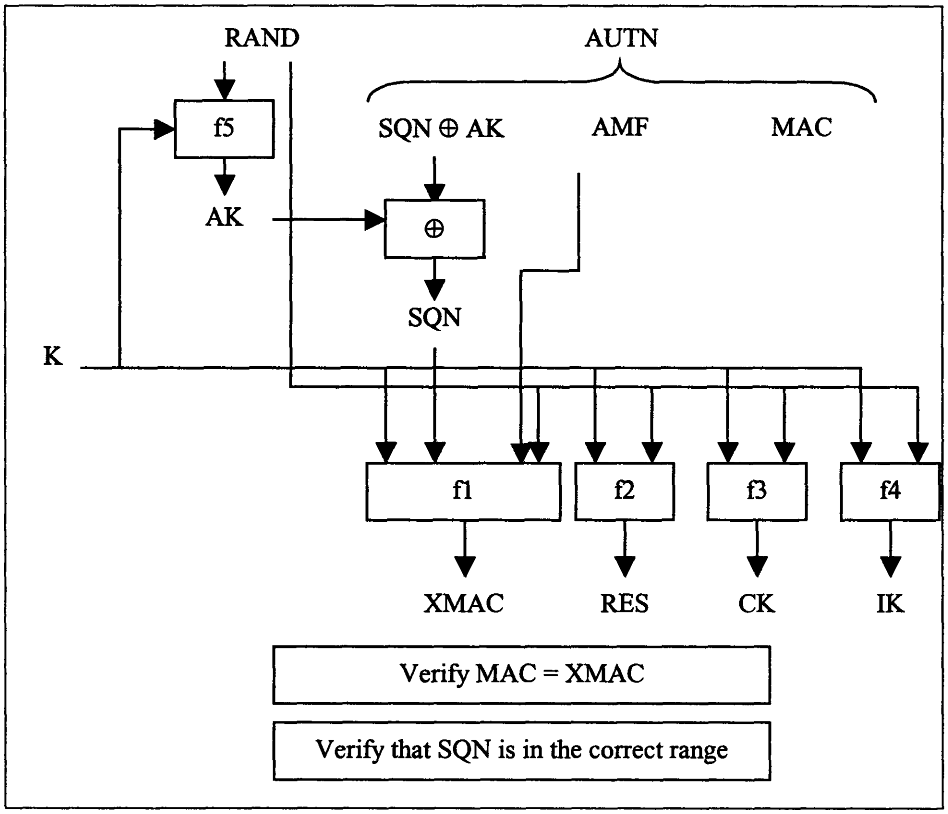
The generation of authentication vectors promises the central authentication functions to secure the system. The HLR/AuC starts by generating a fresh SQN as well as an unpredictable challenge RAND (see Figure 2-5). Deriving K from the subscriber database and applying SQN, AMF, and RAND to the cryptographic functions (f0-f5), the HLR/AuC generates MAC, XRES, CK, IK, and AK. AUTN is also created from SQN, AK, AMF, and MAC. Consequently, an authentication vector (AV) with RAND, XRES, CK, IK, and AUTN is produced in the HLR/AuC. The HLR/AuC repeats this procedure up to five times per authentication request.


Generate SQN
Generate RAND
SQN
RAND
AMF
K
f1
f2
f3
f4
f5
MAC
XRES
CK
IK
AK
AUTN := SQN ⊕ AK || AMF || MAC
AV := RAND || XRES || CK || IK || AUTN

Source: 3GPP TS 33.102 V6.3.0 [15], p. 20, Figure 7.

Figure 2-5: Generation of the Authentication Vectors (AuC)

After receiving the authentication vectors from HLR/AuC, the USIM in the mobile handset checks the received value (see Figure 2-6). Applying the K in the USIM and extracting the RAND and AUTN, the USIM calculates the XMAC, RES, CK, and IK. In the mobile handset, the USIM verifies the MAC and the range of the SQN. The cryptographic functions (f0-f5*) are used in this Authentication and Key Agreement (AKA) procedure and these functions are exclusively implemented in the USIM and AuC (see Table 2-3).



Source: 3GPP TS 33.102 V6.3.0 [15], p. 22, Figure 9.

Figure 2-6: User Authentication Function (USIM)

Table 2-3: Cryptographic Functions in UMTS

Algorithm	Purpose/Usage	O: Operator Specific S: Fully Standardized	Location
f0	Random challenge generating function	O	AuC
f1	Network authentication function	O – (MILENAGE)	USIM and AuC
f1*	Re-synchronization message authentication function	O – (MILENAGE)	USIM and AuC
f2	User challenge-response authentication function	O – (MILENAGE)	USIM and AuC
f3	Cipher key derivation function	O – (MILENAGE)	USIM and AuC
f4	Integrity key derivation function	O – (MILENAGE)	USIM and AuC
f5	Anonymity key derivation function for normal operation	O – (MILENAGE)	USIM and AuC
f5*	Anonymity key derivation function for re-synchronization	O – (MILENAGE)	USIM
f6	MAP encryption algorithm	S	MAP nodes
f7	MAP integrity algorithm	S	MAP nodes
f8	UMTS encryption algorithm	S – (KASUMI)	MS and RNC
f9	UMTS integrity algorithm	S – (KASUMI)	MS and RNC

Note: MILENAGE and KASUMI are names of security algorithms

Source: Koien, G. M, 2004 [5], p. 10, Table 1.

2.5 Security Issues about 3G Mobile Networks

Securing mobile systems sufficiently remains a challenging issue because (1) for performance reasons, 3G mobile systems have to rely on conventional security methods, (2) both secured and unsecured networks are interconnected by global roaming services and (3) it is still difficult to upgrade security features to protect against brand-new and unexpected network attacks.

Security mechanisms must perform within short setup time (practically maximum: 15 sec). In order to avoid setup delays, the 3GPP security working group decided to rely on conventional security methods based on MAC functions, which were already in use in GSM and GPRS networks [5]. Applying the conventional challenge response mechanisms instead of using latest security technologies, 3G security has succeeded to reduce the network delays, but this procedure also makes the system less secure.

Backward compatibility also can result in a security hole in 3G networks. Secured and unsecured networks are connected via several services. Unauthorized network access can occur via other networks that support weaker security mechanisms. While data integrity is mandatory in 3G networks, data encryption is not mandatory in the systems. For example, China is one prominent example of a country that does not use encryption in mobile handsets. Some other countries may also turn off encryption due to export restriction reasons. The Wassenaar agreement allows export of handsets with 128-bit encryption, but other network facilities (e.g., RNC) will be subject to Wassenaar restrictions [17]. Although backward compatibility between different carriers is not mandatory, commercial demands for new services, such as global roaming, force mobile carriers to implement this feature.

3G security builds on the success and lessons learned in GSM systems. Therefore, the scope is limited to well-known security issues. In order to enhance security features and deal with future security threats, 3G security is designed to expand security features. Practically, however, it is very challenging to improve security features. Alternating the security functions requires architectural changes[18] in both the mobile handsets (embedded in IC chips) and the HLR/AuC. Compared with GSM, 3G has more flexibility, but upgrading network software and renewing USIM cards (over 17 million cards) are not an easy task.

Furthermore, the 3G technical specifications, including network architecture, network protocols, and security algorithms, are widely available to the public. 3GPP also provides sample source codes and simulation data for security algorithms. On the basis of the Kerckhoffs's law [18], open-design architecture helps create better security standards, but at the same time it may help intruders find security holes in the system. Potential intruders also have the opportunity to investigate the software closely to identify its vulnerabilities. Given that most of the network attacks are based on software flaws and design errors, an inadequate system design will make the mobile systems vulnerable to unauthorized access to mobile communications.

[18] Sometimes platform extension is applied and sometimes platform renewal is crucial. We will discuss this topic in Chapter 3.2.

3 Network Software Development

3.1 Requirements for Network Software

Software Engineering: The establishment and use of sound engineering principles (methods) in order to obtain economically software that is reliable and works on real machines (Bauer, 1972 [19]).

Software Engineering: The practical application of scientific knowledge in the design and construction of computer programs and the associated documentation required to develop, operate, and maintain them (Boehm, 1976 [20]).

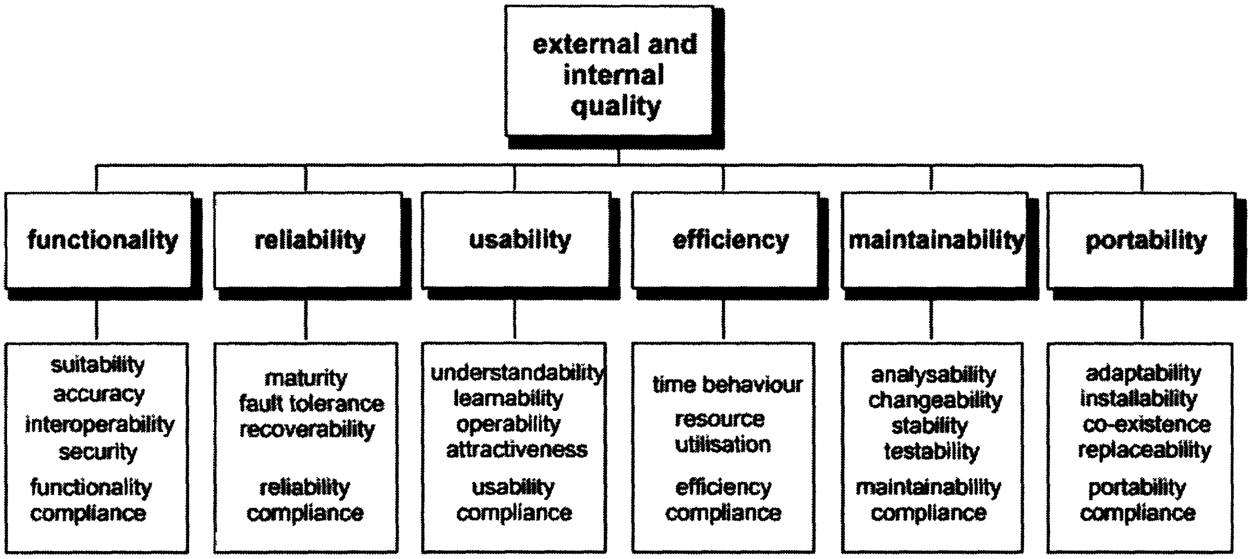
Software engineering is the technological and managerial discipline concerned with systematic production and maintenance of software products that are developed and modified on time and within cost estimates (Fairley, 1985 [21]).

Software engineering is more than just writing and debugging code. As Boehm, Bauer, and Fairley state, the fundamental principles of software engineering include quality, cost (economics), delivery, and the application of knowledge and discipline. These definitions indicate that software engineering should create high-quality software in a systematic, controlled, and efficient manner. Under the quality models identified in ISO9126 and ISO9126-1 (see Figure 3-1), network software[19] development strongly requires such fundamental principles.

ISO's quality attributes are global standards that allow us to comprehensively evaluate software quality. The comparison of network software with other application software suggests that network software differs in the areas of functionality, reliability, efficiency and maintainability (see Figure 3-2). In particular, requirements for reliability, such as maturity (perfection = zero defects) and availability, distinguish network software from other application software [22][23]. Network software must guarantee the critical functions of telecommunications

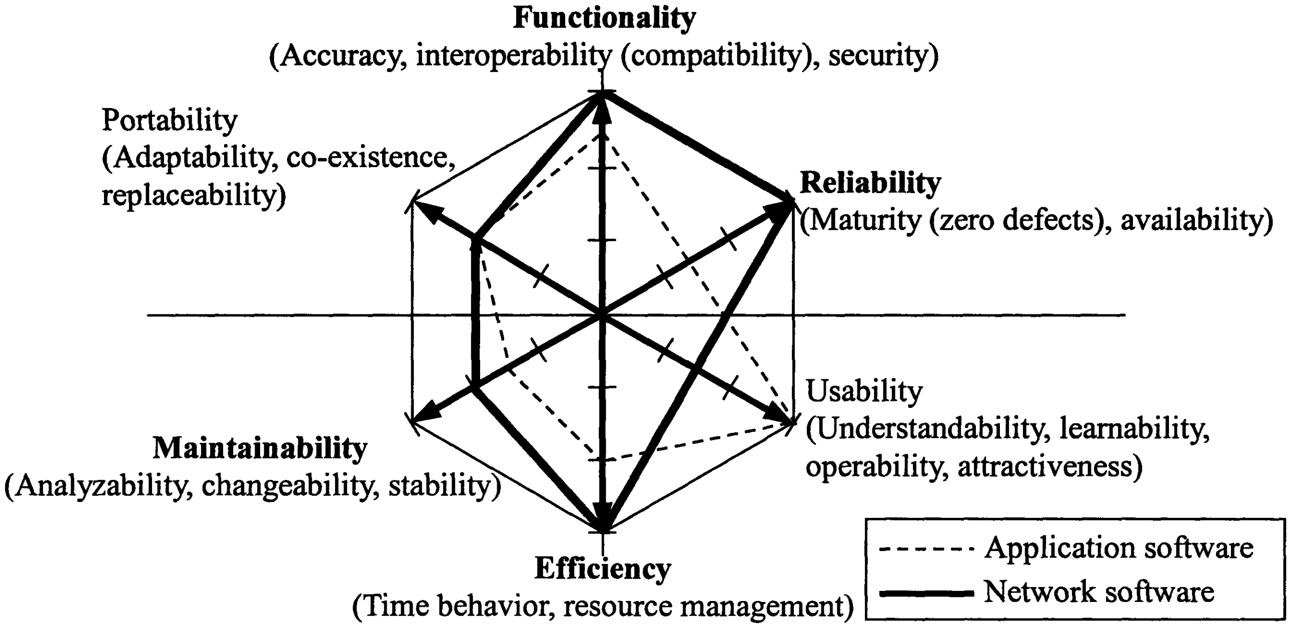
[19] Network software means telecommunications software, but network software provides not only telecommunications but also more sophisticated and value-added communication services. Security features work on network software to protect against attacks.

systems. Thus, reliability is one of the most essential requirements in network software. On the other hand, usability, such as understandability and attractiveness, is considered less important than other quality attributes because network software is designed to satisfy specific needs of telecommunications carriers rather than end users.



Source: ISO/IEC, 9126-1, 2001 [24], p. 7, Figure 4.

Figure 3-1: Quality Model for External and Internal Quality



Note: technical terms in use are based on ISO9126-1 [24]

Source: Author, 2005.

Figure 3-2: Comparison in Quality (Application Software and Network Software)

On the basis of the existing ISO standards, we shall now take a closer look at specific quality attributes that required in network software development.

3.1.1 Reliability (Maturity and Availability)

Every industry in the world depends on some form of telecommunications. Along with energy, financial services, transportation, and vital human services, telecommunications systems are part of the nation's critical infrastructure. Each critical infrastructure is increasingly interdependent on other complex systems. In addition, telecommunications infrastructure itself forms a complex interconnected networked system that stretches over a large geographical area to reach every household and economic entity in a region. Failure of the network software (e.g., a local disturbance in one system) causes large-scale failure via cascading events that can have unexpected and catastrophic consequences. For example, emergency services, such as calls to police, ambulance, and firefighters, also depend on telecommunications systems. If the telephone network goes down, even for a short time, most businesses and individuals could experience severe consequences.

Telecommunications systems must be significantly robust [25]. In theory, a switching system should be available for all but two hours within a 40-year period [26]. Today, telecommunications systems are required to support up to 5 or 6 nines (or 99.999 to 99.9999%) reliability, which translates to between 30 seconds (6 nines) and 5 minutes (5 nines) of downtime per year [27].

However, maintaining the high reliability is a quite challenging issue in network software development. Telecommunications systems are subject to hidden failures—hardware or software failures that only become apparent when a system or some portion of a system is highly stressed due to congestion or fault. In other words, hidden failures are typically not revealed before the system is perturbed. Furthermore, in every transaction, received messages, subscriber information and execution timing vary in the same environment. Thus, it is quite difficult to examine and fix the software failures in operation [28].

New features also increase behavioral complexity [29] and might destroy existing functions (called feature interaction problems [30][31][32]) in telecommunications systems. Thus, new features are carefully designed and tested for "unfailing reliability" before upgrades are implemented. Early detection of failures, flexibility in the architecture (easy upgrading), and redundancy (fault tolerance system) are keys to minimize the serious effects of hidden failures and achieve unfailing reliability.

3.1.2 Security

In every transaction, network software manages sensitive subscriber data including telephone number (MSISDN), identification number (IMSI), time, and location information. Thus, network software should be designed to protect against unauthorized network access, manipulation of data, and repudiation of services. Advanced mathematical algorithms, such as cryptographic functions (f0-f5 in Table 2-3), are applied to ensure the authorized users and prevent interception. These security features make fraudulent communications impossible, and communication messages traveling across networks cannot be obtained or read by anyone other than the authorized users.

In addition to this direct authentication mechanism, network software should facilitate a threat/risk analysis, which identifies imminent and potential risks in 3G systems. Specific threats, such as fraudulent network access and masquerading as another user, are analyzed to eliminate as much as possible threats to the network. Protocol analysis functions help investigate questionable message flows. For example, if HLR receives a number of "Authentication Failure Report" messages (failure messages in user authentication) from VLR/SGSN, it is an indication of unauthorized accesses to the network.

The accumulated experiences of mobile carriers using first generation (analog) systems and second generation (especially GSM or PDC) systems also help to understand current and future threats to mobile systems. For the threat analysis network software must implement strong risk assessment functions (traffic analysis, data trace, and protocol analysis) to help operators identify fraudulent communications.

3.1.3 Compatibility (Interoperability)

Telecommunications carriers have to maintain compatibility[20] in four ways: (1) between generations in the same carrier, (2) in the same generation in the same carrier, (3) between generations in different carriers, and (4) in the same generation among different carriers (see Figure 3-3).

	Same carrier	Different carriers
Different generation	Support several protocols to maintain the previous functions (1) E.g., PDC and UMTS E.g., PDC and cdmaOne E.g., PDC and CDMA2000	Support several protocols to keep and expand mobility (3) E.g., GSM and UMTS E.g., cdmaOne and CDMA2000
Same generation	Support the same protocols and improve functions/services (2) E.g., PDC + new services E.g., UMTS + new services E.g., CDMA2000 + new services	Support the same protocols to keep and expand mobility (4) E.g., UMTS-UMTS E.g., CDMA2000- CDMA2000

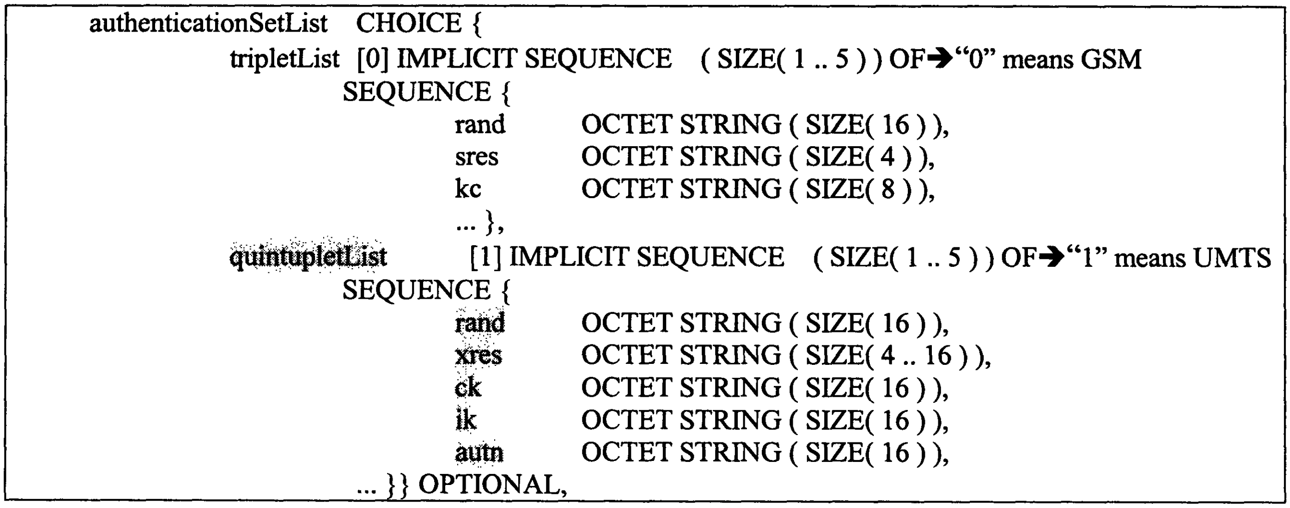
Source: Author, 2005.

Figure 3-3: Compatibility in Telecommunications Systems

[20] In this thesis compatibility means "interoperability."

Among these compatibilities, (3) is the most challenging requirement for network software. Mobile carriers must support at least two protocol versions and verify the availability, confidentiality, and integrity of the services with only limited information. In particular, it is almost impossible to identify the capacity of other networks. Some mobile networks in Europe still support only GSM, while other mobile networks, such as Vodafone and DoCoMo in Japan, provide UMTS services. DoCoMo supports the Extended UniData (XUDT, max= 2048bytes) messages, but other 2G networks only support UniData (UDT, max= 256bytes) messages. In order to achieve compatibility, mobile carriers are often forced to make fundamental changes to the system architecture.

Security functions also have their own versions in the UMTS networks. UMTS defines two versions of application contexts for security functions: (1) infoRetrievalContext-v2 for GSM network (triplet), and (2) infoRetrievalContext-v3 for UMTS network (quintuplet). The message protocol is the same, but mobile carriers have to change the security functions in accordance with the parameter (see Figure 3-4).



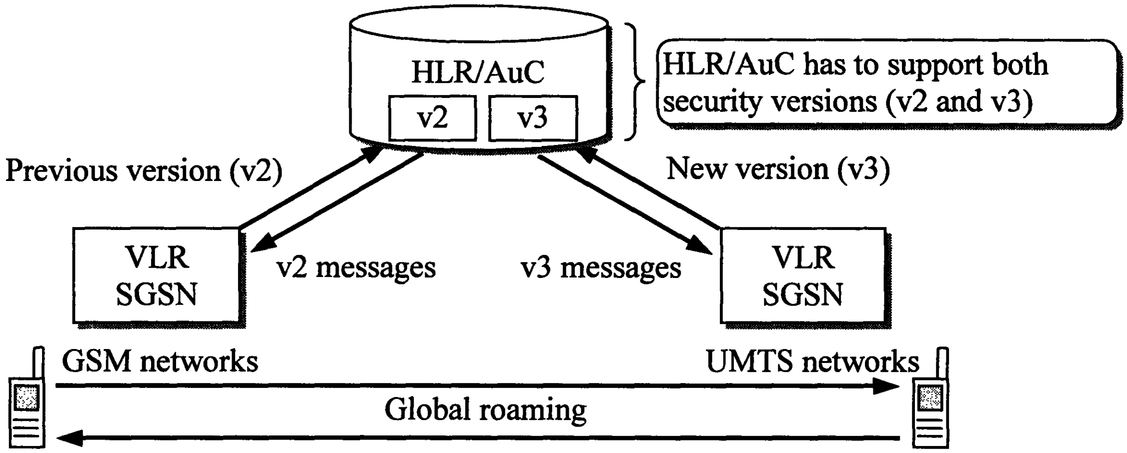
```
    authenticationSetList   CHOICE {
            tripletList  [0] IMPLICIT SEQUENCE   ( SIZE( 1 .. 5 ) ) OF➔"0" means GSM
                    SEQUENCE {
                            rand       OCTET STRING ( SIZE( 16 ) ),
                            sres       OCTET STRING ( SIZE( 4 ) ),
                            kc         OCTET STRING ( SIZE( 8 ) ),
                            ... },
            quintupletList       [1] IMPLICIT SEQUENCE   ( SIZE( 1 .. 5 ) ) OF➔"1" means UMTS
                    SEQUENCE {
                            rand       OCTET STRING ( SIZE( 16 ) ),
                            xres       OCTET STRING ( SIZE( 4 .. 16 ) ),
                            ck         OCTET STRING ( SIZE( 16 ) ),
                            ik         OCTET STRING ( SIZE( 16 ) ),
                            autn       OCTET STRING ( SIZE( 16 ) ),
                    ... }} OPTIONAL,
```

Source: Based on 3GPP TS 29.002 V6.8.0 [11].

Figure 3-4: Authentication Set List (Message Protocol: Send Authentication Info)

Much of the work with the UMTS access architecture has been focused on backward compatibility with GSM/GPRS networks. From a security viewpoint, however, backward compatibility with GSM networks is undesirable because they support weaker security mechanisms than UMTS. Such backward compatibility can result in critical security holes in 3G mobile systems.

Although backward compatibility between different carriers is not mandatory, commercial demands for new services, such as global roaming, are key features of 3G systems. Thus, mobile carriers cannot ignore these compatibility requirements (see Figure 3-5).



Source: Author, 2005.

Figure 3-5: Backward Compatibility in Security functions

3.1.4 Real-time Response (Time Behavior)

Telephony is real-time application software that requires the system to respond immediately to its requests even in overload situations.[21] The generic requirements for switching

[21] Overload condition means that CPU use exceeds 70%.

systems specify deadlines for various responses, such as high-call throughput and low-call setup delays [33][34][35]. In DoCoMo's 3G systems, the performance requirement is 600 transactions/second.[22] In particular, security functions require 2-10 times higher performance than other management functions,[23] such as location management and supplementary service functions. The security message (Send Authentication Info) reaches 540 octets in length, which is the second largest message in the whole MAP protocol. In addition, traffic involving "Send Authentication Info" is estimated to constitute 17.2% of total traffic[24] between HLR and VLR/SGSN. In order to manage huge volume of transactions, many carriers apply the C/C++ programming language.

System performance is important, but mobile carriers must concentrate on both system reliability and performance. If they focus only on system reliability, the network software will require so many error-detecting functions that eventually system performance will deteriorate. On the other hand, if network engineers focus on only system performance, the software will lack architectural design and finally lose configurability and reliability. Under the short development time, it is challenging to fulfill both requirements of performance and reliability. The requirement for real-time response makes it difficult to develop the system design and implement new versions.

3.1.5 Configurability for Geographical Distribution of the System (Maintainability)

In today's switching systems, a call may be distributed among several sites. Heterogeneous clusters of nodes cooperate with each other to set up and complete the call. Network software must not only cover general routing functions but also operate differently in its own environment.

[22] This is the requirement for HLR.

[23] This data is based on the DoCoMo's project.

[24] This data is based on the DoCoMo's network.

Thus, software engineers must design simple network software configurations to manage the network systems.

Network software for the HLR is distributed to all of the HLRs in 3G systems; network software for the MSC, VLR, and SGSN[25] is delivered to all of the MSCs, VLRs, and SGSNs. For example, the number of the HLRs in DoCoMo's network exceeds 140. The total number of CPUs is over 500. Furthermore, a duplex system is applied in each system. In Japan, 91 million calls are generated every day[26] among mobile handsets. A single critical defect in the network software will create hundreds of defects across the entire network and prevent communication services.

In the United States, AT&T's long-distance telephone switching centers crashed on January 15, 1990. A single software defect in the switching center (switching relays) caused cascading failures in the distributed networks. Sixty thousand people lost their telephone service completely for nine hours [36]. Such a disaster shows that broad geographical distribution of the system often magnifies the negative effects of software defects. Configurability is an essential requirement for limiting failures in the entire network.

3.2 Platform Innovations in Network Software

Time-to-market pressure, and requirements for high quality and cost reduction, are driving software development toward more disciplined architecture design styles [37][38]. In the quest to improve flexibility and manage complex systems, firms in many industries are considering platform-based product development [39][40][41]. Two keys to this approach are: (1) the sharing

[25] The MSC, VLR, and SGSN are often integrated in the same network node. In this case only one network software carries out the MSC/VLR/SGSN functions.

[26] Telecommunications Carriers Association (TCA), 2002.

of components (modules[27]) and (2) other function blocks (subsystems[28]) across a family of products.

Like physical products, software consists of myriad subsystems and modules that are connected to each other via numerous interfaces. Thus, the platform-based approach is quite effective in software development. Historical success stories such as the Sony Walkman [42][43], and Microsoft's Windows NT [44] Intel's microprocessors [45] have demonstrated the benefits and the logic behind the platform concept. Platform thinking—the process of identifying and exploiting commonalities among services, target markets, and the processes for creating and delivering offerings—appears to be a successful strategy to create new services at low costs.

In terms of product lifecycle, thinking about platforms for families of products rather than individual products is a key driver behind the success of short-cycle-time companies. In fact, the software development cycle in network software has shortened from two years to a half year. Today, mobile carriers are trying to shorten the software lifecycle from six months to three months. A clear gap between platform concept and practical issues still exists, however, when it comes to designing, testing, implementing, and managing product families and their successive platforms [46].

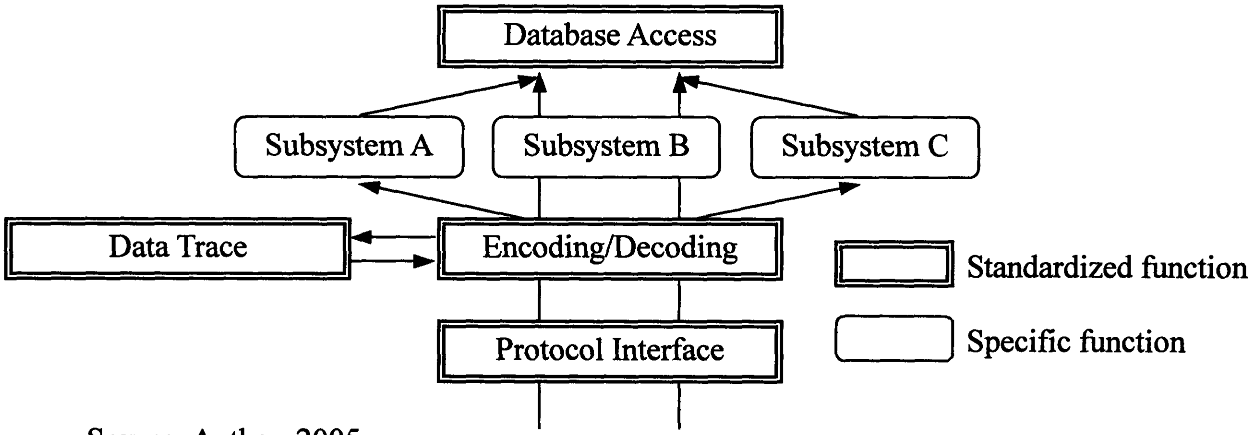
As a first step to filling the gap between platform concept and practical issues, it is essential to review major platform thinking. The literature addresses a variety of concepts related to platform thinking: component standardization, architectural innovation, product architecture, product platform, and product family [8][38][47][48][49][50][51][52][53].

[27] Module is the smallest unit in the software (10-1000 lines).

[28] In this thesis, "subsystem" means "function block." A subsystem consists of several modules.

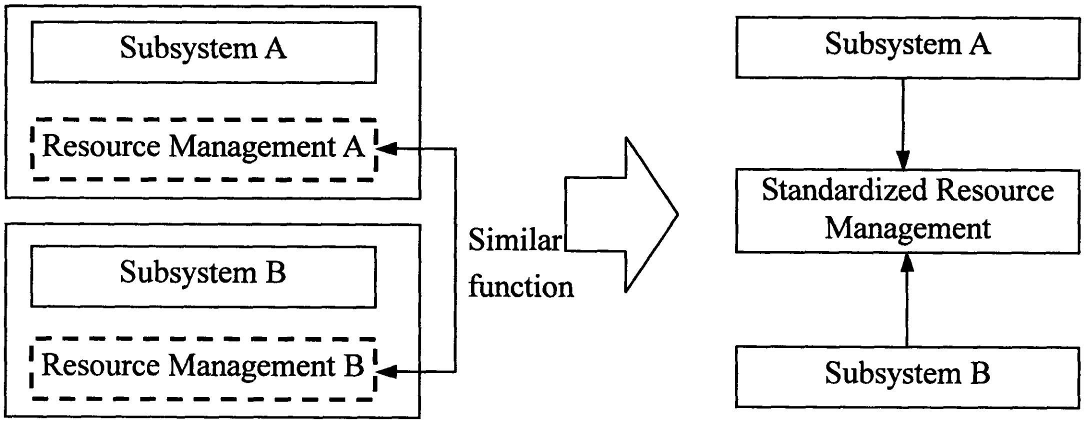
3.2.1 Component Standardization

Standardizing the software components enables the use of the same module or subsystem in multiple subsystems. The use of standard components can lower the complexity, cost, and lead time for product development. Standardized modular systems also provide the ability to achieve product variety through combining and standardizing components [54][55]. Standardization can occur only when (1) a component contains commonly useful functions, and (2) the interface of the component is identical across more than one product [38].



Source: Author, 2005.

Figure 3-6: Component Standardization in 3G Systems (DoCoMo)



Source: Author, 2005.

Figure 3-7: Component Standardization after the First Release (DoCoMo)

Several common functions, such as database access, encoding/decoding, and data trace functions were standardized in DoCoMo's network software (see Figure 3-6) during the design phase of the architecture. After releasing the first version, DoCoMo applied component standardization in order to implement the common resource management function (see Figure 3-7). This standardization took about a year to accomplish, but it allowed DoCoMo to enhance the software resource efficiency[29] and implement more complex message operations.

3.2.2 Architectural Innovation and Product Architecture

Henderson and Clark found that the traditional categorization of innovation as either incremental or radical is incomplete and potentially misleading [52]. The authors define an architectural innovation as "innovations that change the way in which the components of a product are linked together, while leaving the core design concepts (and thus the basic knowledge underlying the components) untouched" (p.10). They stress that "The essence of an architectural innovation is the reconfiguration of an established system to link together existing components in a new way" (p.12). In the network software development, modifications among several subsystems to create new features can be considered an architectural innovation. For example, if DoCoMo decides to implement new SQN management mechanisms, several subsystems have to change interfaces and modify internal functions without changing core design and concept.

In terms of implementation, the authors argue that architectural innovations may create organizational resistance or inertia and tend to hinder the successful adoption of future architectural innovations. In short, once a dominant platform architecture has emerged, the

[29] About 25% of the reduction in memory use

operating platform and accumulated organizational standards come to reflect the core concept of the product and create organizational resistance toward the change. The central idea is given in Figure 3-8.

		Core Concepts	
		Reinforced	Overturned
Linkages between Core Concepts and Components	Unchanged	Incremental Innovation	Modular Innovation
	Changed	Architectural Innovation	Radical Innovation

Source: Henderson, R., Clark, K. B, 1990 [52], p. 12, Figure 1.

Figure 3-8: A Framework for Defining Innovation

Ulrich [38] defined product architecture as (1) the arrangement of functional elements, (2) the mapping from the functional elements to physical components, and (3) the specification of interfaces among interacting physical components. He expands on this definition using several examples and applies it to software development. In a modular architecture, components have one or few functional elements (one-to-one mapping) and interfaces among components are well specified. In an integral architecture components show a complex (non one-to-one) mapping between functional elements and components and interfaces among components are not well defined. The main argument is that product innovations are linked to the architecture of the product (see Figure 3-9).

Several scholars view a modular architecture as ideal. Alexander [56] presents that an optimal design methodology can be achieved by avoiding coupling between components. Suh

[57] argues that a modular architecture is an axiom of good design that can avoid manufacturing/design failures. Meyer and Lehnerd [8] point out that achieving modularity while minimizing the number of interfaces between subsystems is the essence of elegance in software design. Baldwin and Clark [58] describe that modularity in design can tremendously boost the rate of innovation.

Software development also requires modularity and flexibility. In the software engineering, the notion of module cohesion or strength can be considered the one-to-one mapping of functional elements to components [59]. Modular architecture is expected to increase the reuse rate and to avoid complex interfaces among components. Modularity also helps localize the modification of software and enhances software quality. Hac [60] verified that architectural dependencies between components and degree of parallelism in the components affect the software reliability. On the other hand, integrated architecture with unspecified interfaces among components is not desirable for localizing implementation risks.

Product Architecture	Modular	- Variety achieved by combinatorial assembly from relatively few component types. - Can assemble to order from component inventories. - Minimum order lead time dictated by final assembly process.	- May fabricate components to order as well as assemble to order. - May choose to carry component inventories to minimize order lead time. - Infinite variety is possible when components are fabricated to order.
	Integral	- High variety not economically feasible; would require high fixed costs (e.g. tooling), high set-up costs, large order lead times, and/or high inventory costs.	- Variety can be achieved without relatively high inventory costs by fabricating components to order. - Minimum order lead times dictated by both component fabrication time and final assembly time. - Infinite variety is possible.
		Low	High

Component Process Flexibility

Source: Ulrich, K, 1995 [38], p. 430, Figure 8.

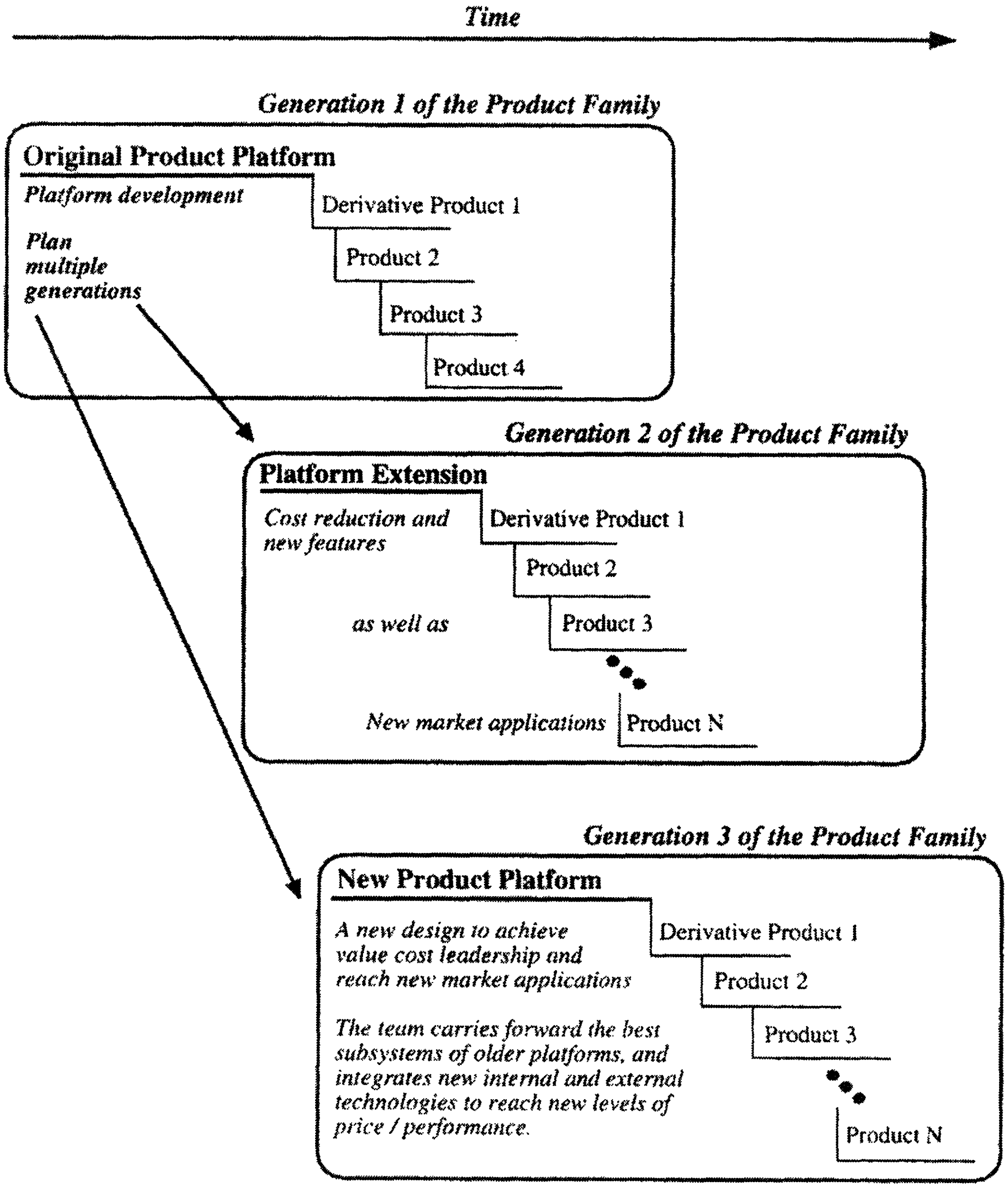
Figure 3-9: Product Architecture and Component Process Flexibility

3.2.3 Product Platform and Product Family

Meyer [8] defines product platform as "a set of subsystems and interfaces that form a common structure from which a stream of related products can be efficiently developed and produced." A product platform is often defined in terms of physical components, but a product platform also can be defined in terms of software. The product platform is the basis for developing new product variants. Several variants can be found in network software. For example, the security function for the GSM network is a variant of the original security function for the UMTS network. The algorithms applied to generate authentication vectors differ, but the basic architecture of the subsystems is quite similar.

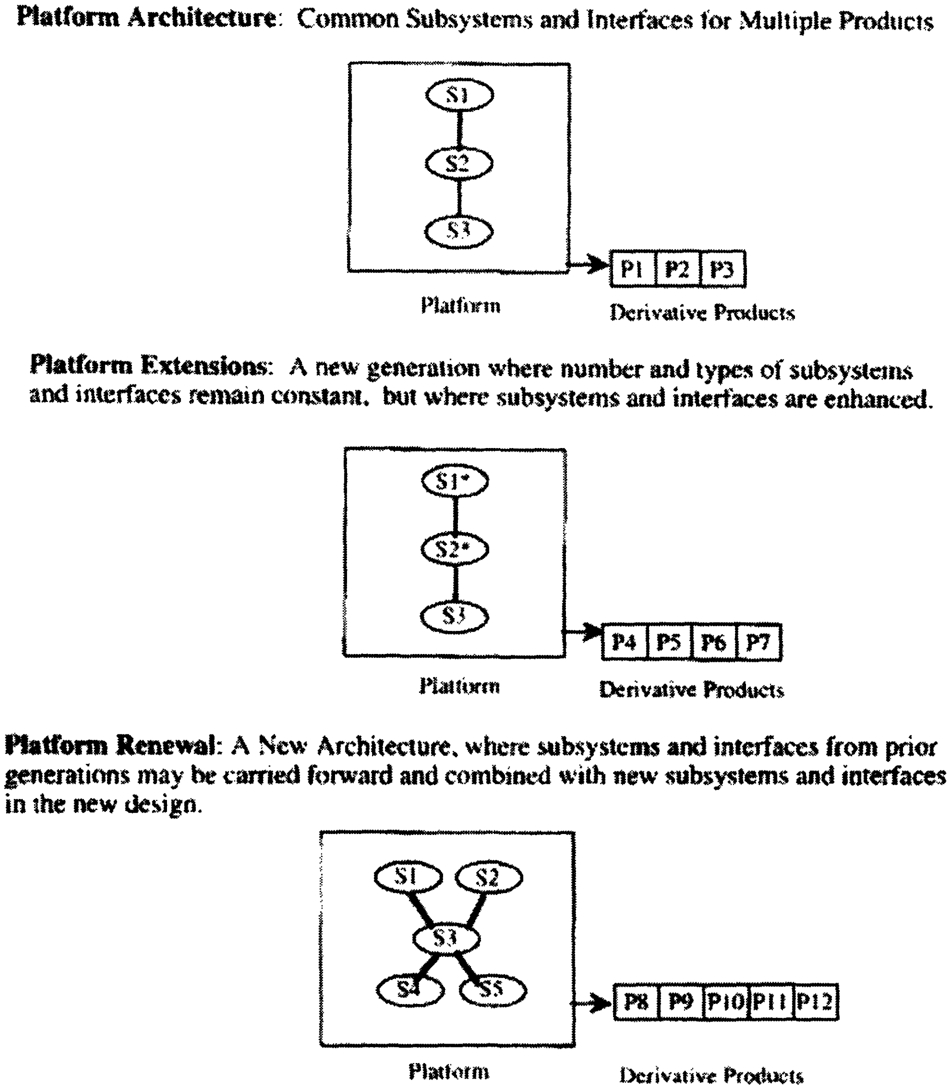
Several authors [8][48][53] define "family" as individual products that share common technology and address related market applications. According to Simpson [50], a product family is a group of related products that share common features, components, and subsystems, and satisfy a variety of market niches. A product family comprises a set of variables, features, or components that remain constant from product to product (product platform) and others that vary from product to product [50]. The distinctive aspects between individual product variants are the difference in their structure. Meyer and Lehnerd [8][48] propose a general framework for product family development (see Figure 3-10) that represents a single product family beginning with the initial development of a product platform. This platform is followed by successive major enhancements to the core product and process technology of that platform, with derivative product development within each generation. New generations of the product family can be based on either an extension of the product platform or on an entirely new product platform. In case of an extension, the group of subsystems and interfaces remains constant.

However, one or more subsystems sometimes must undergo major revision to reduce costs or to add new features. An entirely new platform emerges only when its basic architecture changes and aims at value cost leadership and new market applications (see Figure 3-11). Systems and interfaces from prior generations may be carried forward into the new design but are joined by entirely new subsystems and interfaces [8][51].


Time
Generation 1 of the Product Family
Original Product Platform
Platform development
Plan multiple generations
Derivative Product 1
Product 2
Product 3
Product 4
Generation 2 of the Product Family
Platform Extension
Cost reduction and new features
as well as
New market applications
Derivative Product 1
Product 2
Product 3
Product N
Generation 3 of the Product Family
New Product Platform
A new design to achieve value cost leadership and reach new market applications
The team carries forward the best subsystems of older platforms, and integrates new internal and external technologies to reach new levels of price / performance.
Derivative Product 1
Product 2
Product 3
Product N

Source: Meyer and Lehnerd, 1997 [8], p. 36, Figure 2-4.

Figure 3-10: Product Family Evolution



Platform Architecture: Common Subsystems and Interfaces for Multiple Products

Platform Extensions: A new generation where number and types of subsystems and interfaces remain constant, but where subsystems and interfaces are enhanced.

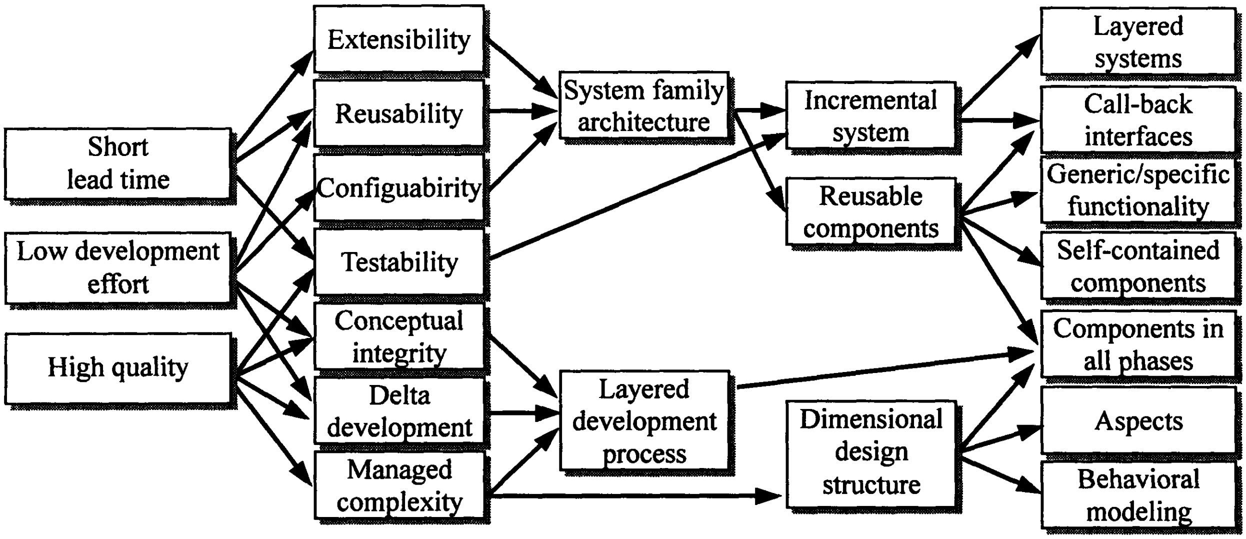
Platform Renewal: A New Architecture, where subsystems and interfaces from prior generations may be carried forward and combined with new subsystems and interfaces in the new design.

Source: Meyer and Lopez, 1995 [51], p. 298, Figure 1.

Figure 3-11: Typology of Platform Change in Product Family Evolution

3.3 Difficulties in Platform Innovation

A variety of requirements can prevent platform innovations of network software. Van Der Linden and Müller [61][62] illustrate the architectural requirements for software (see Figure 3-12). Quality (high quality), cost (low development effort), and delivery (short lead time) are the fundamental software requirements, and these elements significantly affect other architectural requirements: extensibility, reusability, configurability, and so on. Each requirement interconnects and sometimes conflicts with others (e.g., quality vs. cost). In addition, market demands for flexibility, accountability, and robustness also complicate software requirements, creating trade-offs and potential difficulties for platform innovation.



Source: Van der Linden and Müller, 1995 [61], p. 52, Figure 1.

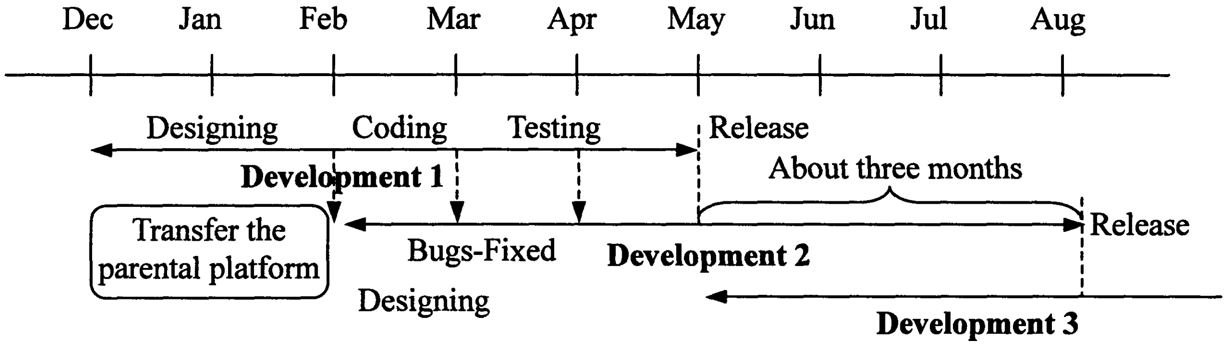
Figure 3-12: Requirements for Network Software

3.3.1 Time-to-Market Pressure

Market demands for new services and severe competition in the mobile telecommunication market create strong pressure to shorten development time. The lead time for platform upgrading has deceased from two years to six months in DoCoMo's system. In today's market, mobile carriers are now trying to shorten the software life cycle from six months to three months. In this situation, mobile carriers are forced to develop several versions of software at the same time (see Figure 3-13).

Three problems lie in this software development timeline: (1) the new version is inevitably based on the imperfect platform of the previous version, (2) the new version is expected to fix all the defects found in the previous version, and (3) architectural change is almost impossible.

In particular, security functions are fundamental features of the platform. The new platform architecture needs sufficient testing to confirm performance and reliability. Under the traditional development process (i.e., the rigid waterfall model) it is quite challenging to alter the software architecture in such a short time. Today's insufficient development time prevents platform innovation and renders the platform architecture less reliable and flexible.



Source: Data is based on the firms' Annual Reports and personal interviews [63].

Figure 3-13: Software Development to Satisfy Short Development Time

3.3.2 Pressures for Cost Reduction

Cost reduction pressures also hinder platform innovations. As Voas mentions, it is quite difficult to achieve better (high quality) and cheaper software in a limited development time [64]. The total budget for software development is also strictly limited. For example, DoCoMo has a budget limit for each software development project (¥5-10 billions) and the company slashes development costs (estimates) by 10-15% across the board. In addition, DoCoMo requires partner developers, such as NEC, NTT Comware, and Fujitsu, to reduce their development costs from ¥10,000 to ¥7,000 per line. In order to reduce development costs, these developers are forced to implement less functional and less flexible (less modularized) functions. In terms of security functions, the generation and management of SQN functions are insufficiently implemented to cut development costs and to meet the delivery deadlines.

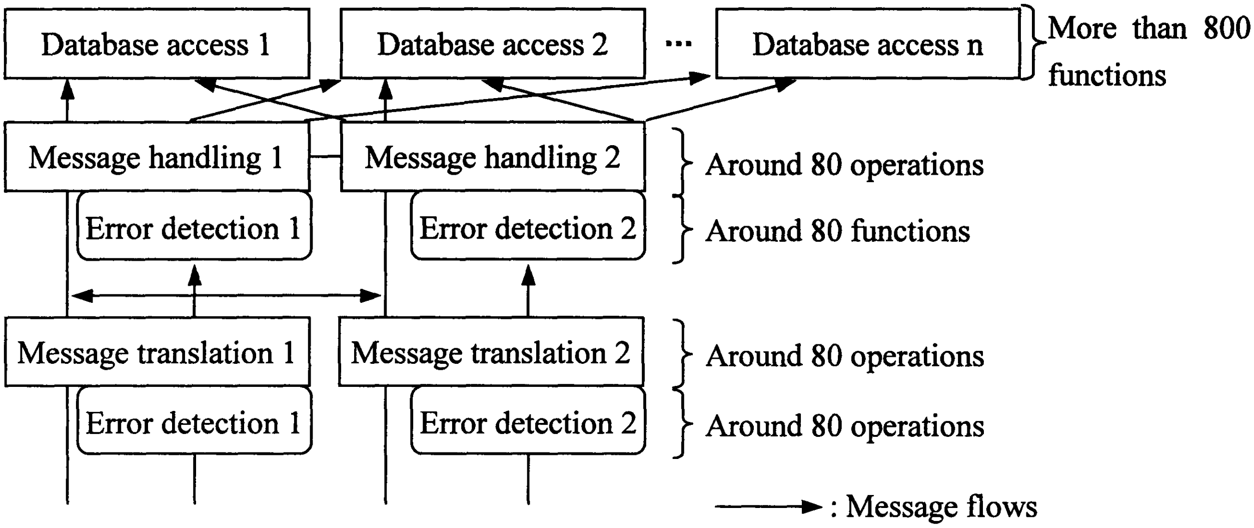
Security is part of the fundamental functions in the platform. Software modifications in the fundamental functions are considered too risky, and such modifications require enormous development time and cost. In addition, unlike other new services, the security function itself does not generate profits, so mobile carriers have less desire to upgrade the security mechanisms. Once security functions are implemented, platform renewal is quite challenging and almost impossible within limited budgets.

3.3.3 Requirements for High Reliability

Network software operates in real-time telecommunications systems. Critical errors immediately produce disastrous outcomes worldwide. Therefore, reliability is crucial in any new platform. However, compared with the existing platform, a new platform is less stable. Before innovating a platform, network engineers must fix all software problems and improve software quality. Most engineers do not have time to change the platform architecture. Ironically, after fixing these problems, network engineers also cannot readily innovate the platform architecture. The debugged platform is considered more stable than a brand-new platform. The requirement for high reliability limits architectural platform innovation and results in low extensibility of the software.

3.3.4 Complex Architecture

The architecture of network software is very complex. A system is comprised of myriad small modules and millions of lines of code. Approximately 80 operations[30] are defined in 3G networks, and each operation has several versions in order to satisfy backward compatibility. Message translation (encoding/decoding) functions, message handling functions (scenario control), and error detection functions (checking for tags, length and values) are individually designed in the system. From the viewpoint of database access, access interfaces (select, update and delete) are developed for each database element. Consequently, over 1,000 modules co-exist in the network software. As the number of supported services increases, the software architecture and module interfaces become more complex. Extensibility and testability are lost, and platform innovation in network software becomes very difficult.



Source: Author, 2005.

Figure 3-14: Complex Architecture and Interfaces in Network Software

[30] The operations are designed for Mobile Application Parts (MAP). Mobile carriers often implement own operations to satisfy specific requirements in their networks.

Sometimes module architecture is also complex. Time-critical modules, such as message operation and network authentication, are designed to maximize system performance. Unlike other modules, these time-critical modules utilize the knowledge of assembler language and form the performance-oriented architecture. Only a few software engineers can evaluate the minimum buffer size, modify the best data type, and apply the most appropriate methods for the operation.

Among the time-critical modules, security functions (see Figure 2-5, Figure 2-6) are considered some of the most difficult functions in the HLR/AuC. If we investigate the lines of code for the security functions, we will find that these security functions occupy less than one percent[31] in DoCoMo's system. However, we also find that the code is extremely sophisticated and that it is quite hard to change the original architecture.

3.3.5 Scalability

Scalability often prevents platform innovations in network software. In order to satisfy the various requirements for functionality, reliability, efficiency, and maintainability, network software development requires a huge volume of initial development (often over 800 KLOC[32,33]) with highly sophisticated technical expertise in the areas of platform design, protocol, and traffic management. Furthermore, every six to twelve months new features with 200-500KL are added to the network software. Griffeth and Lin observed this trend and mentioned that size of the network software and the number of features are steadily increasing [30]. In DoCoMo's case, the accumulated source code has reached about 2.5 million lines of code (HLR). Consequently,

[31] The security algorithm function (f1) needs only 100 lines of code. Even if we add other security algorithm functions, such as f2-f5, these security functions are less than 2 KLOC.

[32] KLOC means "kilo lines of codes." In this thesis KL also means KLOC.

[33] When DoCoMo implemented its 3G systems in 2001, the total development size of the HLR and the switching center (MSC/VLR/SGSN) reached 835KL and over 1.5ML respectively.

thousands of subsystems coexist in the same platform, and the interfaces among modules are increasingly complex. Given the short development time, new features make it difficult to complete software development and accomplish the platform innovation [65][66].

The aggregate size of the network software continually increases because of new functions (services) and requirements for compatibility and robustness. For example, if a new security function is added, it must consider backward compatibility and numerous semi-normal routes. When a new feature is added to the network node, it can affect millions of lines of code and tens or even hundreds of other features. Because of heterogeneity, network carriers must multiply this effect by the number of different kinds of switches in the network. In order to avoid the large-scale failure caused by software defects, numerous sensitive tests are inevitably required (over 10,000 tests in each subsystem). Many network engineers, therefore, hesitate to make architectural changes in the platform.

4 Analysis of the Existing Network Software

Four years have passed since DoCoMo launched the first 3G mobile systems in Japan. DoCoMo initially started 3G services in large cities, such as Tokyo, Kyoto, Osaka, Kobe, and Nagoya areas, and as of October 2005 3G services were available to approximately 99% of Japan's population. With a maximum downlink speed of 384 kbps—forty times faster than conventional wireless data communications—DoCoMo provides smooth and high-capacity communications for large-volume data such as movie images. Today over 17 million subscribers enjoy DoCoMo's 3G services without serious problems.

Security functions are used in almost all mobile terminal-oriented services (e.g., call setup, location update, and supplementary services). Authorized network access prevents fraudulent communications and maintains systems integrity. However, breaches in the security mechanisms can result in serious consequences in mobile communications.[34] DoCoMo recently announced that no fraudulent communications from cloned mobile handsets had been found in its 3G systems. In order to maintain this desirable situation, the security features must be correctly and reliably designed.

Systems performance is also a critical issue in security management. In the near future[35] DoCoMo plans to migrate completely from its 2G services (PDC) to 3G services (UMTS). Security functions will have to manage over 10 billion authentication requests generated by 50 million subscribers. Performance will become a bottleneck in DoCoMo's 3G systems.

Examining the DoCoMo's project data from 1998 to 2003 and analyzing the actual source codes implemented in DoCoMo's network software, I found that DoCoMo's security

[34] If fraudulent communications comprise even one percent of traffic revenue, the damages are ¥212,316 million /year ($202 million/year).

[35] The original target was in 2006; however, the migration is likely to longer.

mechanisms are insufficiently implemented to prevent fraudulent communications and to support high performance. The platform extension already has reached its limits to solve current implementation problems. Thus, platform renewal will be crucial for the DoCoMo's network software. On the basis of the platform concepts discussed previously, this chapter will analyze the security mechanisms in the existing network software.

4.1 Insufficient Implementation of the Authentication Mechanisms

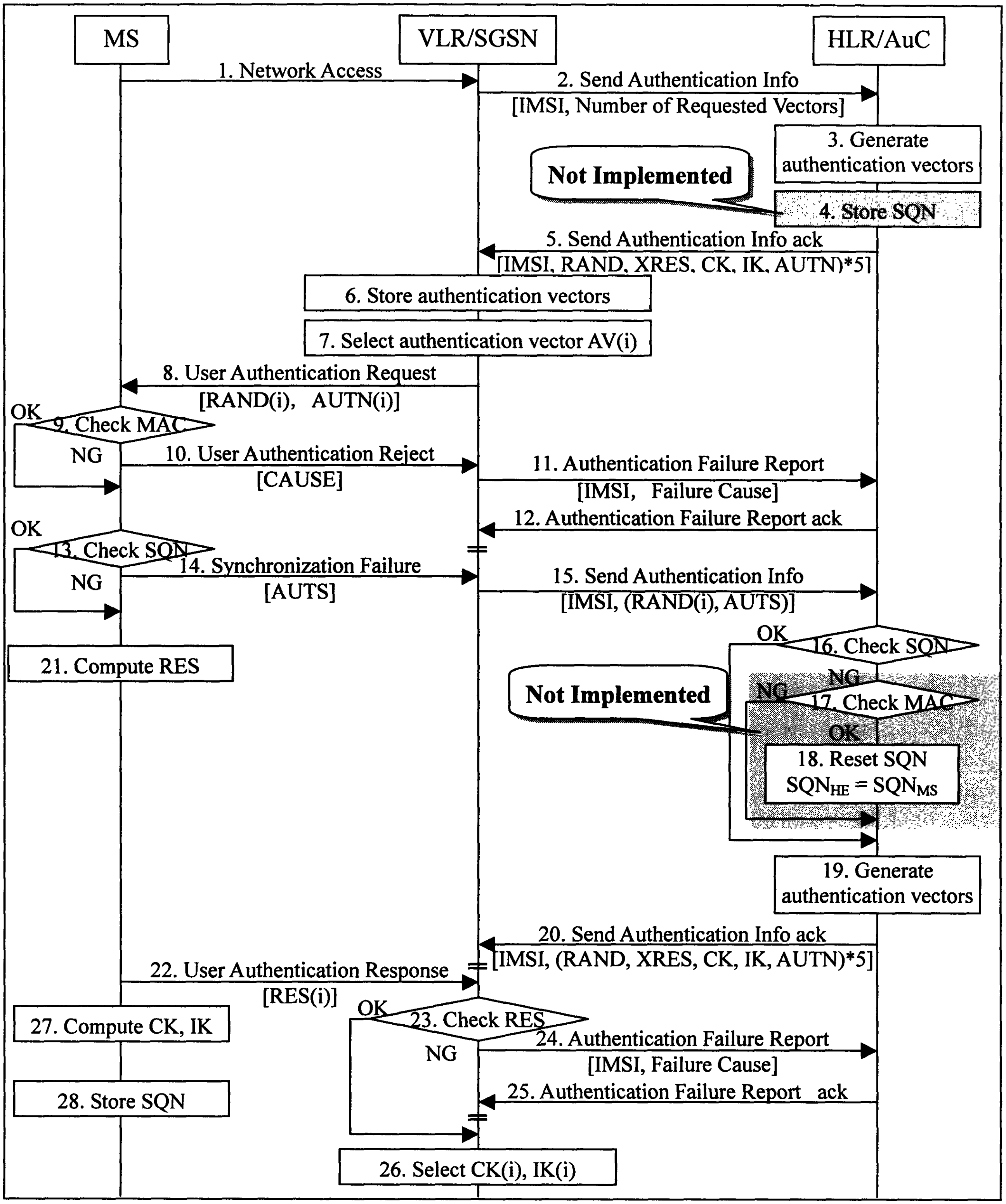
Several security features in DoCoMo's 3G systems differ from the global standards defined by the 3GPP. Table 4-1 and Figure 4-1 show the differences between DoCoMo's implementation and the global specifications. The critical differences are: (1) SQN in the DoCoMo's network has a fixed value and a re-synchronization procedure is not implemented; (2) the network software is designed to support only fixed security parameters; and (3) AMF is not used in DoCoMo's network. In addition, DoCoMo only applies MAP in its network, so the authentication data between the HLR/AuC and the VLR/SGSN can be eavesdropped. The important point is that implemented security features are vulnerable to fraudulent communications (critical security features are missing in the implementation) but no countermeasures have been taken for so far.

The implemented platform architecture also differs from the desired architecture that supports sufficient security mechanisms. The original platform is quite simple and well modularized. If DoCoMo maintains the modularity when implementing new security features, the desired platform will also become modularized. In reality, however, the implemented platform lacks sufficient modularity and several important security features. In order to identify the differences, Figure 4-2 illustrates the original software architecture and the desired software architecture in DoCoMo's systems.

Table 4-1: Differences between DoCoMo's Network and 3GPP Specifications

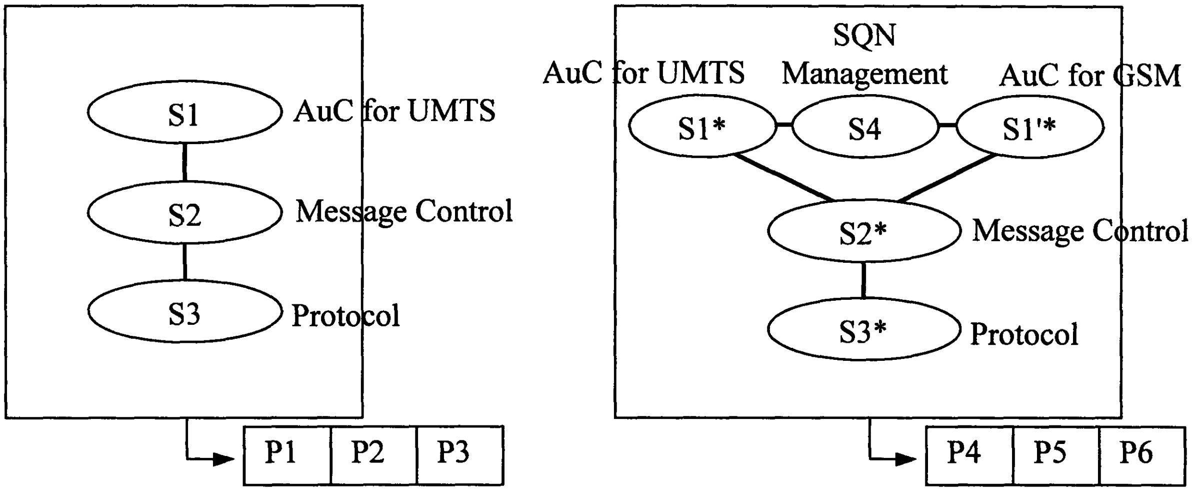
No.	DoCoMo Network	3GPP Specifications (UMTS)	Notes
1	HLR contains AuC in the same hardware and software.	AuC can be separated from HLR.	AuC is integrated as one subsystem in the HLR.
2	The number of requested authentication vectors is always 5.	The number of requested authentication vectors can vary from 1 to 5.	GSM network requests 3 vectors.
3	SQN has fixed value.	SQN always changes.	DoCoMo applies IMSI to create SQN.
4	No re-synchronization procedure is implemented.	Re-synchronization procedure is stipulated.	DoCoMo's HLR does not change the SQN.
5	The size of AUTS is always 16 octets.	The size of AUTS varies from 12 to 16 octets.	DoCoMo only supports the fixed value.
6	The size of XRES is always 16 octets.	The size of XRES varies from 4 to 16 octets.	DoCoMo only supports the fixed value.
7	HLR/AuC does not store SQN.	HLR/AuC stores SQN (SQN_{HE}).	DoCoMo's HLR does not change the SQN.
8	HLR/AuC does not check MAC.	HLR/AuC checks MAC	DoCoMo's HLR reduces the procedure.
9	AMF is not used in authentication procedure.	AMF is used to change the authentication functions.	Top 4 bits= all 0, other 12 bits: do not care
10	Only MAP is implemented to transfer authentication data.	A security extension to MAP called MAPsec is defined	MAP contains no security functionality against eavesdropping.

Source: Author, 2005.


MS
VLR/SGSN
HLR/AuC
1. Network Access
2. Send Authentication Info
[IMSI, Number of Requested Vectors]
3. Generate authentication vectors
Not Implemented
4. Store SQN
5. Send Authentication Info ack
[IMSI, RAND, XRES, CK, IK, AUTN)*5]
6. Store authentication vectors
7. Select authentication vector AV(i)
8. User Authentication Request
[RAND(i), AUTN(i)]
OK
9. Check MAC
NG
10. User Authentication Reject
[CAUSE]
11. Authentication Failure Report
[IMSI, Failure Cause]
12. Authentication Failure Report ack
OK
13. Check SQN
NG
14. Synchronization Failure
[AUTS]
15. Send Authentication Info
[IMSI, (RAND(i), AUTS)]
OK
16. Check SQN
NG
Not Implemented
NG
17. Check MAC
OK
18. Reset SQN
SQN_HE = SQN_MS
19. Generate authentication vectors
21. Compute RES
20. Send Authentication Info ack
[IMSI, (RAND, XRES, CK, IK, AUTN)*5]
22. User Authentication Response
[RES(i)]
OK
23. Check RES
NG
24. Authentication Failure Report
[IMSI, Failure Cause]
25. Authentication Failure Report ack
27. Compute CK, IK
28. Store SQN
26. Select CK(i), IK(i)

Source: Based on 3GPP TS 33.102 V6.3.0 [15].

Figure 4-1: Authentication Mechanisms in DoCoMo's Systems

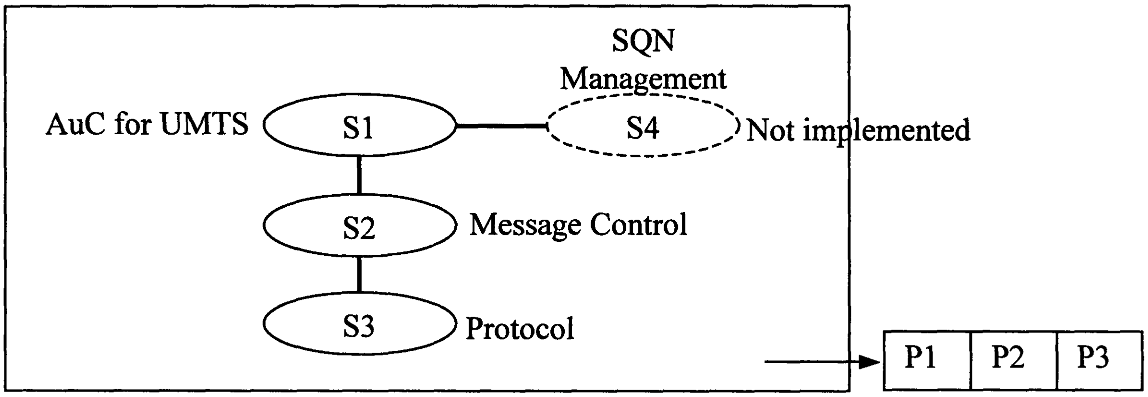


Source: Author, 2005.

Figure 4-2: Comparison of the Software Architecture for Authentication Mechanisms
(The Original Software Architecture (Left) and the Desired Software Architecture (Right))

4.1.1 Insufficient SQN Management

As described in Chapter 2, SQN is applied to protect against replay attacks in the network. Any arbitrary jumps in sequence numbers means possible fraudulent network access. 3GPP specifications suggest several methods for generating SQN: partly time-based, entirely time-based, and not time-based. However, DoCoMo has opted not to implement this mechanism in order to simplify SQN management (see Figure 4-3).

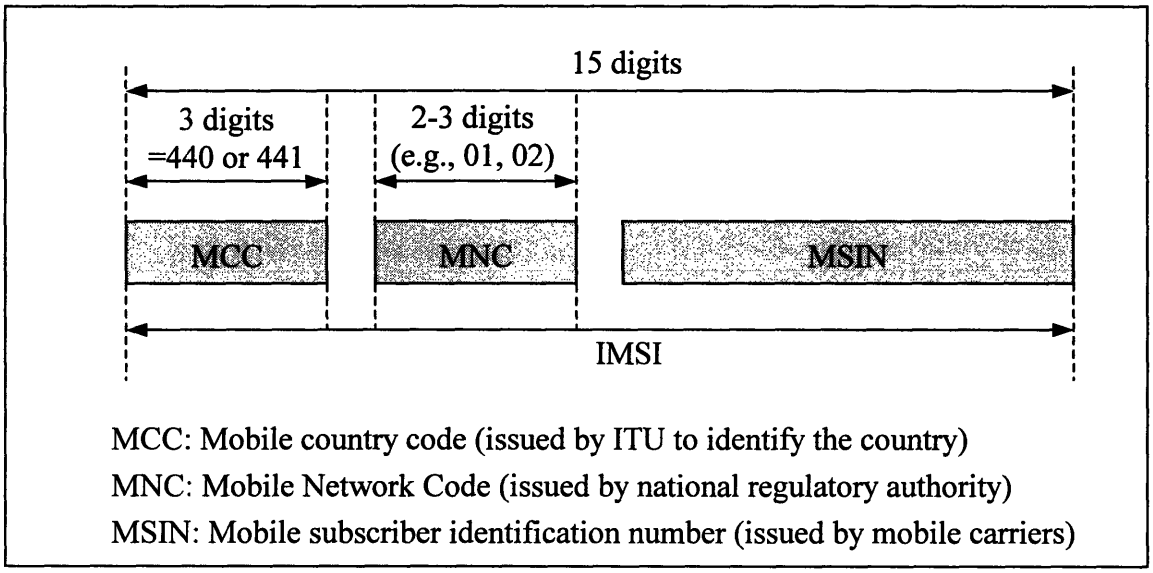


Source: Author, 2005.

Figure 4-3: Software Architecture in DoCoMo's Network (UMTS)

The SQN in DoCoMo's network has a fixed value based on IMSI (see Figure 4-4 and

Figure 4-5), a unique fixed number that identifies a certain subscriber worldwide.[36] Extracting X[37] bits from IMSI, DoCoMo's HLR/AuC generates SQN_{HE}. SQN_{HE} has dummy counters in the top and the bottom of the value, but these counters are not used to verify the SQN. After receiving the authentication parameters, USIM in the mobile handset compares the SQN_{HE} with the same part of the IMSI stored in the mobile handset (SQN_{MS}). The important point is that both SQNs (SQN_{HE} and SQN_{MS}) are generated from the fixed value (IMSI). SQN does not change in the DoCoMo's authentication mechanisms. As long as the mobile handset knows its own IMSI, the SQN comparison is always coincident in the DoCoMo's network (see Figure 4-6). Consequently, HLR and USIM do not have to store and track the value of SQN (SQN_{HE} and SQN_{MS}) in the system.



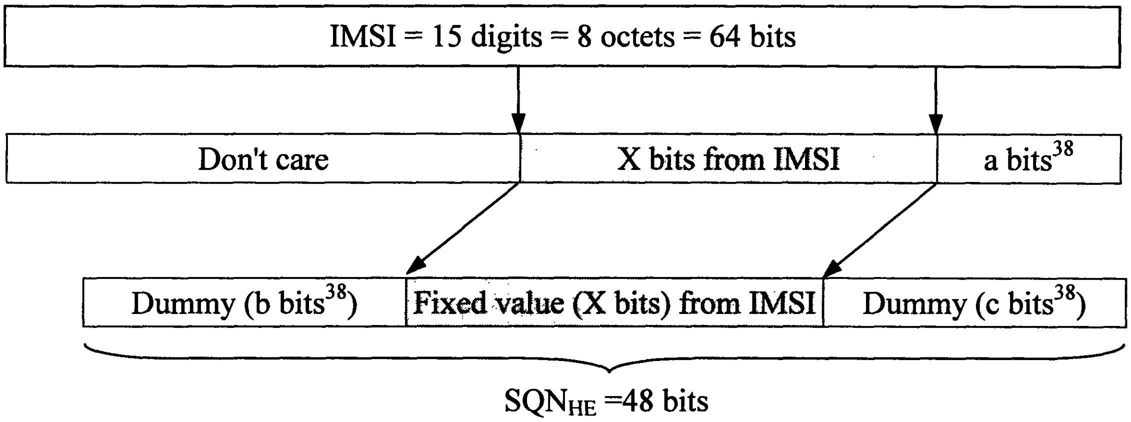
MCC: Mobile country code (issued by ITU to identify the country)
MNC: Mobile Network Code (issued by national regulatory authority)
MSIN: Mobile subscriber identification number (issued by mobile carriers)

Source: 3GPP TS 23.003 V6.5.0 [67].

Figure 4-4: Structure of IMSI in the DoCoMo's Network

[36] IMSI is different from MSISDN (mobile telephone number).

[37] This value cannot be given for security reasons.



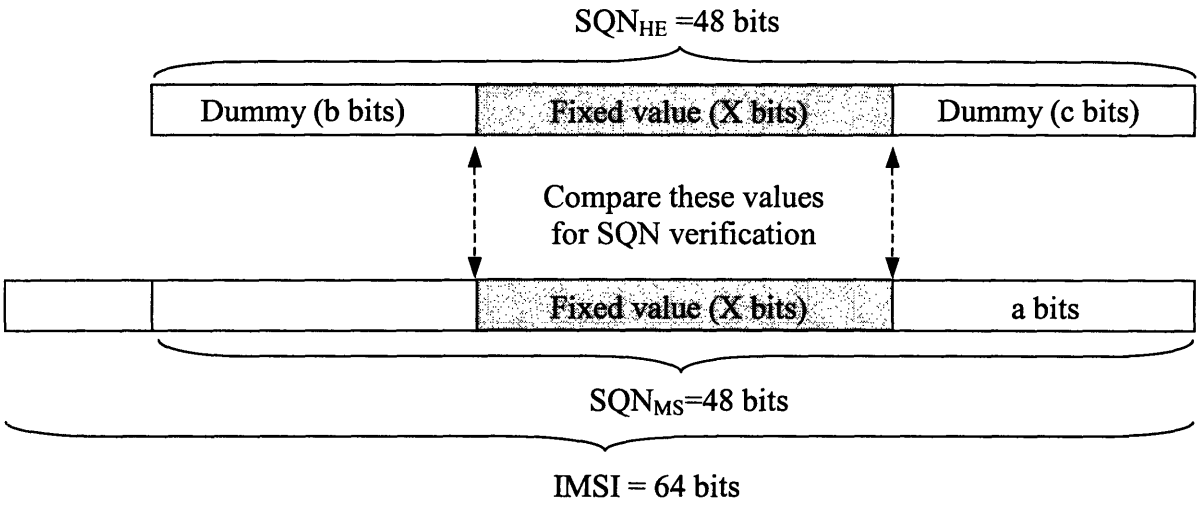
Source: Author, 2005.

Figure 4-5: Generation of SQN in the DoCoMo's Network

DoCoMo also reduced the re-synchronization procedure, which ensures the accuracy and freshness of SQN in the systems. Ideally, re-synchronization procedure can help detect potential fraud. However, as discussed above, the SQN used in the DoCoMo's network always remain the same value. Comparison of SQN (SQN_{HE} and SQN_{MS}) does not provide any significant information. Therefore, DoCoMo decided to omit this re-synchronization procedure completely.

- The generation of SQN is based on a fixed value (SQN comparison is always true).
- The HLR and USIM do not store and track the value of SQN.
- The re-synchronization procedure for SQN is not implemented in DoCoMo's network.

[38] These values cannot be given for security reasons. b+c = constant.



Source: Author, 2005.

Figure 4-6: Verification of SQN in the DoCoMo's Network

4.1.2 Support Only Fixed-length Parameters

Several DoCoMo's functions are designed to support only fixed length parameters. The authentication procedure in GSM supports only traditional authentication algorithms with fixed length parameters (see Figure 4-7), and UMTS supports enhanced authentication algorithm with variable length parameters. In order to enhance the security features, 3GPP specifications recommend that mobile carriers support variable length security parameters (AUTS and XRES). However, DoCoMo did not implement this requirement to reduce development time and cost. Consequently, the authentication message in DoCoMo's network always forms 540 octets in the same order. When eavesdropping on the authentication messages in DoCoMo's network, one can easily analyze the message structure.

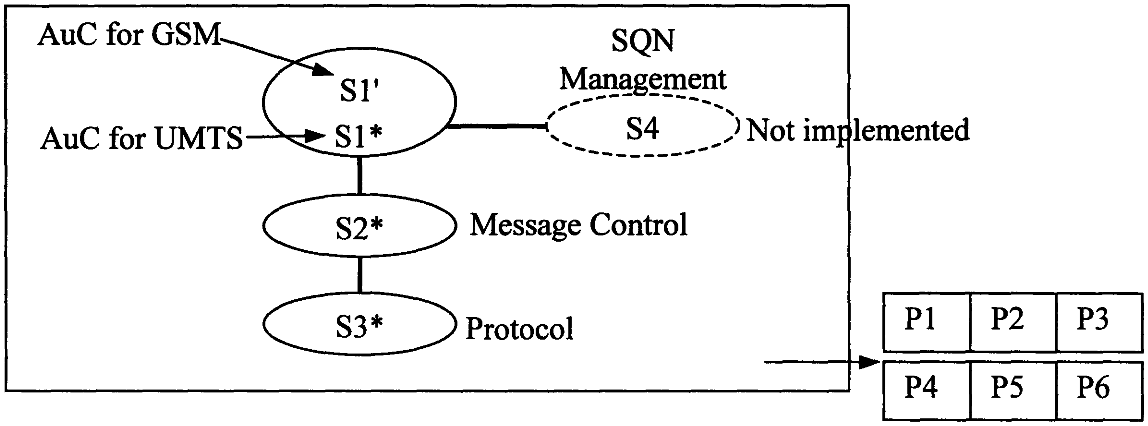
```
AuthenticationTriplet ::= SEQUENCE {
    rand                          RAND,  --16 octets fixed
    sres                          SRES,  --4 octets fixed
    kc                            Kc,    --8 octets fixed
    ...}
```

Source: 3GPP TS 29.002 V6.8.0 [11], p341.

Figure 4-7: Verification of SQN in the DoCoMo's Network

From the viewpoint of the software architecture, the AuC functions for GSM networks are completely integrated in the existing subsystem (see Figure 4-8). In this integral architecture subsystems form a complex (non one-to-one) mapping between functional elements and architecture and interfaces or boundaries between subsystems are not defined. The AuC for UMTS generates the quintuplets (authentication vectors) and derives the triplet for GSM networks by means of the standardized conversion functions in the subsystem. Changes in the UMTS authentication functions directly affect the GSM authentication functions. This integrated software architecture makes network software inflexible.

- DoCoMo only supports fixed length parameters in authentication (Degradation of the authentication mechanisms).
- The AuC functions are completely integrated in DoCoMo's system (software architecture is inflexible).



Source: Author, 2005.

Figure 4-8: Completely Integrated AuC functions

4.1.3 No Support for AMF

The authentication management field, AMF, is a 16-bit component of the authentication vector (part of the AUTN). AMF is supposed to enhance the security features of the network. AMF can be used for purposes, such as support for multiple authentication algorithms, verification of sequence number freshness, and threshold values to change the lifetime of the authentication keys. The use of AMF is not standardized by 3GPP but is specified by each mobile carrier.

Since the first service release in 2001, DoCoMo has not used AMF in its authentication procedure. The structure of AMF in DoCoMo's network is shown in Figure 4-9. The top four bits are all set at 0, and the other bits are ignored ("don't care"). Currently, DoCoMo's network software does not check the top four bits, which means AMF has become an entirely useless parameter in DoCoMo's network. This inadequate security design will make mobile systems vulnerable to unauthorized access in mobile communications.

Also, DoCoMo does not apply SQN management mechanisms. Network authentication is mainly based on the K (secret key) and SQN. Once intruders find out the K by replay attacks, it is quite possible to enable fraudulent communications. Without using AMF in DoCoMo's network, it is challenging to enhance security features.

- DoCoMo has not utilized AMF to enhance security features.
  ➔New security algorithms cannot be applied.
- If K is detected by replay attacks, fraudulent communications are possible.

Top4 bits = all 0	Other 12 bits = don't care

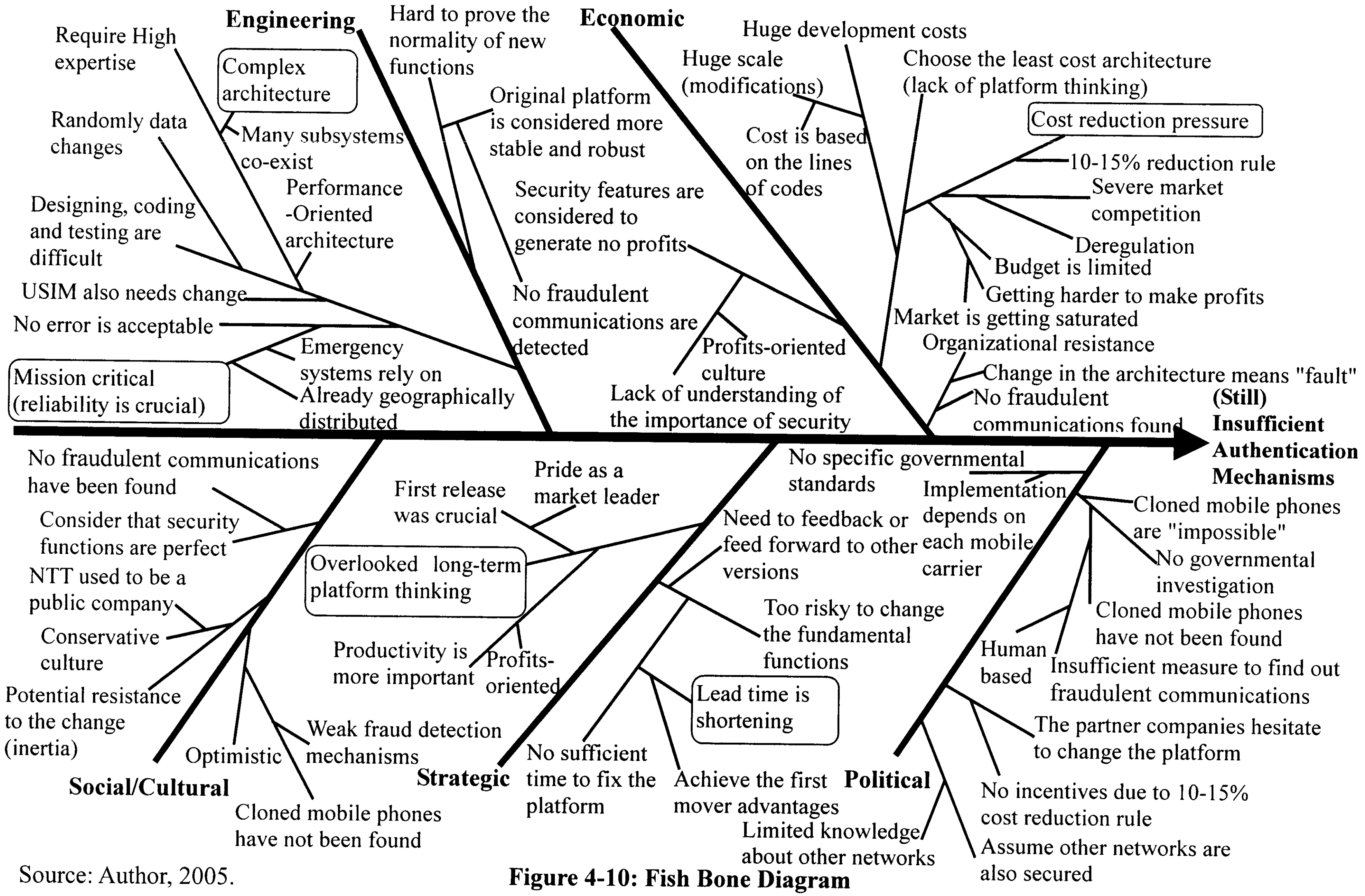
Source: Author, 2005.

Figure 4-9: AMF in DoCoMo's Network

4.2 Reasons for Insufficient implementation

As discussed above, DoCoMo's network authentication mechanisms are insufficiently implemented to prevent fraudulent communications, and this inadequate system design will make mobile systems vulnerable to unauthorized access. In order to analyze this situation, I applied a framework that is designed to highlight the role of organizational processes in complex social and technical systems [68]. This framework focuses on 5 perspectives—strategic, economic, engineering, political, and social/cultural—that reflect years of studies, interviews, observations, research and participation in organization [69].

By integrating the framework above with a fishbone diagram (see Figure 4-10), we can identify five major reasons for the current situation: (1) pressure of time to market, (2) pressure of cost reduction, (3) requirement for high reliability, (4) complex architecture, and (5) insufficient platform thinking.

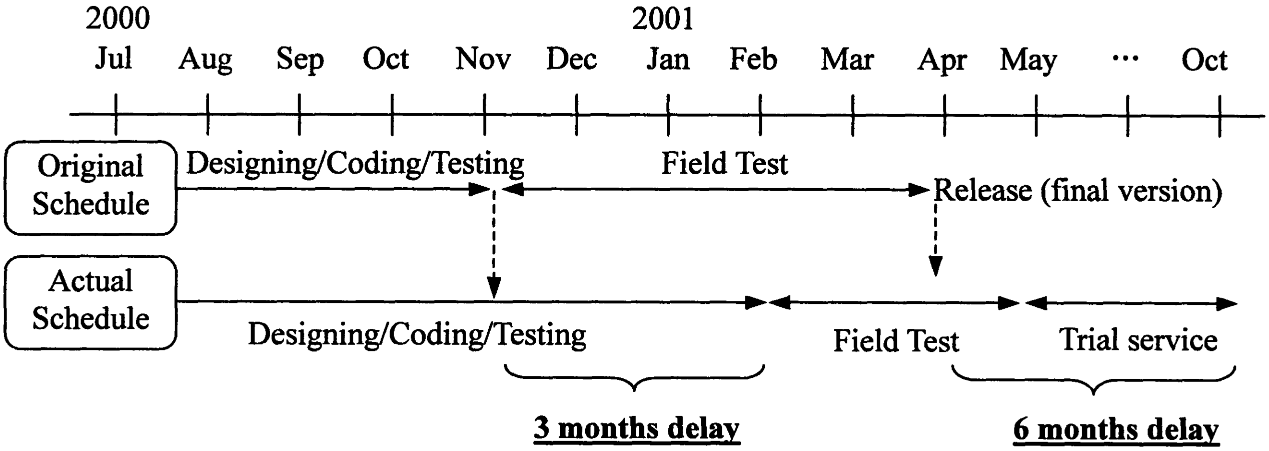

Engineering
Require High expertise
Complex architecture
Many subsystems co-exist
Randomly data changes
Performance -Oriented architecture
Designing, coding and testing are difficult
USIM also needs change
No error is acceptable
Emergency systems rely on
Mission critical (reliability is crucial)
Already geographically distributed
Hard to prove the normality of new functions
Original platform is considered more stable and robust
Security features are considered to generate no profits
No fraudulent communications are detected
Economic
Profits-oriented culture
Lack of understanding of the importance of security
Huge development costs
Huge scale (modifications)
Cost is based on the lines of codes
Choose the least cost architecture (lack of platform thinking)
Cost reduction pressure
10-15% reduction rule
Severe market competition
Deregulation
Budget is limited
Getting harder to make profits
Market is getting saturated
Organizational resistance
Change in the architecture means "fault"
No fraudulent communications found
(Still) Insufficient Authentication Mechanisms
No fraudulent communications have been found
Consider that security functions are perfect
NTT used to be a public company
Conservative culture
Potential resistance to the change (inertia)
Optimistic
Weak fraud detection mechanisms
Cloned mobile phones have not been found
Social/Cultural
First release was crucial
Pride as a market leader
Overlooked long-term platform thinking
Productivity is more important
Profits-oriented
Strategic
No specific governmental standards
Need to feedback or feed forward to other versions
Too risky to change the fundamental functions
Lead time is shortening
No sufficient time to fix the platform
Achieve the first mover advantages
Implementation depends on each mobile carrier
Cloned mobile phones are "impossible"
No governmental investigation
Cloned mobile phones have not been found
Human based
Insufficient measure to find out fraudulent communications
The partner companies hesitate to change the platform
No incentives due to 10-15% cost reduction rule
Assume other networks are also secured
Limited knowledge about other networks
Political

Source: Author, 2005.

Figure 4-10: Fish Bone Diagram

4.2.1 Strategic Viewpoint

As a leading mobile carrier, it was critical for DoCoMo to launch the world's first 3G mobile services. However, in 1999 software development was three months behind schedule. In order to accelerate software development for launch of 3G services worldwide in May 2001,[39] DoCoMo had to simplify its security functions and concentrate on software productivity. In 2000 DoCoMo still fell behind schedule (see Figure 4-11), which meant there was no time to improve the original platform. Time constraints did not allow DoCoMo to apply long-term platform thinking.



Source: Author, 2005.

Figure 4-11: Original Schedule vs. Actual Schedule

Moreover, once security mechanisms are implemented and released to the market, it is much more difficult to change the software architecture. Security comprises fundamental functions in the platform, and failures of upgraded functions can produce critical errors. The software lifecycle has become shortened—from two years, to one year, to three to six months

[39] The original release schedule was May 30, 2001, but actual launch was October 2001.

today. With such insufficient development time, it is too risky to change the original platform. In addition, several new software versions are developed during the same period of time, and each modification must work both backward to earlier version and forward to new versions. However, problems often arise because the network engineers who work on the second (or subsequent) versions are usually different from those who worked on the first version, which causes communication problems in the backward/forward reviewing activity. Serious efforts must be dedicated to modifying the operating platform.

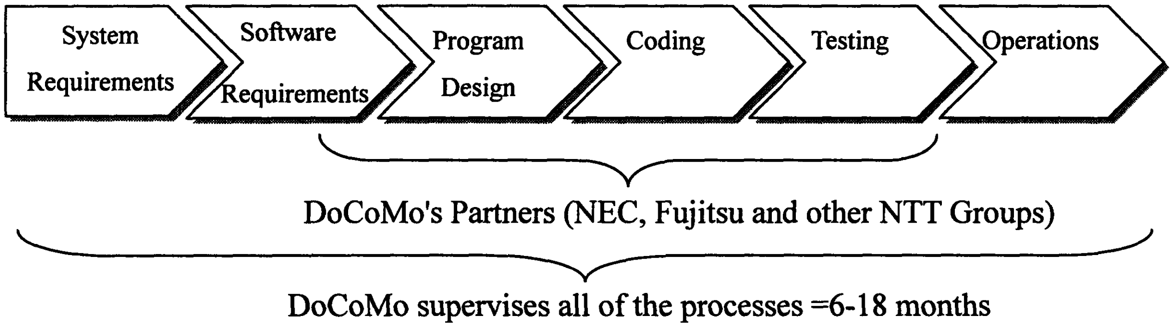
- The first release of the 3G system worldwide was critical for DoCoMo.
- The development fell behind the schedule➔Focus on the productivity
- Time constraints did not allow DoCoMo to apply "platform renewal."
- Once implemented, the current platform is considered more reliable and robust.
- Network engineers change from version to version.
- Platform modifications affect other software versions under development.

4.2.2 Economic Viewpoint

Severe market competition continually forces mobile carriers to reduce the software development costs. In the software development DoCoMo heavily relies on partner companies, such as NEC, NTT Comware, and Fujitsu (see Figure 4-12). Typically, software development costs are based on the scale of development. Therefore, DoCoMo strictly restricts the development size and tries to reduce expenditures to partner companies. In fact, security functions account for only 5 KLOC, which is less than 1% of KLOC in the total software development.

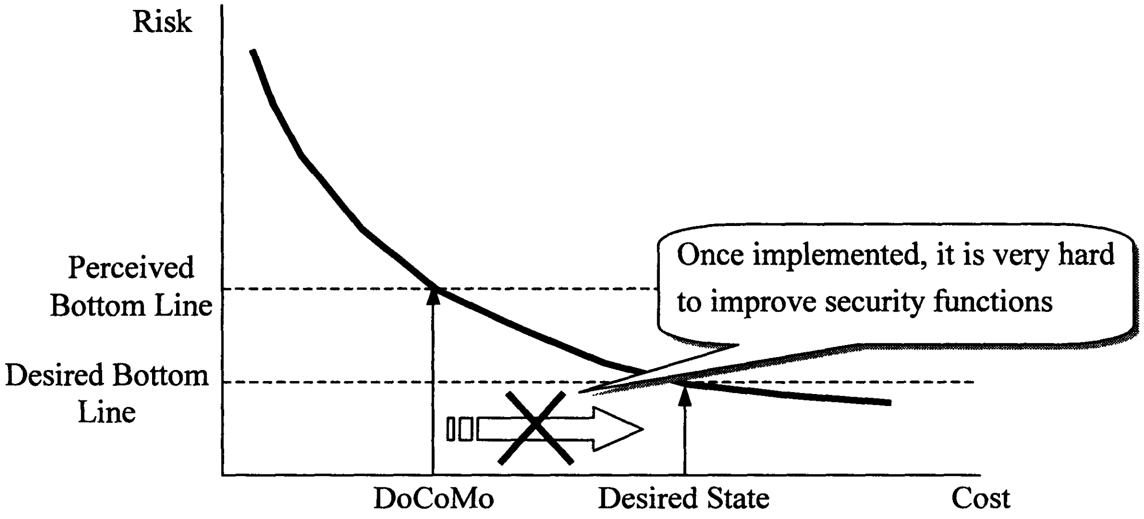
Given the insufficiently implemented security mechanisms, the perceived security level is lower than the desired security level. If there is a high consequence breach of security, the situation will shift to support more security features (see Figure 4-13). However, no fraudulent communications have been found to date, which means the level of security can be considered sufficient. Moreover, unlike other new services, such as global roaming, auto answering, and call waiting, the security function is not viewed as generating profits. Lack of understanding of the importance of security and profit-oriented culture did not allow the organization to implement sufficient security mechanisms.

Making changes in the security functions could imply not only the additional expenditure but also the possibility of failure in the architectural design, which the organization does not wish to admit. Thus, once the bottom line of the security functions is satisfied, it is far more challenging to change the organizational mindset and improve existing functions.


System Requirements
Software Requirements
Program Design
Coding
Testing
Operations
DoCoMo's Partners (NEC, Fujitsu and other NTT Groups)
DoCoMo supervises all of the processes =6-18 months

Source: Author, 2005.

Figure 4-12: Network Software Development of DoCoMo



Source: Author, 2005.

Figure 4-13: Risk vs. Cost for Security Functions

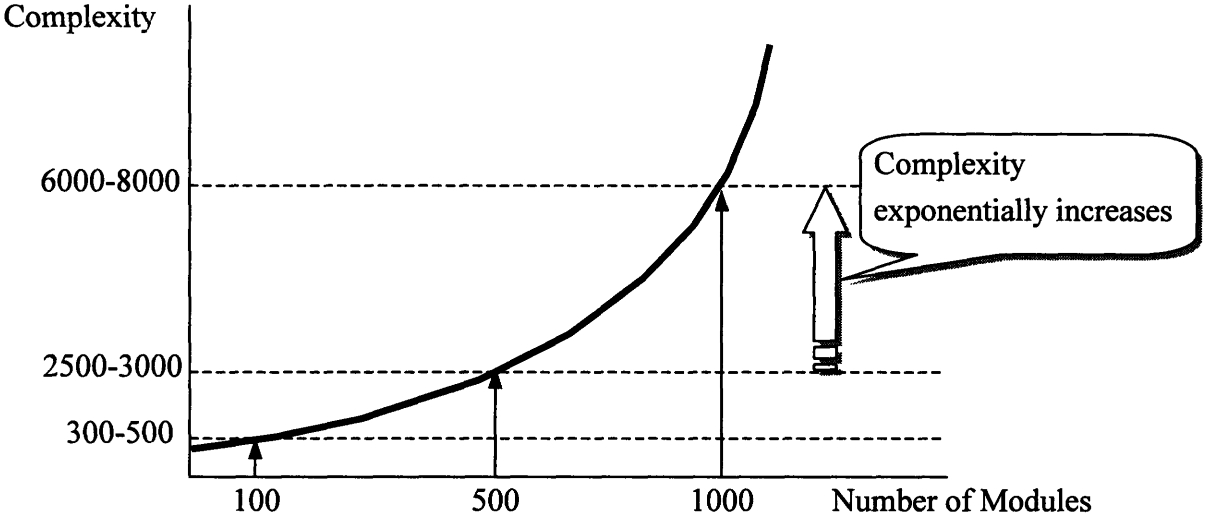
The cost reduction pressure leads DoCoMo to choose the least cost architecture. A limited budget and severe market competition cause this pressure, but it is important to point out DoCoMo's own rule, which is to decrease development costs. Initial development requires sophisticated expertise, so the original cost (payment to partner companies) for one line of codes reaches about ¥10,000. However, the second and third versions are less difficult to develop because the accumulated knowledge allows partner companies to reduce the costs. In the DoCoMo's rule, the discount rate is 10 to 15% in each version. Consequently, the cost per line of codes usually drops from ¥10,000 to ¥7,000.

Security functions are extremely difficult to manage. Considering the enormous efforts required to achieve zero defects, DoCoMo's partner companies tend to avoid changing the original architecture. The problem is that security mechanisms are mission critical, but DoCoMo does not distinguish the difference between mission-critical and non-mission-critical functions. Instead DoCoMo applies the same standards to reduce the development costs.

- In order to reduce the development costs, DoCoMo restricts the software size (scale).
- Security functions are considered less profitable than other services.
- Cost reduction pressure forces DoCoMo to choose the least cost architecture (lack of platform thinking).
- The 10 to 15% cost reduction rule does not give incentives to improve the insufficient platform.
- DoCoMo does not distinguish the difference between mission critical and non-mission critical functions, instead applying the same rule to reduce the development costs.

4.2.3 Engineering Viewpoint

As discussed in Chapter 3.3.4, the architecture of network software is complex with over 1,000 modules in the same platform. These modules are interconnected, and the complexity increases dramatically when the number of modules increases (see Figure 4-14). Furthermore, in order to achieve high performance, several software engineering principles (e.g., less coupling among modules and strong cohesion) are intentionally disregarded when it comes to the security functions. Combining the knowledge of the operating system, hardware, and assembler language, engineers have to develop a performance-oriented architecture. An extremely high level of expertise is required to modify existing security functions (see Table 4-2). After implementing the first version and fixing software defects associated with the security functions, very few engineers can consider changing the basic architecture.



Note: Complexity is based on the number of interfaces among modules (subsystems)
Source: Author, 2005.

Figure 4-14: Number of Modules and Complexity of the System

Network software operates in real-time telecommunication systems. HLR/AuCs are geographically distributed in DoCoMo's network (a total of 140 units and 500+ CPUs), and the same network software operates in different hardware. Like AT&T's case in 1990, critical errors in mission critical functions immediately cause disastrous outcomes worldwide. Thus, zero defects are mandatory in any new platform.

Table 4-2: Examples of the Coding Rules for Time-Critical Modules

No.	Requirements	Expected Effects	Notes
1	Apply minimum buffer size (copy and clear)	400 μ seconds better	It depends on the buffer size.
2	Use the same the buffer between subsystems	60 μ seconds better	Modularization rule is not applied.
3	Avoid using many variables	2-3 μ seconds better	OS prefers fewer variables.
4	Check the search logic	10 μ seconds better	It depends on the number of messages.
5	Avoid bit calculation	2-3 μ seconds better	NEC OS prefers the "ULONG" types
6	Avoid CHAR types	2-3 μ seconds better	NEC OS prefers the "ULONG" types
7	Apply "switch-case" instead of "if-else"	2-3 μ seconds better	NEC OS prefers the "switch-case"

Source: Author, 2005.

In addition, as discussed in Chapter 2, security functions are implemented in both mobile handsets and HLR/AuC. Over 17 million subscribers enjoy 3G services and the number is increasing. Changes in the essential security mechanisms (algorithms) lead to significant modifications in both USIM and HLR/AuC. One software defect can result in critical outcomes. Under this situation, it is quite challenging to prove the normality and necessity of upgrading the security features. Today's authentication mechanisms are not sufficient, but the requirement for high reliability prevents DoCoMo from making architectural innovations in the platform and leads DoCoMo to focus on incremental innovations.

- The platform architecture and security mechanisms are complex.
- A high level of expertise is required to modify the platform.
- Zero defects are crucial in the new platform (requirement for high reliability).
  ➔ It is quite hard to prove the normality and necessity of new security functions.
- USIM in mobile handsets also needs to change when essential security mechanisms (algorithms) are modified.

4.2.4 Political Viewpoint

At present no government agency or investigation is required to verify the security functions. Implementation and security standards depend entirely on the mobile carriers themselves. As long as no evidence of fraudulent communication is found, the government does not impose regulations. If implemented security mechanisms function perfectly, and the network is secure, the situation remains stable and there is little worry. In the real world, however, no company can rely on weak security assumptions such as: (1) fraudulent communications can be detected and traced systematically, and (2) other networks are also secure.

In the discussion of security functions in Chapter 2, I noted that it is possible to track potential fraudulent communications. An analysis of the "Authentication Failure Report (failure messages during user authentication)" messages from VLR/SGSN allows mobile carriers to detect possible fraudulent communications. Data trace functions also help analyze the real traffic data of the target user.

However, DoCoMo has not utilized these mechanisms. Fraud detection relies heavily on human recognition by operators. DoCoMo's HLR/AuC can receive error messages from VLR/SGSN, but no systematic data analysis is conducted. If operators in each location (e.g., Tokyo area, Hokkaido area, and Osaka area) overlook the error information, clues about potential fraudulent communications will be completely lost. In this case only claims or complaints from end-users will likely trigger an investigation into fraudulent communications.

In addition, receiving unfamiliar messages is often considered a defect in the network software rather than a clue about fraudulent communications. On the basis of the DoCoMo's security assumptions (i.e., fraudulent communications are technologically impossible), software defects are more likely than unexpected problems. Without complaints about fraud from end-users, operators rarely expect to encounter fraudulent communications.

DoCoMo proceeds under the assumption that other networks are also secure. Compared with GSM networks, 3G networks are not as widely deployed around the world. If DoCoMo's subscribers go to other networks that support only GSM systems, GSM authentication mechanisms are inevitably applied. Also, some 3G networks do not implement security mechanisms, such as encryption, in their access links (the air zone between a mobile handset and the core network). In these cases DoCoMo cannot guarantee perfect authentication. Network traffic could be subject to eavesdropping and analysis by intruders. DoCoMo's simplified security mechanisms more easily allow intruders to access the network.

DoCoMo's official position is that fraudulent communications, mainly via cloned mobile phones, are impossible in 3G mobile systems. However, we can conclude that this statement is based on weak security assumptions and is not dependable.

- No specific government regulation or investigation exists for security management.
- No sufficient fraud detection mechanisms are applied in DoCoMo's network.
  ➔Risk assessment and risk control[40] are insufficient.
  ➔Fraud detection heavily relies on the human recognition.
  (We cannot trust the announcement that guarantees "perfect security.")
- Other networks (GSM and sometimes UMTS) are not so secured.
  ➔Security holes exist in the communications between DoCoMo and other networks.

4.2.5 Social/Cultural Viewpoint

DoCoMo was spun off from Nippon Telegraph and Telephone Corporation (NTT) in April 1992. Thirteen years have passed, but a basic corporate culture is similar to NTT's, i.e., very conservative. A peace-at-any-price principle prevails, and changes in fundamental functions are considered too dangerous. As Henderson and Clark describe [52], architectural innovations

[40] Risk assessment mainly involves risk identification and risk analysis. Risk control consists of risk management planning, risk resolution and risk monitoring [70].

create organizational resistance or inertia and tend to hinder the successful adoption of future architectural innovations.

No fraudulent communications mainly via cloned mobile phones have been found so far in DoCoMo's 3G systems. Current security functions seem to be operating well. In this conservative culture, it is quite difficult to correct the insufficient authentication mechanisms and suggest platform innovations in its systems.

- DoCoMo's corporate culture is conservative.
- Organizational resistance tends to hinder platform change.

4.3 Tradeoffs between Performance and Security

4.3.1 Dynamic Steps

Another critical bottleneck in DoCoMo's security mechanisms is performance. Network software must guarantee a real-time response for transactions. Table 4-3 shows the number of required transactions in DoCoMo's systems. Security functions are categorized in the basic call function and must operate at 600 TPS within 70% CPU usage. Above 70% CPU usage, some messages cannot be delivered on time (i.e., less than 30 seconds) and sometimes these messages are lost. Among the basic call functions, security functions require the most powerful CPU resources (see Figure 4-15). The security functions (Send Authentication Info + Send Authentication Info ack) need 2.7 to 8.7 times higher CPU performance than other functions. In particular, the generation of authentication vectors (Chapter 2.4) consumes much of the CPU's time. If the generation function is not required, HLR/AuC can process a single authentication transaction in 402.5 μ seconds; but if the generation function is implemented, it takes 2,607.4 μ seconds[41] (6.5 times higher). The total size of the security functions is small (less than 1% of

[41] This data is based on the first version of the software. Thereafter, DoCoMo did not measure the performance.

KLOC in the total software development), but the influence becomes significantly large in the operation.

Table 4-3: Major Differences among Subsystems (Call Functions)

Subsystem	Required Speed[1]	Required Quality	Required Resources
Basic call function (including security)	High (max 600 TPS[2])	Extremely high	Large
Supplementary service function	Low (max 200TPS)	High	Small
O&M function	Very Low (less than 10 TPS)	Medium	Small

Notes:

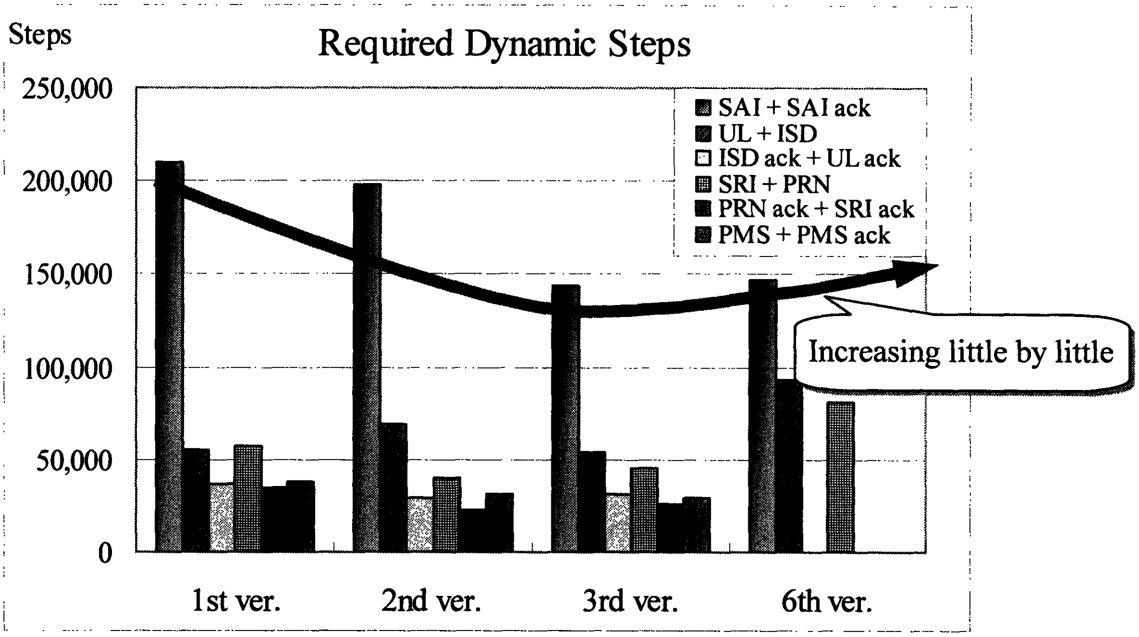
1) Each CPU has to satisfy the required TPS (Transactions Per Second).

2) One transaction equals a pair of messages: receiving and sending.

Source: Author, 2005.

The required number of dynamic steps for the security functions decreased from 209,420 in the first version to 143,861 in the third version. However, this decrease resulted from small modifications, such as changes in the buffer size and avoidance of bit calculation. Most of the small modifications had already been applied to the current platform. In fact, after the third version the number of dynamics steps remained virtually the same. Given the implementation of new services, the number of dynamics steps is now increasing little by little (146,862 steps in the sixth version in 2003). Platform extension to improve performance had reached its limits.

- Security functions (especially, generation of authentication vectors) require the highest CPU performance.
- The lines of codes for security functions are small, but greatly affect the systems performance.
- Small modifications (platform extension) to improve performance have already reached the limit.
- New security features face tradeoffs between performance and functionality.



SAI: Send Authentication Info, UL: Update location, ISD: Insert Subscriber Data, SRI: Send Routing Info, PRN: Provide Roaming Number, PMS: Purge MS

Operation	First Version	Second Version	Third Version	Sixth Version	Note
Send Authentication Info + Send Authentication Info ack	209,420	198,388	143,861	146,862	Authentication
Update location + Insert Subscriber Data	55,154	69,482	54,015	92,960	Location management[42]
Insert Subscriber Data ack + Update location ack	37,221	28,803	30,935		Location management
Send Routing Info + Provide Roaming Number	57,379	39,557	45,135	80,871	Terminating calls (From HLR to VLR)
Provide Roaming Number ack + Send Routing Info ack	34,908	22,879	26,412		Terminating calls (from VLR to HLR)
Purge MS + Purge MS ack	38,294	31,357	28,836	-	Location and routing management
Send Routing Info for SM + Send Routing Info for SM ack	45,975	53,868	47,979	47,046	Supplementary service
MAP_CLOSE	20,056	-	-		Termination
Send Routing Info for GPRS + Send Routing Info for GPRS ack	31,987	31,987	22,996	28,913	Packet calls

Source: Author, 2005.

Figure 4-15: Required Dynamic Steps for Basic Call Functions

[42] This location management is DoCoMo's specific operation including both circuit and packet update location.

4.3.2 Network Traffic

The network traffic affects the systems performance. Authentication messages influence the systems by (1) amount of traffic and (2) message length. On the basis of the network traffic in PDC services, it is estimated that the traffic for "Send Authentication Info (SAI)" constitutes 17.2% of total traffic in HLR (see Table 4-4). Actual traffic data from the Yokohama unit supports this traffic model (see Figure 4-16). These traffic data suggest that authentication mechanisms operate more frequently than terminating call functions and short message functions. Thus, the increase in security traffic will significantly affect systems performance.

Table 4-4: Network Traffic of 3G Services

No.	Operation	Ratio	Max Length (Octets)	Note
1	Update location + Insert Subscriber Data	30.9%	98 (UL) 702 (ISD)	Location management
2	Insert Subscriber Data ack + Update location ack	30.9%	21 (ISD ack) 33 (UL ack)	Location management
3	Send Authentication Info + Send Authentication Info ack	17.2%	65 (SAI) 540 (SAI ack)	Authentication
4	Send Routing Info, Send Routing Info ack	6.1%	127 (SRI) 151 (SRI ack)	Terminating calls
5	Provide Roaming Number, Provide Roaming Number ack	3%	137 (PRN)	Terminating calls
6	Send Routing Info for GPRS, Send Routing Info for GPRS ack	9.1%	97 (SRG) 101 (SRG ack)	Packet communications
7	Others	2.8%	112 (SRS ack)	Supplementary services

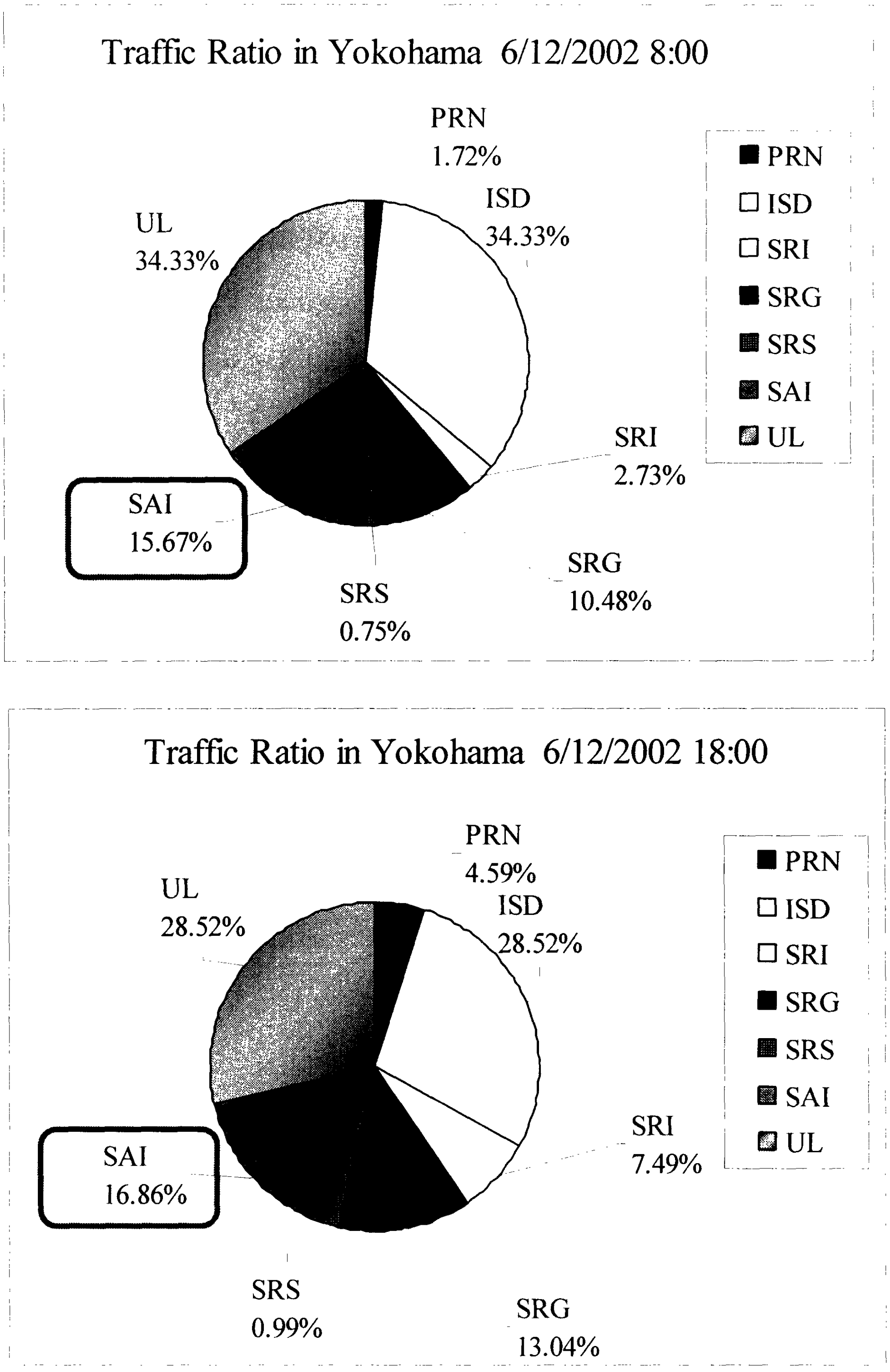
UL: Update location, ISD: Insert Subscriber Data, SAI: Send Authentication Info, SRI: Send Routing Info, PRN: Provide Roaming Number, SRG: Send Routing Info for GPRS, SRS: Send Routing Info for SM

Source: Author, 2005.

The length of the SAI message also can result in low systems performance. SAI comprises 540 octets of authentication data (the second-largest of all the messages). At the same time, the capacity and number of physical links are limited (e.g., 384kbps, 16 links in rural area). Thus, a heavy volume of SAI can create severe traffic congestion in the network. When adding new security features, network engineers must consider the impact of network traffic in order to avoid network congestion.

- Security traffic comprises 17.2% of all the network traffic in HLR.
- The message length of the SAI is the second longest.

➔Security messages can cause severe network traffic congestion.



Source: Author, 2005.

Figure 4-16: Network Traffic in DoCoMo's 3G Systems

4.3.3 CPU Usage

The number of dynamic steps and the traffic model help to clarify the relationship between dynamic steps and CPU usage. The CPU requires a certain amount of dynamic steps in order to execute the process. If the quantitative relation between these two parameters can be determined, we can evaluate the impact of new features added to the existing platform.

Integrating data from Figure 4-15 with Table 4-4, we can estimate the required number of dynamic steps to complete 100 messages (see Table 4-5). The graph of CPU usage is quite linear (see Figure 4-17). According to the graph, CPU usage increases 9.6%, 9.2% and 8.0% (per 100 transactions) in the first, second and third version respectively. Thus, the relationship between dynamic steps and CPU usage can be derived as follows.

- The first version: 7,387,116 steps (100 messages): 9.6% up→1.29956E-06 %up/step
- The second version: 7,201,126 steps (100 messages): 9.2% up→1.27758E-06 %up/step
- The third version: 5,797,529 steps (100 messages): 8.0% up→1.3799E-06%up/step

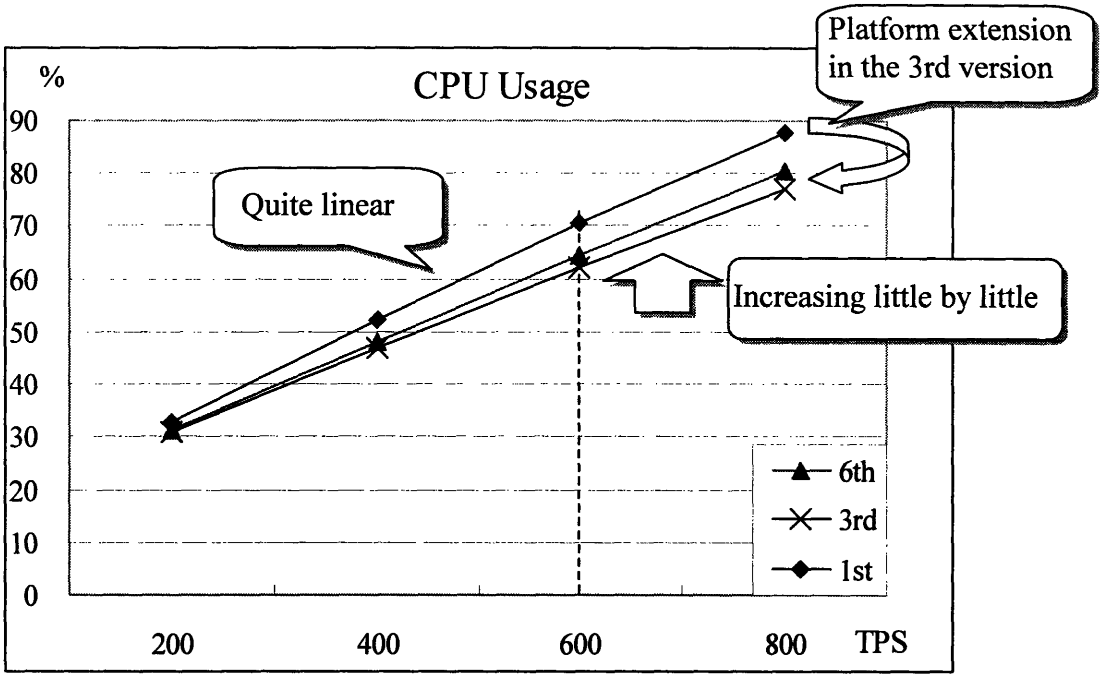
Average: 1.31901E-06%up/step (7,581 steps added→1% up/100TPS)

Table 4-5: Network Traffic Model and Dynamic Steps

No.	Operation	Traffic/100 Messages	Traffic * Dynamic Steps		
			1st Version	2nd Version	3rd Version
1	SAI-SAI ack	17.2	3,602,024 (48.8%)	2,146,994 (29.8%)	1,669,064 (28.8%)
2	UL-ISD	30.9	1,704,259	890,013	955,892
3	ISD ack-UL ack	30.9	1,150,129	3,412,274	2,474,409
4	SRI-PRN	6.1	350,012	241,298	275,324
5	PRN ack-SRI ack	3.0	104,724	68,637	79,236
6	SRG-SRG ack	9.1	291,082	150,830	134,341
7	SRS-SRS ack	2.8	184,887	291,081	209,264
All	Total	100	7,387,116	7,201,126	5,797,529

UL: Update location, ISD: Insert Subscriber Data, SAI: Send Authentication Info, SRI: Send Routing Info, PRN: Provide Roaming Number, SRG: Send Routing Info for GPRS, SRS: Send Routing Info for SM

Source: Author, 2005.



	200 TPS	400 TPS	600 TPS	800 TPS
1st version	32.8%	52.3%	70.7%	87.6%
2nd version	31.2%	49.8%	67.7%	85.0%
3rd version	30.7%	46.8%	62.0%	77.2%
4th version	30.9%	46.9%	62.1%	77.5%
5th version	31.0%	47.9%	64.0%	80.3%
6th version	31.3%	48.1%	64.3%	80.5%

Source: Author, 2005.

Figure 4-17: CPU Usage of DoCoMo's 3G Systems

Compared with the number of dynamic steps in current functions (e.g., authentication = 143,861steps), the number of acceptable dynamic steps is quite small (7,581 steps). The first two versions had improper memory usage in their security functions, so CPU usage exceeded system requirements (600 TPS within 70% CPU usage). When implementing the third version, DoCoMo applied the platform extension to improve the unnecessary functions. This approach helped enhance software performance, and CPU usage dropped from 67.7% to 62% (600 TPS). However, after the improvement, no significant platform innovations have been applied to the network software. New mobile services force CPU usage to increase little by little. If CPU usage increases even 1% (600 TPS) in each version,[43] the platform will be able to accept only six more versions (through 2006). In particular, the security functions already occupy 28.8% to 48.8% of the total number of dynamic steps. The number of 3G subscribers is also increasing. This result indicates that DoCoMo's platform has almost reached its limits. It is quite challenging to implement new security features without platform innovation (platform renewal).

- The relationship between dynamic steps and CPU usage can be determined (7,581 steps added➔1% increase in CPU usage/100TPS)
- DoCoMo's platform will reach the limits of its systems performance around 2006.
- When DoCoMo implements new security features, platform renewal will be crucial.

[43] This situation means that increase in 1264 steps/100TPS is acceptable (7581steps/6 =1264).

5 Securing Against Fraud in Mobile Communications

Since October 2001, mobile carriers have been deploying 3G mobile systems. Today, 61 UMTS commercial services operate worldwide, and more than 33 million subscribers enjoy 3G services.[44] With a global roaming service, 3G networks are becoming more interconnected with other 2G and 3G networks. In the near future, seamless interoperability among mobile networks is expected to be achieved. Network access and authentication requests will be generated both inside and outside the network. Therefore, securing their own mobile systems against unauthorized access will become even more essential.

To date, no cloned mobile handset or fraudulent communications have been found in 3G mobile systems. However, the lessons learned from the existing network software (DoCoMo) suggest that the security mechanisms implemented today are imperfect and vulnerable. In particular, replay attacks seeking to obtain network authentication can break security codes and allow fraudulent communications.

To a greater or lesser extent, other mobile carriers have similar problems.[45] Based on the lessons learned from the existing software in 3G mobile systems, this chapter will suggest how mobile carriers can avoid potential fraudulent communications and secure their mobile systems.

5.1 Learning from DoCoMo's Network Software Development

DoCoMo's case offers important lessons: (1) implemented security features are vulnerable to fraudulent communications, (2) various pressures, such as requirements for a shorter lead time, cost reduction, and high reliability, prevent sufficient implementation of the security features and

[44] UMTS Forum, as of August 2005

[45] Interviews with several network engineers in other 3G systems suggest similar security problems. However, specific information cannot be given for security reasons.

skew the platform architecture, and (3) the platform has its limits and platform renewal should be applied in order to break through the old constraints.

First, a clear gap exists between global standards and actual implementation. Several critical security features, such as SQN management and AMF, are sometimes missing in the platform. In particular, replay attacks can break security codes and allow fraudulent communications. Removing the possibility of fraudulent communications requires fixing the platform architecture for authentication mechanisms. In order to avoid the large-scale failure caused by software defects, highly sensitive tests are required. Furthermore, upgrading network software and renewing USIM cards (over 17 million cards) are also necessary, which is neither easy nor cheap. Indeed, mobile carriers will not choose this option until fraud becomes a major problem in their networks.

Second, in order to meet delivery schedules and reduce development costs, mobile carriers tend to reduce the required functions. When developing network software, network engineers are forced to concentrate on optimizing the current software and tend to overlook extensibility for the next version. The lack of long-term platform thinking results in complex and inflexible platform architecture. In such a complex architecture, the current operating platform is considered more reliable than a new one. In fact, fundamental change may trigger hidden failures and cause serious consequences in the telecommunications systems. Once the original platform is implemented, architectural change becomes quite difficult due to organizational resistance. In fact, in 2001-2002, DoCoMo discussed architectural changes in its operating platform, but DoCoMo concluded it was too risky to apply such architectural changes. This kind of organizational resistance and inertia regarding platform modifications remains a key challenge.

Third, DoCoMo's case suggests that platform extension cannot be a fundamental solution for resolving platform performance problems. Security-related traffic presents 17.2%, and security features take up nearly 30% of the CPU performance. If we assume that (1) the number of 3G subscribers is growing, and (2) that new user-oriented application services will require more network authentication procedures, platform renewal will be inevitable in order to satisfy the required performance targets. To improve performance, network engineers can apply a platform extension strategy by configuring the network software and reducing unnecessary procedures. However, this method can be effective only once. In DoCoMo's case, after the third version of network software, no significant improvements in performance have been achieved.

DoCoMo's case also indicates that the extensibility of the platform can be measured by estimating the number of dynamic steps. Before reaching performance limits, mobile carriers can take appropriate measures to avoid system failure.

- A clear gap exists between global standards and actual implementation.
- Various pressures (e.g., short lead time, cost reduction) prevent sufficient implementation of the security features and skew the platform architecture.
- Platform change is difficult owing to organizational resistance and/or inertia.
- The platform has its limits, and platform renewal should be applied in order to break the constraints.

5.2 Strategies for Securing Mobile Systems

As discussed in Chapter 2.5, securing mobile systems sufficiently remains a challenging issue. However, mobile carriers can reduce potential risks by understanding current platform conditions and addressing architectural and implementation problems. Based on the platform thinking, Figure 5-2 and Figure 5-3 show basic strategies to secure mobile systems[46]. Substantial differences exist before and after deployment of services, so the classification is below:

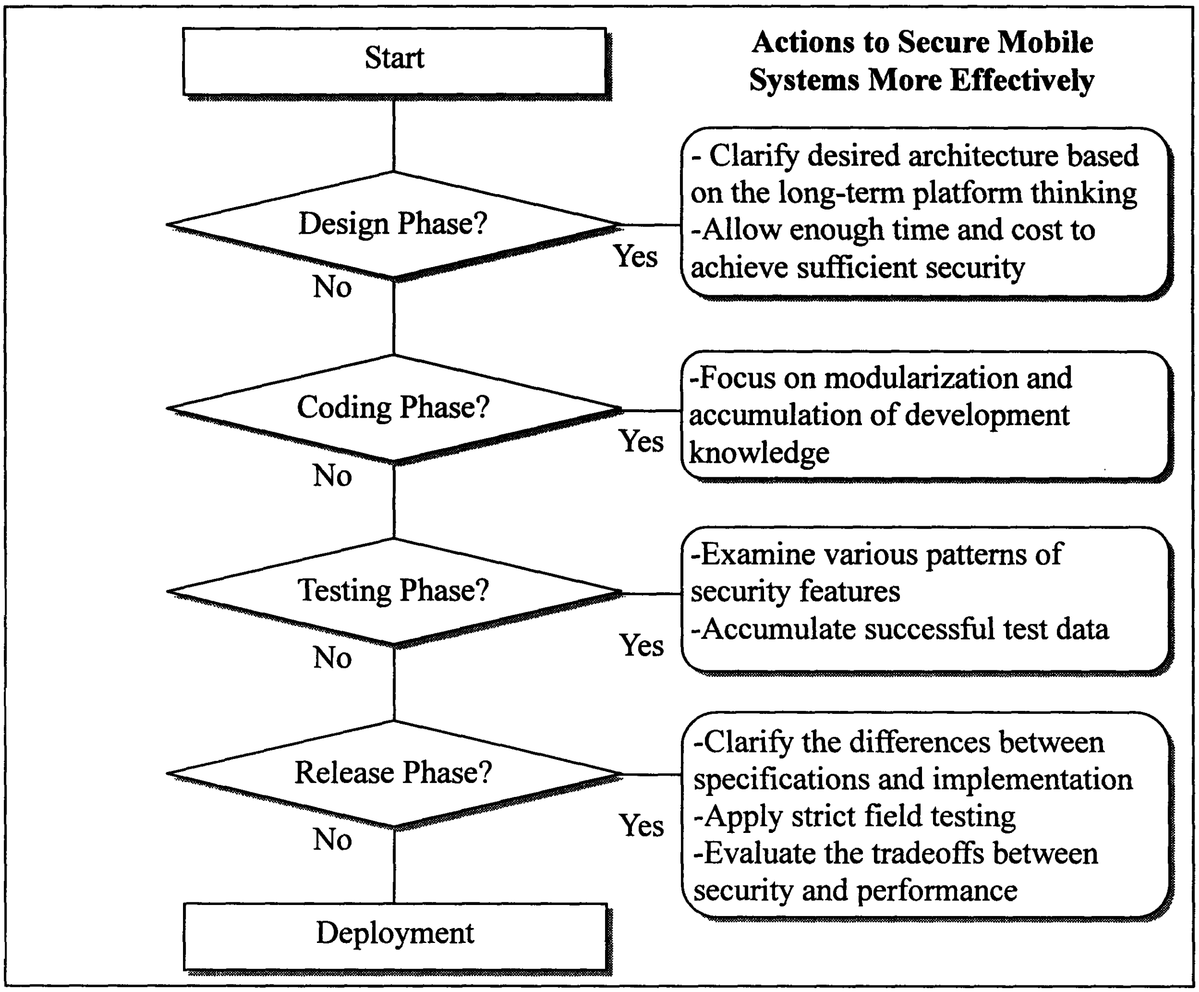
- Before deployment: Software development phase based on a waterfall model
- After deployment: Implementation level and platform architecture (see Figure 5-1)

[46] More research is needed to support implementation strategies (e.g., research on Vodafone's case).

		Implementation Levels (Security Features)		
		Sufficient	Insufficient, not so critical	Critical
Platform Architecture	Flexible, modularized	Well-secured in the long term	Secured in the long term	Vulnerable in the short term
	Inflexible, not modularized	Well-secured in the short term	Secured in the short term	Vulnerable in the long term

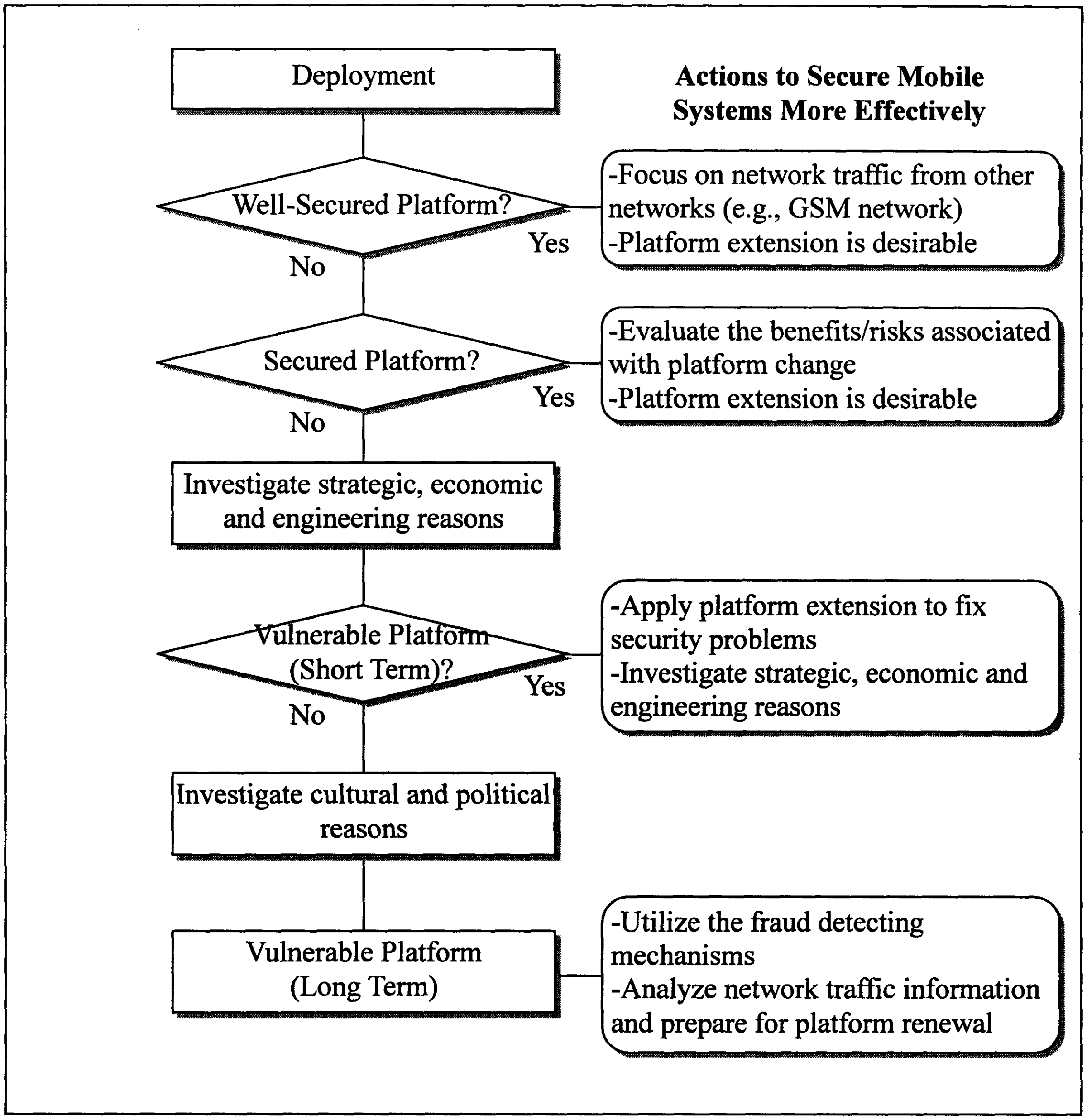
Source: Author, 2005.

Figure 5-1: Classification of Security Levels


Start
Actions to Secure Mobile Systems More Effectively
Design Phase?
Yes
No
- Clarify desired architecture based on the long-term platform thinking
-Allow enough time and cost to achieve sufficient security
Coding Phase?
Yes
No
-Focus on modularization and accumulation of development knowledge
Testing Phase?
Yes
No
-Examine various patterns of security features
-Accumulate successful test data
Release Phase?
Yes
No
-Clarify the differences between specifications and implementation
-Apply strict field testing
-Evaluate the tradeoffs between security and performance
Deployment

Source: Author, 2005.

Figure 5-2: Basic Strategies to Secure Mobile Systems (Before Release)


Deployment
Actions to Secure Mobile Systems More Effectively
Well-Secured Platform?
Yes
No
-Focus on network traffic from other networks (e.g., GSM network)
-Platform extension is desirable
Secured Platform?
Yes
No
-Evaluate the benefits/risks associated with platform change
-Platform extension is desirable
Investigate strategic, economic and engineering reasons
Vulnerable Platform (Short Term)?
Yes
No
-Apply platform extension to fix security problems
-Investigate strategic, economic and engineering reasons
Investigate cultural and political reasons
Vulnerable Platform (Long Term)
-Utilize the fraud detecting mechanisms
-Analyze network traffic information and prepare for platform renewal

Source: Author, 2005.

Figure 5-3: Basic Strategies to Secure Mobile Systems (After Release)

5.2.1 Before Deployment

Design Phase

The design phase directs the future platform architecture. If network engineers focus only on satisfying current requirements and overlook the flexibility and extensibility of the future platform, the platform will lose its potential capabilities. Better designs will require substantially less rework and redesign [71]. In this sense, the design phase is the most important phase in software development.

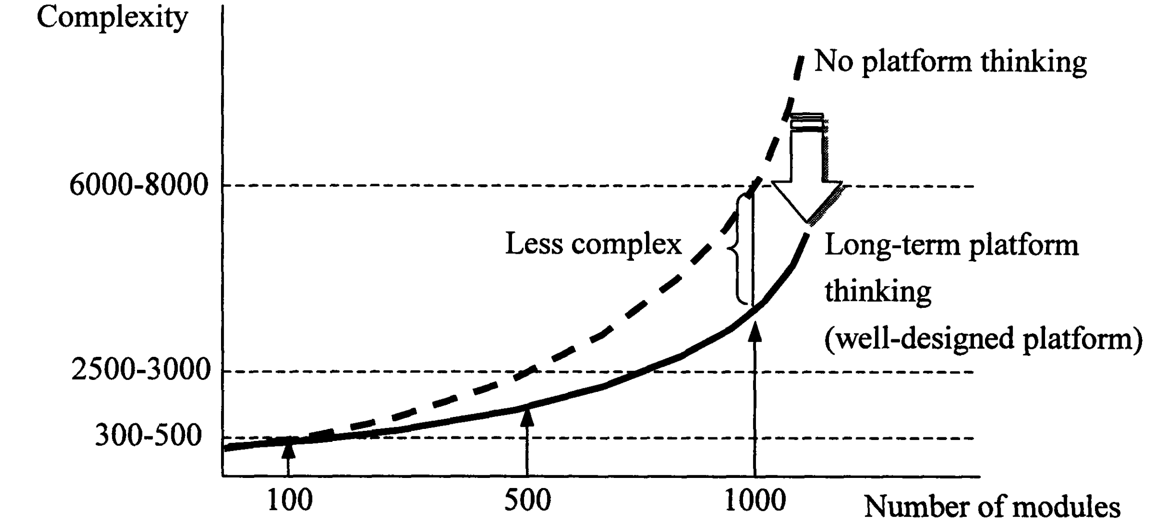
Before coding the software program, mobile carriers should clarify the desired architecture on the basis of long-term platform thinking. In order to avoid unnecessary defects and improve productivity, more complete specifications should exist prior to coding phase [72]. In the design phase, mobile carriers should specify interfaces among subsystems (functional elements) and try to achieve a modular architecture. Many unnecessary subsystems and redundant interfaces should be revised or eliminated to reduce complexity and improve extensibility (see Figure 5-4).

When DoCoMo developed its network software, global standards were not fixed and user requirements often changed, making it very challenging to design the entire systems effectively. Today, technical specifications and user requirements are virtually fixed, and mobile carriers planning to deploy 3G systems can design the platform architecture accurately and effectively.

Developing network software on schedule with the lowest possible costs is important, but mobile carriers should not focus on this requirement to the exclusion of other important requirements. Organizational resistance after the software release may not allow mobile carriers to achieve architectural changes in the network software. Thus, a certain amount of time and cost is necessary in order to satisfy sufficient security levels. DoCoMo's case suggests that security features require an enormous volume of CPU performance. A flexible and extensible platform can survive when new services are implemented and the platform confronts the "expansion phase" (see Figure 5-5).

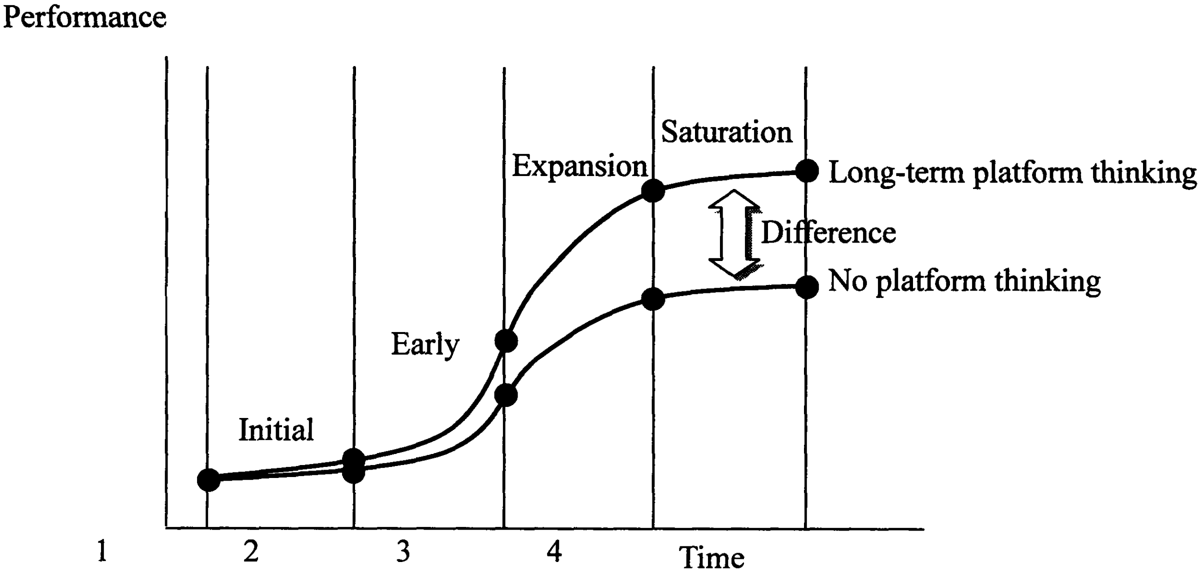
- A desired architectural design based on long-term platform thinking must be clarified.
  ➔System architecture should be less complex
- Flexibility and extensibility should be considered in the platform.

- Enough time and cost are required to achieve sufficient security.
- A well-designed platform can survive during the expansion phase.



Note: Complexity is based on the number of interfaces among modules (subsystems)
Source: Author, 2005.

Figure 5-4: Long-Term Platform Thinking and Complexity of the System



Source: Author, 2005.

Figure 5-5: Phases of Platform Innovation

Coding Phase

In the coding phase mobile carriers should focus on modularizing the functions and accumulating development knowledge (e.g., coding and debugging knowledge). Security features comprise fundamental functions in the platform and require extremely sophisticated expertise. Modularization helps create the highly complex software and localize the modifications. Well-designed modules can be used to create similar functions in the next version. For example, if mobile carriers develop well-modularized UMTS security functions, these carriers can apply the modules to create GSM security functions as well.

Software development knowledge (information development [73]) will also be accumulated in this phase. Mobile carriers should list up coding and debugging know-how and utilize this knowledge in future development. Specific hardware characteristics often restrict software development. In DoCoMo's case, the NEC OS and Fujitsu OS have different coding requirements; the same coding and configuration do not work in different machines. Thus, accumulated knowledge should continue to be accessible through succeeding software development in order to avoid mistakes found in previous versions. This knowledge will allow new network engineers working on succeeding versions to understand the coding methods and improve the platform.

- Mobile carriers should focus on modularization and accumulation of development knowledge.
- Accumulated knowledge should be accessible to succeeding software versions.

Testing Phase

During the testing phase, mobile carriers should focus on collecting data that identify the specific factors underlying and contributing to the problems. Defect analysis[47] and retesting (regression test) are crucial to ensure that the software quality levels are reached [74]. The ultimate goal of testing phase is to achieve the perfection (zero defects[48]) of the software.

Testing of security features is quite challenging because data length is very long and the data itself changes randomly for encryption. Thus, simple data, such as "0000" and "FFFF" is often applied to test security functions. In DoCoMo's case, the company often temporarily suspended the security functions in order to reduce complexity for testing.

If mobile carriers simplify the testing too much, it becomes difficult to justify the security features between mobile handsets and HLR/AuC. A wide variety of testing for security features should be necessary to guarantee the functions. At the same time, mobile carriers should establish automated testing systems, including mobile handsets or HLR/AuC simulators, to effectively test the security features. Sometimes an automated test adds considerable complexity and requires more efforts from the test team, but it can also provide valuable assistance for justifying security features in the right environment. In succeeding versions, test results can be used to confirm the utility and reliability of new functions in a new platform.

- Mobile carriers should test various patterns of security features to justify authentication mechanisms.
- Automated testing allows various testing patterns and helps confirm security features.
- Accumulated successful test data can confirm the new functions of succeeding platforms.

Release Phase

Not all of the security features may be implemented in the original version of the software.

[47] Defect analysis includes the followings: defect tracking information, defect type, phase where the defect is injected and removed, and time required to fix the defect.

[48] Unexpected software defects are not included.

However, mobile carriers need to understand what is possible and impossible after the deployment. For example, modifications related to mobile handsets (USIM) and network systems are quite challenging (e.g., changes in security algorithms and implementation of SQN management). Mobile carriers must clarify the differences (see Table 4-1) and evaluate potential risks, such as eavesdropping, masquerading, unauthorized network access, and manipulation of messages.

Substantial differences exist prior to and following deployment of services. Strict and sufficient field tests are crucial to ensure reliability, functionality, and performance. Unexpected problems associated with critical security features should be recognized and measures should be taken to minimize negative impacts before final release. In order to improve software quality and platform architecture, reschedule of a commercial release is acceptable. In addition, DoCoMo's case shows that security traffic consists largely of network traffic, and security features require high CPU performance. Mobile carriers should precisely evaluate the systems performance before deployment.

- Complexity in the platform should be reduced.
  ➔Rescheduling of a final release is acceptable.
- Original version does not have to acquire full security features, but mobile carriers should clarify the differences between specifications and implementation.
- Strict field testing is crucial to confirm security functions.
- Mobile carriers should evaluate the tradeoffs between security and performance.

5.2.2 After Deployment

Well-Secured Platform (Long/Short Term)

Sufficient security features that cover the full specifications of global standards will enable mobile carriers to focus on possible unauthorized access by intruders using other 2G or 3G networks. Compared with the security levels of other 2G and 3G networks, this network can be considered more secure. The risks of fraudulent network access are more likely from outside their own network. Mobile carriers can identify future threats that may not have been anticipated in the global specifications. This experience will help develop new security features in the global standards.

At the same time, sufficient security features require high CPU performance and database resources. Mobile carriers must evaluate the extensibility of the platform and resolve tradeoffs between security and performance. If the current platform lacks flexibility and modularity, mobile carriers should improve the platform architecture to satisfy future demands. Given the fully implemented security features, platform extension is desirable to avoid implementation risks.

- Risk assessment should focus on network traffic from other networks.
- Mobile carriers can address future threats.
- Platform extension is desirable when mobile carriers add some security features.

Secured Platform (Long/ Short Term)

Mobile carriers often simplify their security features in order to reduce costs and complexity. If a missing feature is not critical (e.g., if the platform supports only fixed-length parameters) and platform architecture is well designed, the existing platform will remain viable. As discussed earlier, security comprises a fundamental part of the platform. The risks involved in changing the platform sometimes exceed the benefits gained by adding or fixing security features. Thus, the criteria should be whether the system can detect and prevent replay attacks, whether eavesdropping allows intruders to identify the message protocol, and whether the system is flexible enough to upgrade the security features.

An analysis of network traffic is also necessary. Mobile carriers have to collect the network data from inside and outside of the network and check possible unauthorized access.

Mobile carriers also can ask a third party to objectively evaluate the security levels. At the same time, mobile carriers need to foster a tolerant culture for architectural change in the network software. It is essential to educate network engineers that architectural change in the platform does not imply fault on the part of the engineers.

- The current platform remains viable if missing security features are not critical.
- Evaluation criteria must be clear and objective.
- A tolerant culture should be fostered to facilitate necessary architectural changes.
- An analysis of network traffic (inside and outside the network) is required.

Vulnerable Platform (Short Term)

If the current platform lacks critical security features but the existing platform is modularized and flexible, mobile carriers should fix or add the missing security mechanisms. Given the implementation risks, platform extension is reasonable to fix security problems. At this point, however, implementation of new application services, such as multi-calling and videoconferencing, is not recommended. These new services require complex architecture and it becomes difficult to identify any defects associated with security features. This platform extension should focus only on upgrading security features, which means that mobile carriers sometimes have to delay the launch of new services in order to fix the security problems. Given today's severe market competition, this decision might be hard to accept. However, mobile carriers need to understand that an unsecured system ultimately will cost far more than any benefits gained from new services.

At the same time, mobile carriers should investigate why current situation happened. In DoCoMo's case, the 10-15% cost reduction rule, too-short lead time, and the requirement for extremely high reliability removed the incentives for architectural changes based on long-term platform thinking and resulted in skewed platform architecture. Consequently, authentication mechanisms were insufficiently implemented. Some reasons are understandable, but others are not. Mobile carriers must evaluate these findings and fix the fundamental problems.

- Platform extension for upgrading security features should occur separately from normal implementation of new application services.
- Mobile carriers should accept the delay in offering new services until updated security features are stabilized (2 to 6 months).
- Mobile carriers should determine why the existing platform is insufficient.

  ➔Problems should be fixed before applying platform extension.

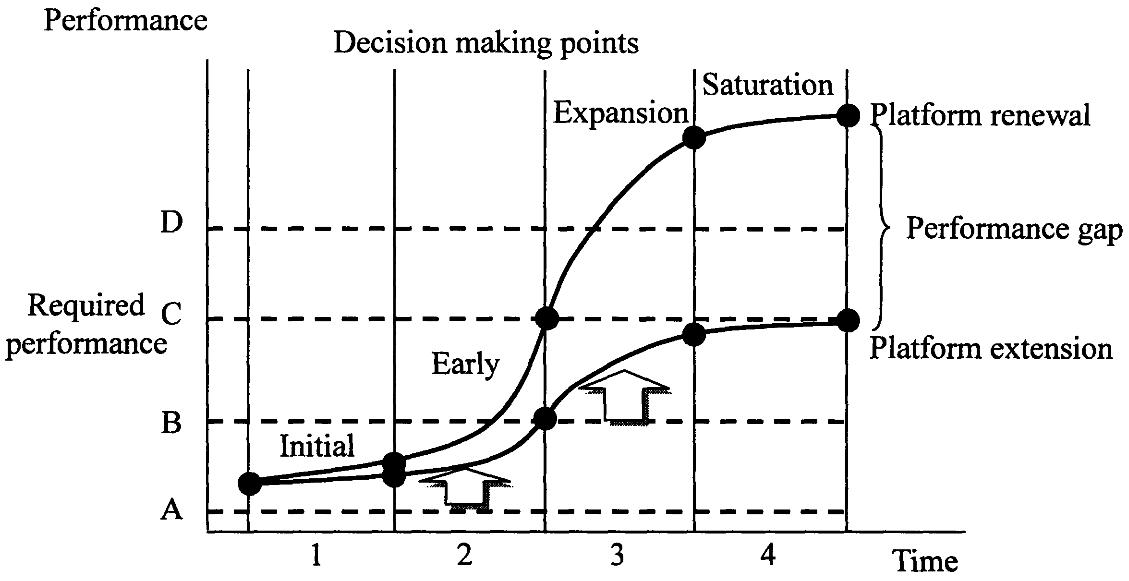
Vulnerable Platform (Long Term)

If the existing platform lacks critical security features and flexibility, mobile carriers have to rely on peripheral security mechanisms, such as traffic analysis, data trace, and protocol analysis. For example, if HLR receives an "Authentication Failure Report" from VLR/SGSN, the HLR can alert network operators who can identify useful specific information, such as telephone number (MSISDN), identification number (IMSI), time, and location. These operators can systematically trace the calls associated with this subscriber and analyze the message protocols. To be effective, a good system must be simple, accurate, and easy to use [75]. Utilization of fraud detection mechanisms (risk control) allows early detection of fraudulent communications. Mobile carries also should listen to the end users and concentrate on software quality from the customer perspectives [76] because unexpected threats are often implied in end-user claims.

However, this strategy is not a long-term solution. This platform will remain vulnerable. Replay attacks may decipher encrypted security messages and allow intruders to access the network. In addition, this platform cannot survive if the required performance rises. At present, the required performance is at Level "A" in Figure 5-6. As the number of 3G subscribers increases and new services are added, the required performance will rise to Level "B" or "C". The platform extension approach in an inflexible platform cannot provide sufficient performance for the future. In order to fix this fundamental problem, platform renewal is essential. However, platform renewal often results in substantial migration risks and organizational resistance to architectural change. As a first step toward platform renewal, mobile carriers should clarify the boundaries inside or outside of the subsystems (see Figure 5-7).

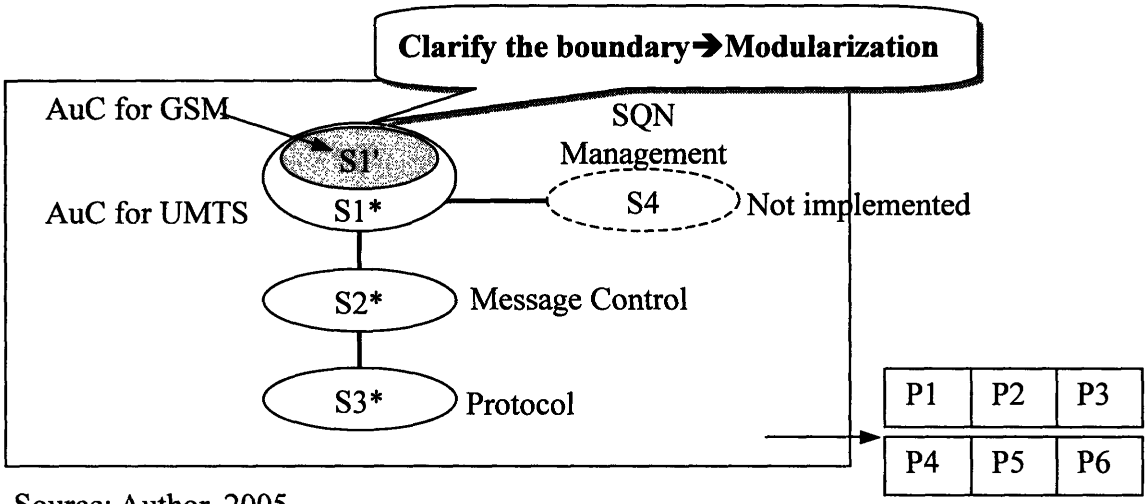
Given organizational resistance based on cultural and political reasons, platform renewal cannot be achieved without top-down decision making. Early decision making is better for platform renewal because it takes at least two years to accomplish such renewal. Before the required performance changes from Level "A" to "B" in Figure 5-6, mobile carriers have to (1) evaluate the capability of the platform (e.g., performance based on dynamic steps), and (2) forecast performance requirements for the future. Top executives also must stress that a statement such as "no fraudulent communications have been found so far" does not mean that the system is secured. Several years may be needed to reorient the mindset of network engineers and fulfill the platform renewal. However, this step is vital before mobile carriers encounter a wider range of fraudulent communications.

- Peripheral security mechanisms allow mobile carriers to detect fraudulent communications.
  ➔The detection mechanisms should be systematic.
- Claims from end users can help identify fraudulent communications.
- Platform renewal is essential as part of long-term strategy.
- Before platform renewal is undertaken, mobile carriers should clarify the boundaries of the subsystems.
- A top-down approach is required to mitigate organizational resistance and apply platform renewal.
- Early decision making is essential before required performance exceeds the maximum performance capability of the platform.



Source: Author, 2005.

Figure 5-6: Platform and Required Performance



Source: Author, 2005.

Figure 5-7: Clarify the Boundaries Inside or Outside of the Subsystems

6 Conclusion

Fraudulent communications are not new. Ever since the inception of the first generation mobile systems, mobile carriers have strived to prevent fraudulent communications. Using digital ciphering mechanisms, GSM systems have contributed to preventing fraudulent communications. However, some of the security mechanisms in GSM already have become insufficient and outdated. Given the fraudulent communications found in GSM networks, more advanced security mechanisms are required in 3G mobile systems. Security algorithms have become more sophisticated, and new and longer security parameters have been applied. New security operations have also been implemented to detect potentially fraudulent communications.

6.1 Security in 3G Mobile Systems

Network security ensures the consistency, integrity, and reliability of telecommunications systems, and authorized network access can prevent fraudulent communications and maintain the availability of each system. Currently, however, none of the 3G mobile systems is perfectly secured, for the following reasons.

1) For performance reasons, 3G mobile systems must rely on conventional security methods, which allow mobile carriers to reduce network delays but also make the system less secure.
2) Both secured and unsecured networks are interconnected by global roaming services. Fraudulent network access can be generated from outside networks that support weaker security mechanisms.[49]

[49] For example, China and some countries turn off encryption due to export restriction reasons.

3) As a practical matter, it is difficult to upgrade security features to protect against brand-new and unexpected network attacks that may not have been anticipated in the global specifications.

In addition, the 3G technical specifications, including network architecture, network protocols, and security algorithms, are widely available to the public. 3GPP also provides sample source codes and simulation data for security algorithms. Open-design architecture helps create better security standards, but at the same time today's sophisticated intruders can defeat telecommunication networks by applying the knowledge of 3G specifications. Thus, an inadequate system design will make mobile systems highly vulnerable to unauthorized access to mobile communications.

6.2 Lessons Learned in Actual 3G Mobile Systems

To date, no cloned mobile handsets and no fraudulent communications have been found in 3G mobile systems. However, the lessons learned from the existing network software (DoCoMo) suggest that the security mechanisms implemented today are imperfect and vulnerable. In particular, replay attacks to obtain network authentication can break security codes and allow fraudulent communications. DoCoMo's case provides us three important lessons:

1) Critical security features are sometimes missing in implementation (a clear gap exists between global standards and actual implementation).

2) Limited development time, cost reduction pressure and requirement for high reliability have forced mobile carriers to implement the insufficient and inflexible authentication mechanisms and skew the platform architecture.

3) Platform architecture has its limits. However, mobile carriers can take appropriate

measure by evaluating the extensibility of the platform. In order to satisfy performance requirements platform renewal should be applied.

6.3 Strategies for Mobile Carriers

Telecommunication is a basic, necessary service for individuals and corporations. Thus, security features must perform correctly on network software to protect against frauds. Even a single security breach can result in costly economic damage and serious failure in the critical infrastructure.

Prevention of such security threats remains a challenging issue. However, mobile carriers can reduce potential risks by understanding current platform conditions and addressing architectural problems beforehand.

Before Deployment

Substantial differences exist before and after deployment of the services. Mobile carriers must remember that architectural change is far more difficult to accomplish after launching the original platform.

Before deployment, mobile carriers should clarify the desired architectural design based on long-term platform thinking. The design phase has a direct impact on the future platform architecture. In this sense, the design phase is the most important phase. Not all of the security features may be implemented in the original version. However, mobile carriers have to understand what is possible and impossible after deployment.

Some security features require significant modifications in both mobile handsets and core network (HLR/AuC and VLR/SGSN). Upgrading the network software and renewing the USIM cards will be necessary. This upgrading is not easy because mobile handsets and core network systems are widely distributed. Serious efforts and several years will be necessary to resolve this situation. Until such problems are fixed, mobile carriers will have to accept some security holes in the network.

Mobile carriers also should evaluate whether the platform can survive during the system expansion phase. Modularization is essential to increase the flexibility and extensibility of the platform. Well-defined modularization helps localize implementation risks and increase systems reliability.

Before launching a system, mobile carriers should conduct strict field tests to ensure the reliability, functionality, and performance of security features. In order to improve software quality and the platform architecture, it is acceptable to reschedule a commercial release. After deployment, mobile carriers should take action to secure their own networks based on implemented levels of security and platform conditions.

After Deployment

Ideally, security features should be well designed and implemented, but as a practical matter, this situation is hard for mobile carriers to achieve. The security mechanisms implemented today tend to be imperfect and vulnerable. Based on platform conditions, four basic strategies can be categorized for securing mobile systems: (1) well-secured, (2) secured, (3) vulnerable (short term), and (4) vulnerable (long term).

First, in a well-secured platform, mobile carriers must focus on risk assessment. Compared with the security levels of other 2G and 3G networks, this network is more secure. Mobile carriers should analyze future threats coming from other 2G or 3G networks. This risk assessment will help develop new security features to meet global standards. At the same time, sufficient security features require high CPU performance and database resources. Evaluation of the tradeoffs between security and performance is necessary. With fully implemented security features, platform extension is desirable to avoid implementation risks.

Second, if missing security features are not especially critical, the current platform can be considered secure and will remain viable. Mobile carriers can ask a third party to objectively evaluate the security levels. The criteria must be: (1) the system can detect and prevent replay attacks, (2) messages cannot be identified through eavesdropping on traffic data, and (3) the system is flexible enough to upgrade its security features. Sometimes architectural innovation is necessary to achieve better security features. Therefore, mobile carriers should foster a tolerant culture for architectural change in the network software.

Third, even if the current platform is vulnerable, a modularized and flexible platform allows mobile carriers to fix or add the missing critical security features. In order to reduce implementation risks, platform extension should focus only on upgrading security features, otherwise the upgrade becomes complex, and it will be hard to identify the defects associated with security features. Today's severe market competition often prevents top executives from making this painful decision. However, mobile carriers need to understand that an unsecured system ultimately will cost far more than any benefits gained from new services.

Finally, if the existing platform lacks critical security features and flexibility, mobile carriers must rely on peripheral security mechanisms to secure the systems. Such fraud detection mechanisms should be systematic, and include traffic analysis, data trace, and protocol analysis to identify potential fraud. However, this strategy cannot be a long-term solution. The current platform will remain vulnerable. Replay attacks can decipher encrypted security messages and allow intruders to access the network. Furthermore, performance requirements may exceed the maximum performance capabilities of the platform owing to an increased number of subscribers and/or the deployment of new services. Platform renewal becomes essential as part of the long-term strategy. However, such renewal may cause substantial migration risks and encounter organizational resistance. Thus, early decision making by top management (using a top-down approach) is crucial. At the same time, mobile carriers should foster a tolerant culture for architectural changes in network software. Such a culture of tolerance will facilitate the necessary architectural changes. Even so, platform renewal will likely take at least two years to accomplish, but these steps are essential before mobile carriers encounter further serious and wide-ranging fraudulent communications

In the near future, communications systems such as wired or wireless networks, satellite systems, and the Internet will converge via TCP/IP. The boundaries among networks will diminish, and this convergence will allow many people to access these networks. New technology developments and telecommunications convergence will represent a major challenge to the UMTS security architecture.

Four years have passed since the first 3G mobile system was deployed. To date, no fraudulent communications have been found in 3G mobile networks. At the same time, however, no perfectly secured system exists in 3G mobile networks. History repeats itself. Just as mobile carriers experienced fraudulent communications in the first generation and the second generation networks, it is likely to confront similar situations in 3G mobile networks.

Telecommunication is an essential service in human life, and network security supports the fundamental features of telecommunications services. By addressing architectural and implementation problems beforehand, mobile carriers can manage unexpected security problems in the next generation networks and secure the systems against fraud in mobile communications.

END

Appendix

Abbreviations and Terminologies

1G:	the First generation mobile technologies
2G:	the second generation mobile technologies
3G:	the third generation mobile technologies
3GPP:	The Third Generation Partnership Project (UMTS Network)
3GPP2:	The Third Generation Partnership Project 2 (CDMA2000 Network)
4G:	the Forth generation mobile technologies
5G:	the Fifth generation mobile technologies
AMPS:	Advanced Mobile Phone System
AuC:	Authentication Center
bps:	bits per second
CDMA:	Code Division Multiple Access
CN:	Core Network
DB:	Database
EDGE:	Enhanced Data GSM Environment
EV-DO:	Evolution Data Only
EV-DV:	Evolution Data Voice
FDMA:	Frequency Division Multiple Access
FOMA:	Freedom Of Mobile multimedia Access
GGSN:	Gateway GPRS Support Node
GMSC:	Gateway Mobile Switching Center
GPRS:	General Packet Radio Service
GSM:	Global System for Mobile Communications
GSM-MAP:	Global System for Mobile Communications–Mobile Application Part
HLR:	Home Location Register
HSDPA:	High Speed Downlink Packet Access
IMT-2000:	International Mobile Telecommunications 2000
IP:	Internet Protocol (IPv4: IP version 4 and IPv6: IP version 6)
IS-95:	Interim Standard 95
ISDN:	Integrated Services Digital Network
ISUP:	Integrated Services Digital Network User Part
ITU:	International Telecommunication Union

ITU-T:	The ITU Telecommunication Standardization Sector
Kbps:	Kilobits per second
MAP:	Mobile Application Part
MAPsec:	Mobile Application Part security
MC-CDMA:	MultiCarrier-Code Division Multiple Access
MS:	Mobile Station
MSC:	Mobile services Switching Center
O&M:	Operation and Maintenance
OFDM:	Orthogonal Frequency Division Multiplexing
PDC:	Personal Digital Cellular
PDC-P:	Personal Digital Cellular Packet
PLMN:	Public Land Mobile Network
PSTN:	Public Switched Telephone Network
QoS:	Quality of Service
RNC:	Radio Network Controller
SCCP:	Signaling Connection Control Part
SCP:	Service Control Point
SGSN:	Serving GPRS Support Node
TACS:	Total Access Communication System
TC:	Transaction Capabilities
TDMA:	Time Division Multiple Access
UMTS:	Universal Mobile Telecommunications System
VLR:	Visitor Location Register
VoIP:	Voice over IP
W-CDMA:	Wideband CDMA

End Notes

[1] K. Boman, G. Horn, P. Howard, and V. Niemi, "UMTS Security," *Electronics & Communication Engineering Journal* 14, no. 5 (2002): 191-204.

[2] Michael Walker, Ray Forbes, and Dieter Gollman, "Security Aspects of Third Generation Mobile Telecommunications," IEE Colloquium on Mobility in support of Personal Communications, London, 16 June, 1993:5/1-5/4.

[3] A. Mehrotra and L. Golding, "Mobility and security management in the GSM system and some proposed future improvements," *Proceedings of the IEEE* 86, no. 7 (1998): 1480–1496.

[4] J. C. Cooke, and R. L. Brewster, "Cryptographic Security Techniques for Digital Mobile Telephones," IEE 2nd International Conference on Private Switching Systems and Networks, London, 23-25 June, 1992:123-130.

[5] Geir M. Koien, "An introduction to access security in UMTS," *IEEE Wireless Communications* 11, no. 1 (2004): 19-25.

[6] GSM Association, *Membership & Market Statistics*, GSMA, 2004.

[7] Ian Goldberg, "Cryptanalysis of the GSM Identification Algorithm," DefCon, 1988.

[8] Marc Meyer and Alvin Lehnerd, *The Power of Product Platforms Building Value and Cost Leadership* (New York: Free Press, 1997).

[9] 3rd Generation Partnership Project Technical Specification Group Core Network, *3GPP TS 23.002 V6.6.0 Network architecture*, 3GPP, 2004.

[10] 3rd Generation Partnership Project Technical Specification Group Core Network, *3GPP TS 09.02 V7.9.0 Mobile Application Part (MAP) specification*, 3GPP, 2001.

[11] 3rd Generation Partnership Project Technical Specification Group Core Network, *3GPP TS 29.002 V6.8.0 Mobile Application Part (MAP) specification*, 3GPP, 2004.

[12] Organization for Economic Co-operation and Development, *Guidelines for the Security of Information Systems*, OECD, 1992.

[13] Organization for Economic Co-operation and Development, *OECD Guidelines for the Security of Information Systems and Networks: Toward a Culture of Security*, Paris: OECD, 2002.

[14] P. Howard, M. Walker and T. Wright, "Towards a coherent approach to third generation system security," 3G Mobile Communication Technologies, 26-28 March 2001:21-27.

[15] 3rd Generation Partnership Project Technical Specification Group Core Network, *3GPP TS 33.102 V6.3.0 3G Security; Security architecture*, 3GPP, 2004.

[16] M. Zhang and Y. Fang, “Security analysis and enhancements of 3GPP authentication and key agreement protocol,” *IEEE Transactions on Wireless Communication* 4, no. 2 (2005): 734-742.

[17] The Wassenaar Agreement on export controls for conventional arms and dual-use goods and technology, http://www.wassenaar.org/.

[18] Auguste Kerckhoffs, “La cryptographie militaire,” *Journal des sciences militaires* IX (January 1883): 5-38, (February 1883): 161-191.

[19] F. L. Bauer, “Software Engineering,” *Information Processing* 71 (1972): 530-538.

[20] B. W. Boehm, “Software Engineering,” *IEEE Transactions on Computers* 25, no. 12 (1976): 1226-1241.

[21] R. Fairley, *Software Engineering Concepts* (New York: McGraw-Hill, 1985).

[22] Adrian A. Dolinski and Stuart E. Glickman, “Software Reliability Estimations for Telecommunications Products,” IEEE Journal on selected areas in communications 12, no. 2 (1994): 292-301.

[23] Ming-Yee Lai and Karl F. Rauscher, “Total reliability management for telecommunications software,” Technical Program Conference Record, IEEE in Houston. GLOBECOM '93, IEEE, vol1, 1993: 505-509.

[24] International Organization for Standardization, *ISO/IEC 9126-1, Software engineering -- Product quality -- Part 1: Quality model*, ISO, 2001.

[25] Robert L. Erickson, Dileep R. Saxena, and Gary G. Brush, “A View of Reliability and Quality Measurements for Telecommunications Systems,” *IEEE Journal on Selected Areas in Communications* 8, no. 2 (1990): 219-223.

[26] William W. Everett and Shinichi Honiden, “Guest Editors’ Introduction: Reliability and Safety of Real-Time Systems,” *IEEE Software* 12, no. 3 (1995): 13-16.

[27] Evan Marcus and Hal Stern, *Blueprints for High Availability: Designing Resilient Distributed Systems* (New York: John Wiley & Sons, 2000).

[28] Lawrence Bernstein, “Better software through operational dynamics,” *IEEE Software* 13, no. 2 (1996): 107-109.

[29] Pamela Zave and Michael Jackson, “A Component-Based Approach to Telecommunication Software,” *IEEE Software* 15, no. 5 (1998): 70-78.

[30] Nancy D. Griffeth and Yow-Jian Lin, “Guest Editors’ Introduction: Extending Telecommunications Systems: the Feature-Interaction Problem,” *IEEE Computer* 26, no. 8 (1993): 14-18.

[31] Leslie D. Fife, “Feature interaction-how it works in telecommunication software,” *IEEE Potentials* 15, no. 4 (1996): 35-37.

[32] Dirk O. Keck and Paul J. Kuehn, "The Feature Interaction Problem in Telecommunications Systems: A Survey," *IEEE Transactions on Software Engineering* 24, no. 10 (1998): 779-796.

[33] Aurel A. Lazar, "Programming Telecommunication Networks," *IEEE Network* 11, no. 5 (1997): 8-18.

[34] D. Schmidt, A. Gokhale, T. Harrison, and G. Parulkar, "A High-Performance End System Architecture for Real-Time CORBA," *IEEE Communications* 35, no. 2 (1997): 72–78.

[35] M. C. Chan and A. A. Lazar, "Connection Services on the xbind Broadband Kernel," OPENSIG Workshop on Open Signaling for ATM, Internet and Mobile Networks, University of Cambridge, England, 17–18 April, 1997.

[36] Bruce Sterling, *The Hacker Crackdown: Law and Disorder on the Electronic Frontier* (New York: Bantam Books, 1992).

[37] D. Robertson and K. Ulrich, "Planning for Product Platforms," *Sloan Management Review* 39, no. 4 (1998): 19-31.

[38] K. Ulrich, "The role of product architecture in the manufacturing firm," *Research Policy* 24, no. 3 (1995): 419-440.

[39] V. Krishnan and S. Gupta, "Appropriateness and impact of platform-based product development," *Management Science* 47, no.1 (2001): 52-68.

[40] A. K. Gupta, and W. E. Souder, "Key drivers of reduced cycle time," *Research Technology Management* 41, no. 4 (1998): 38-43.

[41] J. R. Hauser, "Metrics thermostat," *Journal of Product Innovation Management* 18, no. 3 (2001): 134-153.

[42] S. Sanderson and M. Uzumeri, "Managing product families: the case of the Sony Walkman," *Research Policy* 24, no. 5 (1995): 761-782.

[43] M. Uzumeri and S. Sanderson, "A framework for model and product family competition," *Research Policy* 24, no. 4 (1995): 583-607.

[44] M. A. Cusumano and R. W. Selby, *Microsoft Secrets: How The World's Most Powerful Software Company Creates Technology, Shapes Markets, And Manage People* (New York: Free Press, 1995).

[45] Annabelle Gawer and Michael A. Cusumano, *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation* (Boston: Harvard Business School Press, 2002).

[46] Allen B. Tucker and Barry W. Boehm, "Point/Counterpoint: On the Balance between Theory and Practice / Software Engineering Is a Value-Based Contact Sport," *IEEE Software* 19, no. 5 (2002): 94-97.

[47] N. Sundgren, "Introducing interface management in new product family development," *Journal of Product Innovation Management* 16, no. 1 (1999): 40-51.

[48] M. H. Meyer and J. M. Utterback, "The product family and the dynamics of core capability," *Sloan Management Review* 34, no. 3 (1993): 29-47.

[49] M. H. Meyer, P. Tertzakian, and J. M. Utterback, "Metrics for managing research and development in the context of the product family," *Management Science* 43, no. 1 (1997): 88-111.

[50] T. W. Simpson, J. R. A. Maier, and F. Mistree, "Product platform design: method and application," *Research in Engineering Design* 13, no. 1 (2001): 2-22.

[51] M. Meyer and L. Lopez, "Technology Strategy in a Software Products Company," *Journal of Product Innovation Management* 12, no. 4 (1995): 294–306.

[52] R. Henderson and K. B. Clark, "Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms," *Administrative Science Quarterly* 35, no. 1 (1990): 9-30.

[53] J. M. Utterback, *Mastering the Dynamics of Innovation* (Boston: Harvard Business School Press, 1994).

[54] Armen Zakarian and Gary J. Rushton, "Development of Modular Electrical Systems," *IEEE/ASME Transactions on Mechatronics* 6, no. 4 (2001): 507-520.

[55] R. Sanchez, "Fitting together a modular approach," *Manufacturing Engineer* 81, no. 5 (2002): 216-218.

[56] Christopher W. Alexander, *Notes on the Synthesis of Form* (Cambridge: Harvard University Press, 1970).

[57] Nam P. Suh, *The Principles of Design* (New York: Oxford University Press, 1990).

[58] C. Y. Baldwin and K. B. Clark, "Managing in an age of modularity," *Harvard Business Review* 75, no. 5 (1997): 84-93.

[59] Stephen R. Schach, *Object-Oriented and Classical Software Engineering*, 5th ed. (New York: McGraw-Hill, 2002).

[60] A. Hac, "Using a software reliability model to design a telecommunications software architecture," *IEEE Transactions on Reliability* 40, no. 4 (1991): 488-94.

[61] Frank J. van der Linden and Jürgen K. Müller, "Creating architectures with building blocks," *IEEE Software* 12, no. 6 (1995): 51-60.

[62] Frank van der Linden, and Jürgen K. Müller, "Composing Product Families from Reusable Components," Proceedings of the 1995 International Symposium and Workshop on Systems Engineering of Computer Based Systems, Tucson, AZ, 6-9 March 1995: 35-40.

[63] NTT DoCoMo, Inc, 1999-2003 Annual Report (Tokyo: NTT DoCoMo, 1999-2003).

[64] Jeffrey M. Voas, "Quality time - faster, better, and cheaper," *IEEE Software* 18, no. 3 (2001): 96-97.

[65] Michael Cusumano and David Yoffie, *Competing on Internet Time Lessons from Netscape and Its Battle with Microsoft* (New York: Free Press, 2000).

[66] David B. Yoffie and Michael A. Cusumano, "Judo strategy: the competitive dynamics of internet time," *Havard Business Review* 77, no. 1 (1999): 70-81.

[67] 3rd Generation Partnership Project Technical Specification Group Core Network, *3GPP TS 23.003 V6.5.0 Numbering, addressing and identification*, 3GPP, 2004.

[68] Deborah G. Ancona, Thomas Kochan, Maureen Scully, John Van Maanen, and D. Eleanor Westney, *Managing for the Future: Organizational Behavior and Processes*, 3rd ed. (Ohio: South-Western College Publishing, 2004).

[69] Susan T. Fiske and Shelley E. Taylor, *Social Cognition*, (New York: Random House, 1984).

[70] B. W. Boehm, *Tutorial: Software Risk Management* (New York: IEEE Computer Society Press, 1989).

[71] Nikolai N. Mansurov and Robert L. Probert, "A scenario-based approach to the evolution of telecommunications software," *IEEE Communications Magazine* 39, no. 10 (2001): 94-100.

[72] Alan MacCormack, Chris F. Kemerer, Michael A. Cusumano and Bill Crandall, "Trade-offs between Productivity and Quality in Selecting Software Development Practices," *IEEE Software* 20, no. 5 (2003): 78-85.

[73] Robert G. Mays, "Applications of Defect Prevention in Software Development," *IEEE Journal on Selected Areas in Communications* 8, no. 2 (1990): 164-168.

[74] R. H. Dunn, "Quality Assurance for Telecommunications Software," I*EEE Journal on Selected Areas in Communications* 4, no. 7 (1986): 1002-1008.

[75] Vijay Garg, David Ness-Cohn, Tim Powers and Larry Schenkel, "Direction for element managers and network managers," *IEEE Communications Magazine* 36, no. 10 (1998): 132-138.

[76] John P. Hudepohl, "Measurement of Software Service Quality for Large Telecommunications Systems," *IEEE Journal on Selected Areas in Communications* 8, no. 2 (1990): 210-218.

Bibliography

3rd Generation Partnership Project Technical Specification Group Core Network. *3GPP TS 09.02 V7.9.0 Mobile Application Part (MAP) specification*. 3GPP, 2001.

———. *3GPP TS 23.002 V6.6.0 Network architecture*. 3GPP, 2004.

———. *3GPP TS 23.003 V6.5.0 Numbering, addressing and identification*. 3GPP, 2004.

———. *3GPP TS 29.002 V6.8.0 Mobile Application Part (MAP) specification*. 3GPP, 2004.

———. *3GPP TS 33.102 V6.3.0 3G Security; Security architecture*. 3GPP, 2004.

Alexander, Christopher W. *Notes on the Synthesis of Form*. Cambridge: Harvard University Press, 1970.

Ancona, Deborah G., Thomas Kochan, Maureen Scully, John Van Maanen, and D. Eleanor Westney. *Managing for the Future: Organizational Behavior and Processes*. 3rd ed. Ohio: South-Western College Publishing, 2004.

Baldwin, C. Y., and K. B. Clark. "Managing in an age of modularity." *Harvard Business Review* 75, no. 5 (1997): 84-93.

Bauer, F. L. "Software Engineering." *Information Processing* 71 (1972): 530-538.

Bernstein, Lawrence. "Better software through operational dynamics." *IEEE Software* 13, no. 2 (1996): 107-109.

Boehm, B. W. "Software Engineering." *IEEE Transactions on Computers* 25, no. 12 (1976): 1226-1241.

———. *Tutorial: Software Risk Management*. New York: IEEE Computer Society Press, 1989.

Boman, K., G. Horn, P. Howard, and V. Niemi. "UMTS Security." *Electronics & Communication Engineering Journal* 14, no. 5 (2002): 191-204.

Chan, M. C., and A. A. Lazar. "Connection Services on the xbind Broadband Kernel." OPENSIG Workshop on Open Signaling for ATM, Internet and Mobile Networks, University of Cambridge, England, 17–18 April, 1997.

Cooke, J. C., and R. L. Brewster. "Cryptographic Security Techniques for Digital Mobile Telephones." IEE 2nd International Conference on Private Switching Systems and Networks, London, 23-25 June, 1992:123-130.

Cusumano, M. A., and R. W. Selby. *Microsoft Secrets: How The World's Most Powerful Software Company Creates Technology, Shapes Markets, And Manage People*. New York: Free Press, 1995.

Cusumano, Michael, and David Yoffie. *Competing on Internet Time Lessons from Netscape and Its Battle with Microsoft*. New York: Free Press, 2000.

Dolinski, Adrian A., and Stuart E. Glickman. “Software Reliability Estimations for Telecommunications Products.” IEEE Journal on selected areas in communications 12, no. 2 (1994): 292-301.

Dunn, R. H. “Quality Assurance for Telecommunications Software.” *IEEE Journal on Selected Areas in Communications* 4, no. 7 (1986): 1002-1008.

Erickson, Robert L., Dileep R. Saxena, and Gary G. Brush. “A View of Reliability and Quality Measurements for Telecommunications Systems.” *IEEE Journal on Selected Areas in Communications* 8, no. 2 (1990): 219-223.

Everett, William W., and Shinichi Honiden. “Guest Editors’ Introduction: Reliability and Safety of Real-Time Systems.” *IEEE Software* 12, no. 3 (1995): 13-16.

Fairley, R. *Software Engineering Concepts*. New York: McGraw-Hill, 1985.

Fife, Leslie D. “Feature interaction-how it works in telecommunication software.” *IEEE Potentials* 15, no. 4 (1996): 35-37.

Fiske, Susan T., and Shelley E. Taylor. *Social Cognition*. New York: Random House, 1984.

Garg, Vijay, David Ness-Cohn, Tim Powers and Larry Schenkel. “Direction for element managers and network managers.” *IEEE Communications Magazine* 36, no. 10 (1998): 132-138.

Gawer, Annabelle, and Michael A. Cusumano. *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*. Boston: Harvard Business School Press, 2002.

Goldberg, Ian. “Cryptanalysis of the GSM Identification Algorithm.” DefCon, 1988.

Griffeth, Nancy D., and Yow-Jian Lin. “Guest Editors’ Introduction: Extending Telecommunications Systems: the Feature-Interaction Problem.” *IEEE Computer* 26, no. 8 (1993): 14-18.

GSM Association. *Membership & Market Statistics.* GSMA, 2004.

Gupta, A. K., and W. E. Souder. “Key drivers of reduced cycle time.” *Research Technology Management* 41, no. 4 (1998): 38-43.

Hac, A. “Using a software reliability model to design a telecommunications software architecture.” *IEEE Transactions on Reliability* 40, no. 4 (1991): 488-94.

Hauser, J. R. “Metrics thermostat.” *Journal of Product Innovation Management* 18, no. 3 (2001): 134-153.

Henderson, R., and K. B. Clark. “Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms.” *Administrative Science Quarterly* 35, no. 1 (1990): 9-30.

Howard, P., M. Walker and T. Wright. “Towards a coherent approach to third generation system security.” 3G Mobile Communication Technologies, 26-28 March 2001:21-27.

Hudepohl, John P. “Measurement of Software Service Quality for Large Telecommunications

Systems." *IEEE Journal on Selected Areas in Communications* 8, no. 2 (1990): 210-218.

International Organization for Standardization. *ISO/IEC 9126-1, Software engineering -- Product quality -- Part 1: Quality model.* ISO, 2001.

Keck, Dirk O., and Paul J. Kuehn. "The Feature Interaction Problem in Telecommunications Systems: A Survey." *IEEE Transactions on Software Engineering* 24, no. 10 (1998): 779-796.

Kerckhoffs, Auguste. "La cryptographie militaire." *Journal des sciences militaires* IX (January 1883): 5-38, (February 1883): 161-191.

Koien, Geir M. "An introduction to access security in UMTS." *IEEE Wireless Communications* 11, no. 1 (2004): 19-25.

Krishnan, V., and S. Gupta. "Appropriateness and impact of platform-based product development." *Management Science* 47, no.1 (2001): 52-68.

Lai, Ming-Yee, and Karl F. Rauscher. "Total reliability management for telecommunications software." Technical Program Conference Record, IEEE in Houston. GLOBECOM '93, IEEE, vol1, 1993: 505-509.

Lazar, Aurel A. "Programming Telecommunication Networks." *IEEE Network* 11, no. 5 (1997): 8-18.

MacCormack, Alan, Chris F. Kemerer, Michael A. Cusumano and Bill Crandall. "Trade-offs between Productivity and Quality in Selecting Software Development Practices." *IEEE Software* 20, no. 5 (2003): 78-85.

Mansurov, Nikolai N., and Robert L. Probert. "A scenario-based approach to the evolution of telecommunications software." *IEEE Communications Magazine* 39, no. 10 (2001): 94-100.

Marcus, Evan, and Hal Stern. *Blueprints for High Availability: Designing Resilient Distributed Systems*. New York: John Wiley & Sons, 2000.

Mays, Robert G. "Applications of Defect Prevention in Software Development." *IEEE Journal on Selected Areas in Communications* 8, no. 2 (1990): 164-168.

Mehrotra, A., and L. Golding. "Mobility and security management in the GSM system and some proposed future improvements." *Proceedings of the IEEE* 86, no. 7 (1998): 1480–1496.

Meyer, M. H., and J. M. Utterback. "The product family and the dynamics of core capability." *Sloan Management Review* 34, no. 3 (1993): 29-47.

Meyer, M. H., P. Tertzakian, and J. M. Utterback. "Metrics for managing research and development in the context of the product family." *Management Science* 43, no. 1 (1997): 88-111.

Meyer, M., and L. Lopez. "Technology Strategy in a Software Products Company." *Journal of Product Innovation Management* 12, no. 4 (1995): 294–306.

Meyer, Marc, and Alvin Lehnerd. *The Power of Product Platforms Building Value and Cost Leadership*. New York: Free Press, 1997.

NTT DoCoMo, Inc. 1999-2003 Annual Report. Tokyo: NTT DoCoMo, 1999-2003.

Organization for Economic Co-operation and Development. *Guidelines for the Security of Information Systems*. Paris: OECD, 1992.

———. *OECD Guidelines for the Security of Information Systems and Networks: Toward a Culture of Security*. Paris: OECD, 2002.

Robertson, D., and K. Ulrich. "Planning for Product Platforms." *Sloan Management Review* 39, no. 4 (1998): 19-31.

Sanchez, R. "Fitting together a modular approach." *Manufacturing Engineer* 81, no. 5 (2002): 216-218.

Sanderson, S., and M. Uzumeri. "Managing product families: the case of the Sony Walkman." *Research Policy* 24, no. 5 (1995): 761-782.

Schach, Stephen R. *Object-Oriented and Classical Software Engineering*. 5th ed. New York: McGraw-Hill, 2002.

Schmidt, D., A. Gokhale, T. Harrison, and G. Parulkar. "A High-Performance End System Architecture for Real-Time CORBA." *IEEE Communications* 35, no. 2 (1997): 72–78.

Simpson, T. W., J. R. A. Maier, and F. Mistree. "Product platform design: method and application." *Research in Engineering Design* 13, no. 1 (2001): 2-22.

Sterling, Bruce. *The Hacker Crackdown: Law and Disorder on the Electronic Frontier*. New York: Bantam Books, 1992.

Suh, Nam P. *The Principles of Design*. New York: Oxford University Press, 1990.

Sundgren, N. "Introducing interface management in new product family development." *Journal of Product Innovation Management* 16, no. 1 (1999): 40-51.

The Wassenaar Agreement on export controls for conventional arms and dual-use goods and technology. http://www.wassenaar.org/.

Tucker, Allen B., and Barry W. Boehm. "Point/Counterpoint: On the Balance between Theory and Practice / Software Engineering Is a Value-Based Contact Sport." *IEEE Software* 19, no. 5 (2002): 94-97.

Ulrich, K. "The role of product architecture in the manufacturing firm." *Research Policy* 24, no. 3 (1995): 419-440.

Utterback, J. M. *Mastering the Dynamics of Innovation*. Boston: Harvard Business School Press, 1994.

Uzumeri, M., and S. Sanderson. "A framework for model and product family competition." *Research Policy* 24, no. 4 (1995): 583-607.

Van der Linden, Frank J., and Jürgen K. Müller. "Creating architectures with building blocks." *IEEE Software* 12, no. 6 (1995): 51-60.

———. "Composing Product Families from Reusable Components." Proceedings of the 1995 International Symposium and Workshop on Systems Engineering of Computer Based Systems, Tucson, AZ, 6-9 March 1995: 35-40.

Voas, Jeffrey M. "Quality time - faster, better, and cheaper." *IEEE Software* 18, no. 3 (2001): 96-97.

Walker, Michael, Ray Forbes, and Dieter Gollman. "Security Aspects of Third Generation Mobile Telecommunications." IEE Colloquium on Mobility in support of Personal Communications, London, 16 June, 1993:5/1-5/4.

Yoffie, David B., and Michael A. Cusumano. "Judo strategy: the competitive dynamics of internet time." *Havard Business Review* 77, no. 1 (1999): 70-81.

Zakarian, Armen, and Gary J. Rushton. "Development of Modular Electrical Systems." *IEEE/ASME Transactions on Mechatronics* 6, no. 4 (2001): 507-520.

Zave, Pamela, and Michael Jackson. "A Component-Based Approach to Telecommunication Software." *IEEE Software* 15, no. 5 (1998): 70-78.

Zhang, M., and Y. Fang. "Security analysis and enhancements of 3GPP authentication and key agreement protocol." *IEEE Transactions on Wireless Communication* 4, no. 2 (2005): 734-742.