韓國 OA學會 論文誌
第3卷 第4號, 1998. 12.

98-3-4-3-6

개방형 분산 컴퓨팅 시스템에서의 C-API 메타니즘 개발에 관한 연구

이 상 기* 최 용 락**

A Development of C-API Mechanism for Open Distributed Computing Systems

Lee Sang-gi* Choi Yong-rak**

요 약

본 논문은 개방형 분산 Computing system에서 여러 분산 어플리케이션 프로그래머들에게 범용 암호 서비스 제공을 위한 C-API(Cryptographic-Application Program Interface) Mechanism개발에 대하여 설명한다.

C-API Mechanism는 응용 프로그래머에게 공통적으로 사용 할 수 있는 암호 알고리즘 및 인터페이스를 제공함으로써 프로그래머가 암호 알고리즘을 알지 못하여도 분산 어플리케이션이 보안 서비스를 제공할 수 있다. 따라서, 본 논문에서는 다양한 응용 환경 또는 시스템 하부구조에 독립적으로사용될 수 있는 공용 암호 서비스 구조를 설계하여 공통적으로 이용할 수 있도록 하였다. 이러한 구조는 응용 프로그래머에게 각종 암호화 관련서비스 및 키관리 서비스를 응용 프로그램과 운영체제의 제약없이 사용할 수 있는 장점이다.

Abstract

This paper describes a C-API(Cryptographic-Application Program Interface) mechanism that can serve cryptographic service to one or more application programmers in an open distributed computing system.

Generic cryptographic service, provides application programmers with cryptographic algorithms and interfaces which can be shared so that the programmers can program distributed applications containing security services even though they have no detailed knowledge of cryptographic algorithms. Therefore, in this paper, a generic C-API mechanism is designed that can be used independently from various application environments and basic system structures so that programmers can use it commonly. This mechanism has the advantage that allows application programmers be able to use some cryptographic services and key management services not considering of the application program and operating system.

* 대전대학교 컴퓨터공학과 박사과정
** 대전대학교 컴퓨터공학과 교수

논문접수 : 98.11.9. 심사완료 : 98.12.11.

1. INTRODUCTION

Computing environment is under change from isolated to network environment. This gives us many advantages including sharing resources, quick information exchange, but also disadvantages like tapping transmitting information, invading and destroying private information, or illegal accessing to systems. To prevent these disadvantage, applications in distributed environment need security services, for example, keeping secrecy, authentication, blocking deny, control access.[1,4] But to install these kinds of security services to each application program, all the application programmers have got to be familiar with security policies and cryptographic algorithms, causing much redundant efforts for same purpose from all the programmers. To avoid this redundancy, a generic security service interface which can be used for all application is required. Therefore in this paper We will analyze the relationship between GSS-API(Generic Security Service Application Program Interface) and GCS-API(Generic Cryptographic Service Application Program Interface) which are suggested as generic security service system for distributed environment by IETF(Internet Engineering Task Force) and will suggest a method to study C-API for generic cryptographic service.

2.. The Structure of Generic Security Service

IETF suggested two standards of generic security services for open distributed environment, GSS-API and GCS-API, as shown in fig. 1.

In Fig.1, the application programs on top level request data protection services through the middle and bottom level services. In this case, as the middle level , GSS-API provide easy and useful interface for distributed application and as the bottom level, GCS-API is charge of the detailed responsibility of cryptographic security policy. [3]

2.1 GSS-API

GSS-API provides generic security services for distributed environment. GSS-API consists of two parts-interfaces for security services and a logical mechanism that provides security services. Interfaces reside in local systems and provide an access for it through libraries linked to application codes. A logical mechanism is transparent to application and usually resides on reliable the third remote security host Therefore the data transmission between Interfaces and mechanism causes network traffic. So network traffic needs authentication and key management mechanism and must provide transparency to applications. [2,5] Fig.2. shows the principle of distribution between application and GSS-API server.

2.2 GCS-API

GCS-API is the infrastructure that makes it possible for network application programmer to receive security services even they have little knowledge of security mechanism. But GCS-API itself do not support any cryptographic algorithms or key exchange algorithms but when

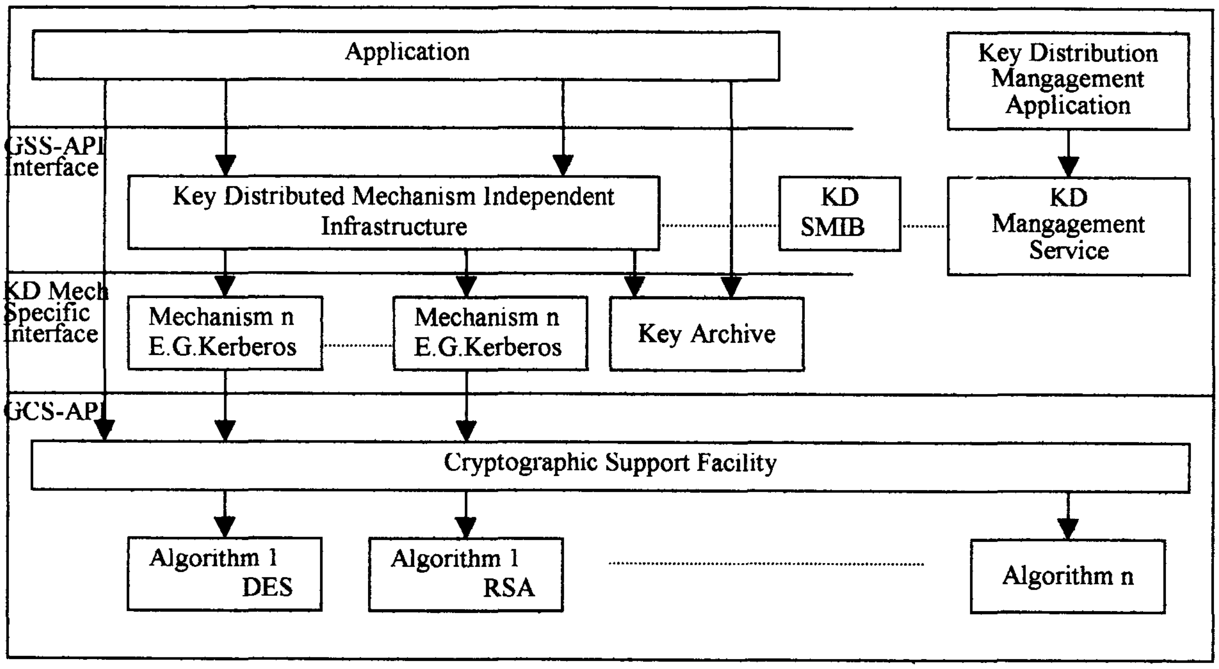


Fig.1. Hierchical structure of cryptographic service

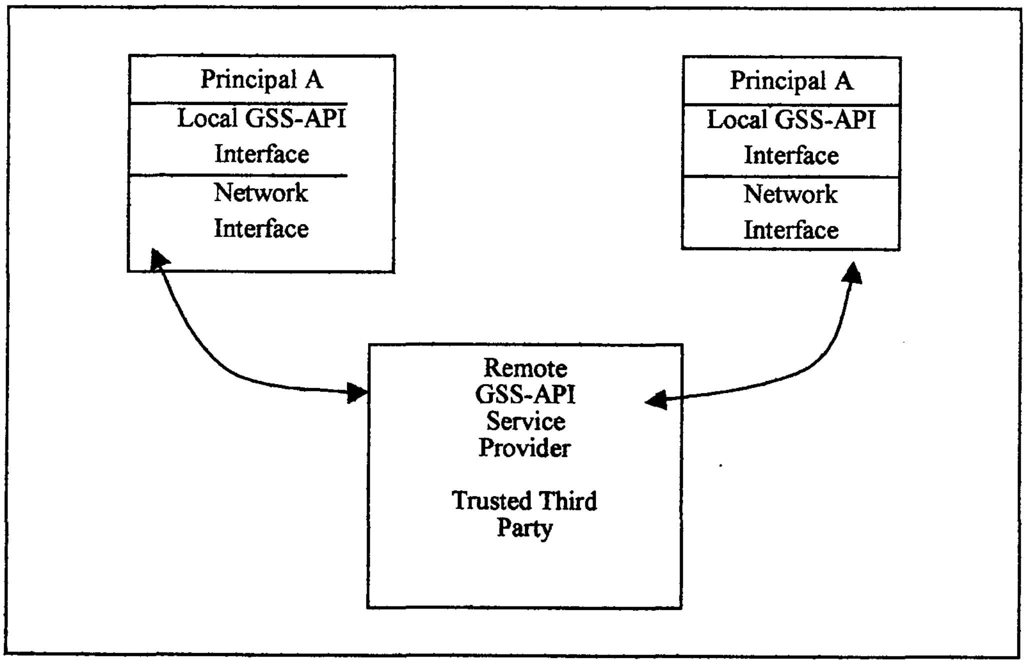


Fig.2. GSS-API distributed structure

<**Table 1:** Classifications of CSF functions>

Classification	Contents
Session Management	**Set/Unset the session between application and CSF**
gcs_initialize_session	For initializing session
gcs_terminate_session	Cut the session and unset the security connection
Cryptographic Context Retrieval Function	**For handling CC**
gcs_delete_cc	Remove/Unset the handling of CC
gcs_list_cc	Providing searching of CC
gcs_retrieve_cc	Providing use and searching of CC
Key Creation	**For creating key**
gcs_derive_key	Getting key from input parameter
gcs_generate_key	Generating key value
Hash and Signature Function	**For providing integration and digital sign.**
gcs_generate_checkvalue	Creating digital sign.
gcs_verify_checkvalue	Verifying digital sign.
gcs_verify_hash	Generating hash value
gcs_generate_random_number	Generating random number
Data Encipherment Function	**For providing encryption of data**
gcs_encipher_data	Providing encryption for data
gcs_decipher_data	Providing decryption for data
gcs_protect_data	Providing encryption, digital sign, authentication
gcs_decipher_verify	Providing decryption, digital sign, authentication
Cryptographic Context Storage Function	**For providing CC store and delete**
gcs_store_cc	Store CC and assign defined name
gcs_remove_cc	Remove CC from CSF

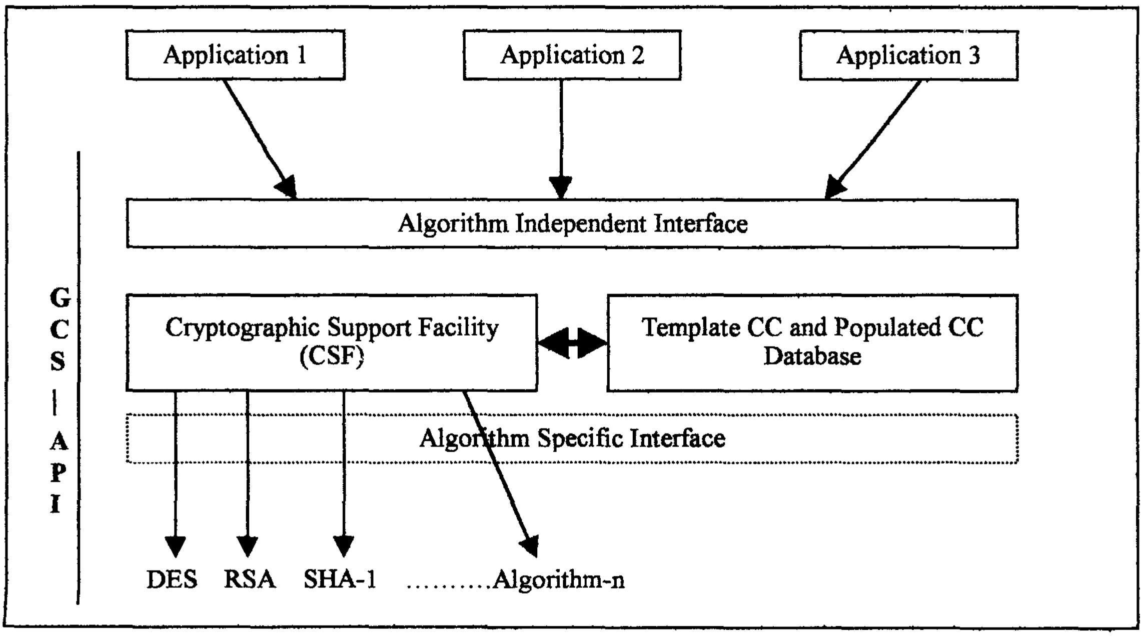

Application 1
Application 2
Application 3
Algorithm Independent Interface
GCS-API
Cryptographic Support Facility (CSF)
Template CC and Populated CC Database
Algorithm Specific Interface
DES
RSA
SHA-1
..........Algorithm-n

Fig.3.. Basic GCS-API model

an application request secure services, it calls requested cryptographic algorithms and provide them to the call application. GCS-API's outline structure is shown in Fig.3. When an application requests an information protection services, Cryptographic Support Facility(CSF) calls needed cryptographic service after referencing Cryptographic Context(CC).

In Fig.3. , each applications is an entity that requests information protection. GCS-API provide two conventions for compatible application development. First, it is independent from algorithm. GCS-API conceals the inner complicated parts of algorithms so that the caller can be provided cryptographic services even he or she doesn't know about cryptographic algorithms or parameters. Second, it is independent from execution. That is, GCS-API conceals the detailed execution to callers so that the callers can be provided security services regardless of where the cryptographic algorithms execute, software or hardware.

And GCS-API manage a database called CC for information of security services. Generally, to execute cryptographic algorithms, you need not only the information about data and key, also need some parameters related with the information about which algorithms will be used, and how they are used. Therefore CC maintains all information of cryptographic operation in capsule so that GCS-API can execute the algorithms independently.[6]

3. GCS-API Design

GCS-API structure consists of 3 large parts. First is CSF part which maintains interface with applications. Second is CC part which manages all the information of cryptographic services. The last is Cryptographic algorithm part.

3.1CryptographicSupportFacility(CSF)

3.1.1 Service Support Function

CSF is a set of interface. That is when applications request a security service, CSF provide security services by calling adequate cryptographic algorithms. CSF services consists of functions for initializing CSF, key management function, function for protecting transmitting data as shown in Table.1.

3.1.2 Function Authentication Policy

To use specific function, you have got to have an illegal access authentication. In GCS-API the authentication policy is defined by GCS-API function and access authentication and it uses a specific key. GCS-API callers are to provided authentication for an access to keys generated by subjects that is being operated by callers or for an access keys that the subjects that is being generated by caller have an authentication. The mechanism how this authentication policy is executed and managed is dependent on each constructions. The support for initializing session between caller and CSF is included the constructions specification and it is authenticated by the caller's ID and an adequate control information is defined.

The functions the caller executes on a specific key is determined by the authentication policy based on an assigned set of function to the caller of GCS-API. These functions are related with the caller rather than a subject which takes the place of the caller's action. The caller must control the policy that a function which was provided additional authentication allow a service a specific subject requested. Defined authentication policy is as followed.

- **GCS_C_ENCYPHER_DECIPHER**
  This authentication allows the caller to use gcs_encipher_data( ) and gcs_decipher_data( ). The use of those functions will be restricted by the deployment of cryptographic services and legal control.
- **GCS_C_SELECTION**
  This authentication allows the caller to use Protected Key Management function (Excluding setting and modifying the key use policy)
- **GCS_C_KEY PROTECTION**
  This authentication allows the caller to use Clear Key Management function.

3.1.3 Security Policy

A special control must be applied for cryptographic algorithms caused by important job in system security. In many nations, the export of cryptography software is restricted by laws. For example, ITAR(USA Government International Traffic in Arms Regulations) restricts exports of products including cryptographic services. And actually, data-cryptographic services are supplied and controlled domestically. [7,8]

The implementation of CSF must consider rigorous secure requirement as followed.

- CSF must prevent non-authentication access to cryptographic services.
- CSF must prevent the access to an private or secrete data.
- CSF must verify the control information related with keys before they are used.
- According to the policy, CSF must require the caller to be authenticated before it accesses to the requested services. Therefore, both cryptographic services and cryptographic products will contain authentication and access information.
- GCS-API's advantage is the fact that keys are never referenced by any nonauthentication caller in disclosed forms. At the upper level of CSF interface, operation keys are protected(for example, encryption with CSF master-key). Authenticated callers are key distribution services who need to connect operation keys with other related information to create mechanism-defined tokens. And, the fact that CSF access control's sub-version has more important security meaning for key management service interface rather than general application cryptographic service interface is worth notice.

3.2 Cryptographic service caller and system configuration

3.2.1 Cryptographic service caller

Cryptographic service callers are classified as followed according to cryptographic related policy and the using level of keys.

① Cryptographic unaware

Cryptographic services can be called by Cryptographic unaware who do not know about any detailed the services. Cryptographic callers will require encryption and integration services for entities like files and messages to application infrastructure provider. The callers don't know how those protection services act. That is, the transmission type, generating encryption or check value, if a cryptographic algorithm used symmetric/asymmetric cryptographic algorithms.

② Cryptographic Aware

Cryptographic aware callers have quite knowledge about detailed parts of cryptographic services. Therefore, they know if the data is being enciphering, if check values are being generating. Cryptographic aware callers may or followed to cryptographic policy aware and cryptographic policy unaware.

- Cryptographic Policy Unaware
- Cryptographic Policy Aware
- Cryptographic Policy Selecting
- Cryptographic Policy Enforcing

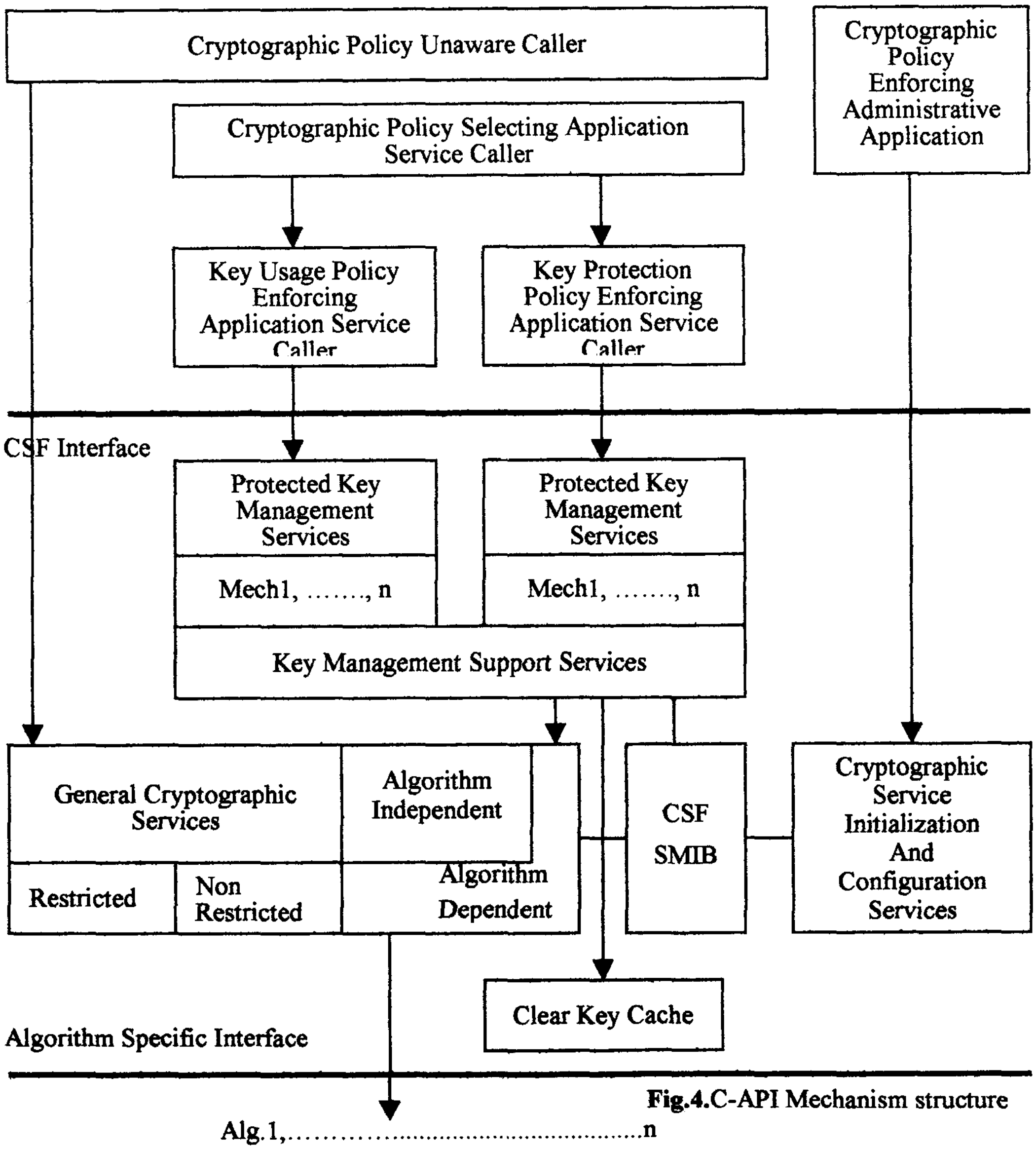
  Key Usage Policy Enforcing

  Key Protection Policy Enforcing

3.2.2 C-API Mechanism Support System

C-API Mechanism Support System for cryptographic services caller can be shown as Fig.4.

At the top level, applications request data protection services through middle and infrastructure services. Usually, these applications don't know about cryptography. The middle and infrastructure are charge of detailed parts of security policy and set the cryptography context. The applications go down by selecting specific key distribution protocol and algorithm and start with key distribution services independent from C-API mechanism to be a part of safe session setting.



Cryptographic Policy Unaware Caller
Cryptographic Policy Enforcing Administrative Application
Cryptographic Policy Selecting Application Service Caller
Key Usage Policy Enforcing Application Service Caller
Key Protection Policy Enforcing Application Service Caller
CSF Interface
Protected Key Management Services
Mech1, ..., n
Protected Key Management Services
Mech1, ..., n
Key Management Support Services
General Cryptographic Services
Algorithm Independent
Restricted
Non Restricted
Algorithm Dependent
CSF SMIB
Cryptographic Service Initialization And Configuration Services
Clear Key Cache
Algorithm Specific Interface
Alg.1,...n

Fig.4.C-API Mechanism structure

The services of this specification are implemented in CSF. The boundaries shown in other level interfaces are also important. CSF interface represents the check point for the possibility that cryptographic keys are manipulated in disclosed form by non-authentication callers whose cryptographic keys are not stored. The keys protected cryptographically are treated and referenced intransparently.

CSF provides generic cryptographic set and key management services interfaces located upper other different algorithm and various implementations of that algorithm. CSF service interfaces can hide specific algorithms

CSF provides the followings for applications and infrastructure of applications.

- A support for driving of given cryptographic transformations or key management operations
- It allows you not to consider if the detailed parts of operations or techniques are implemented by software or hardware
- It allows you to choose if it makes documents of given operation Quality of protection.
- It allows you to choose if a specified cryptographic algorithm makes documents of using operations

As shown in above figure, CSF provides the two programming interfaces, API and SPI, between various kinds of cryptographic aware callers and following services forms.

API consists of interfaces for generic cryptographic services and key protection management services.

- Generic Cryptographic Services This provides encryption, decryption, generation of check value, verification of check value and is executed by CSF callers and embedded CSF function for key management support
- Protected Key Management Support Services This provides generation, store, distribution services to cryptographic policy selection callers and key using policy execution callers

SPI consists of key management services.

- Clear Key Management Support Services This provides generation, store, distribution services of disclosed key to key protection policy execution callers

3.3 Cryptographic Context(CC)

CC can be considered as an database which maintains all the information for cryptographic operations execution in capsule style. CC outline structure is shown in Fig.5. CC contains the contexts of algorithm identifiers, algorithm

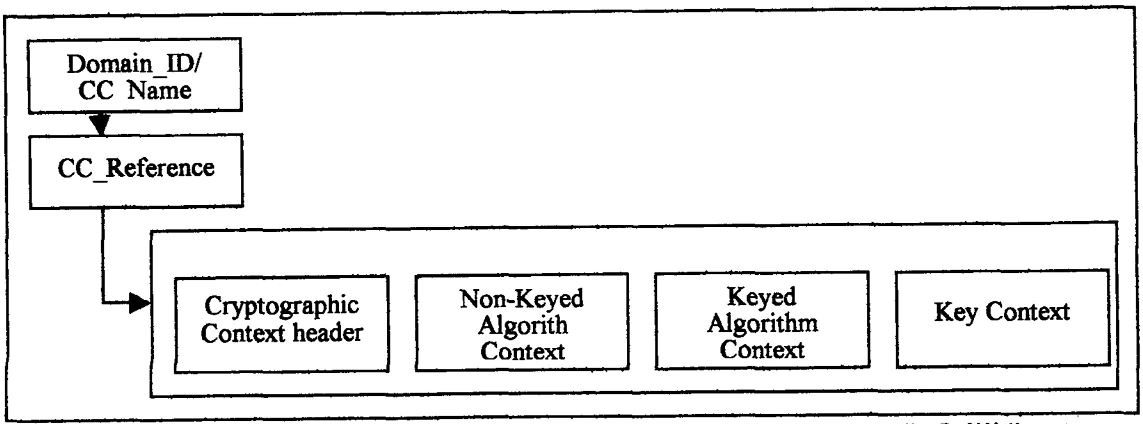


Fig.5. CC Structure

restricts like difference of key length in each algorithm through CC. There are two kinds of CC, Template CC and Populated CC. Template CC is generated by CSF manager and usually used for security policy setting for a domains.

Therefore, template CC only restricts available cryptographic mechanism for application, can not be used directly, and has no keys. Populated CC is generated by GCS-API using Template CC, has keys, and can be used direct cryptographic services.[8,9,10]

4. CONCLUSIONS

Todays computing environment is constructed within the massive network based on worldwide open distributed computing systems. This topology provides an easy-sharing of resources, overcoming of time and space restrictions. But also has many security vulnerable issues. Therefore, each application programmers needs security services, and this paper describes a way how to design generic C-API mechanism service structures which can be used independently from various applications and system infrastructures and a way how to use it commonly.

This structure can provide programmers with various services like encryption, decryption, integrity check, generating hash and random number and key management services without restrictions caused by application programs and operating systems.

In the future, by broadly applying the relations of public key infrastructure and other security related mechanism, more ideal service support systems, both generic and portable, must be studied.

참고문헌

[1] William Stalling, "Network and Internetwork Security", Prentice Hall, 1995

[2] D.P. Barton, L.J. O'Conner, "Implementing Generic Security Services in a Distributed Environment", May, 1995

[3] X/Open Company Ltd., "Generic Cryptographic Service API(GCS-API) Base-Draft 8", X/Open Preliminary Specification, April, 1996.

[4] Sead Muftic, Morris Sloman, "Security architecture for distributed systems", computer communications volume 17 number 7, July, 1994

[5] Per Kaijser, Tom Parker, Denis Pinkas, "SESAME: The solution to security for open distributed systems", computer communications volume 17 number 7, July. 1994

[6] W. Caelli, I. Graham, L. O'Conner, "Cryptographic Application Programming Interfaces(API)", Computer & Security, 1993.

[7] T. K. Park, C. K. Kang, " Security Management of Network System", communication information security volume 6, number 3, September. 1996

[8] J.H. Park. "Security framework on Information Super Highway", communication information security volume 6, number 3, September. 1996.

[9] J. O. Jung, "System and Network Security on Information Super Highway", information science, volume 14, number 3, March. 1996.

[10] N.A. Nazario, " Security policies for the federal public key infrastructure", 19th NISSC, baltimore, oct. 22-24, 1996.

저자소개

최용락

현재 : 대전대학교 컴퓨터공학과 교수

관심분야 : 컴퓨터보안, 전자상거래, 시스템인증

이상기

현재 : 대전대학교 컴퓨터공학과 박사과정수료

관심분야 : 컴퓨터보안및인증, CALS/EC, 기업경영혁신과정보기술