# CORBA 환경에서 개선된 QoP 관리 및 제어모델

## (An Enhanced QoP Management and Control Model in CORBA Environments)

이 희 종 † 이 승 룡 †† 전 태 웅 †††

(Heejong Lee) (Sungyoung Lee) (Taewoong Jeon)

**요 약** CORBA 보안 서비스는 네트워크를 기반으로 하는 분산 환경 하에서 데이타 전송 시 사용자가 요구하는 수준의 비밀성 보장과 무결성 제공을 위해 비보호, 무결성, 비밀성, 무결성 및 비밀성과 같은 파라미터를 갖는 QoP 기능을 지원하고 있다. 그러나 기존의 QoP 기능은 전자상거래, 재무, 통신, CORBA Med와 같이 광범위한 CORBA의 응용 영역들간에 특정 암호화 알고리즘에 대한 서로 다른 정책을 가질 경우 전송되는 데이타에 대한 무결성과 비밀성을 지원할 수 없는 문제점을 갖고 있다. 이를 해결하기 위하여 본 논문에서는 광범위한 CORBA의 응용 영역들간에 데이타 전송 시 암호화 알고리즘 관리 및 제어 기능을 개선한 QoP 모델을 제안한다. 개선된 QoP 관리 및 제어 모델은 OMG에서 발표한 RFP의 요구사항을 기반으로 설계되었으며, 특정 암호화 알고리즘들의 관리와 실행 중에 사용되는 암호화 알고리즘의 변경 제어와 암/복호화시 비도 설정을 지원한다. 구현 결과 개선된 QoP 관리 및 제어 모델은 CORBA IDL로 설계하여 개발자로 하여금 응용 개발의 용이성을 부여하였으며, 광범위한 CORBA응용 영역들간에 암호화 알고리즘에 대한 QoP 기능을 지원한다. 또한, ISO/IEC 8824(ASN.1)에서 정의된 OID를 사용하여 암호화 알고리즘의 호환성을 제공한다.

***Abstract*** CORBA (Common Object Request Broker Architecture) security service must supports the QoP (Quality of Protection) to guarantee confidentiality and integrity that are requested by the users when the system transmits the data in distributed environments. The conventional QoP functions, however cannot support integrity and confidentiality of the transmission data in CORBA applications such as electronic commerce, electronic finance, tele-communications, and CORBAMed if they have their own different encryption policies. To resolve these problems, we propose an enhanced QoP scheme in CORBA that improves the management and control functions to handle the existing encryption algorithms more efficiently than ever before. The proposed scheme is created by adopting the OMG's RFP (Request for Proposal). It has the modification and control functions for the existing encryption algorithms in the course of their management and implementation. It also can support the encryption and decryption confidentiality. The experimental result shows that the proposed scheme makes the developers to build the applications more easily since it uses CORBA ID. It also supports the QoP function for the encryption algorithms in CORBA applications. Furthermore, the invented QoP scheme provides an interoperability among the encryption algorithms since it uses OID (Object Identifier) defined by the ISO/IEC 8824(AS7N.1).

## 1. 서 론

일반적인 시스템에서의 보안은 어떤 정보에 대해 인가되지 않은 접근을 시도하거나 운영을 방해하는 것으로부터 시스템 보호를 목표로 한다. 이것은 비밀성(confidentiality)과 무결성(integrity)이라는 두가지 기본적인 특성을 갖는다. 비밀성이란 오직 인가된 사용자에게 정보의 접근을 허용하는 것을 말하며, 무결성이란 오직 정당한 권한을 가진 사람에게 인가된 방법만으로 정보 변경을 허용함을 일컫는다[1]. 이같은 특성을 만족시

† 비 회 원 : 네플(주) 연구원
hjlee@netple.com
†† 종신회원 : 경희대학교 전자계산공학과 교수
sylee@oslab.kyunghee.ac.kr
††† 종신회원 : 고려대학교 전산학과 교수
jeon@tiger.korea.ac.kr
논문접수 : 2000년 1월 20일
심사완료 : 2000년 8월 29일

키기 위해서는 정보의 암호화는 필수적이다.

QoP(Quality of Protection)란 보안 기능들의 집합이며 인증(authentication), 비밀성, 무결성, 부인거부(non-repudiation) 등의 조합들로 구성되어 있다[2]. QoP는 네트워크 환경에서 데이타 전송 시 사용자가 요구하는 수준의 무결성 및 비밀성을 지원하는데 이용된다. 기존의 시스템에서 QoP는 비보호(no-protection), 무결성, 비밀성, 무결성 및 비밀성(integrity and confidentiality)과 같은 유형들로 나누어지며, 시스템 개발자는 전송할 메시지에 대한 안전성을 이러한 유형에 따라 설정한다. 각각의 유형에 대한 구체적인 메시지 보호와 관련되어서 어떤 암호화 알고리즘을 사용할 것인가? 키는 어떻게 관리할 것인가? 에 대한 정책은 개발자가 전담하여 설정한다.

하지만, 광범위한 CORBA 응용 영역에서 기존의 QoP 유형만으로는 실제 시스템에 사용되는 암호화 알고리즘에 대한 정보를 알 수 없다. 또한, 기존의 QoP 기능은 전자상거래, 재무, 통신, CORBAMed와 같이 광범위한 CORBA 응용 영역들 사이에서 특정 암호화 알고리즘들이 서로 다른 정책을 가질 경우, 전송되는 데이타에 대한 무결성과 비밀성을 지원할 수 없는 문제점을 갖고 있다. 예를 들면, 암호화 알고리즘에 대한 정책이 다른 경우 메시지 암호화 시 서로 다르게 암/복호화 과정을 수행하면 메시지에 대해 적절한 보안 적용이 불가능하다. 따라서, 클라이언트와 타겟간의 암호화 알고리즘 정보에 대한 관리와 제어를 할 수 있고, 동일한 암호화 알고리즘 정책을 설정할 수 있는 인터페이스가 요구된다. 현재 OMG(Object Management Group)에서는 이러한 요구들을 수용하여 QoP 관리 및 제어 기능을 개선하기 위한 RFP(Request for Proposal)를 발표하였으며[4,5,11,12] 현재 IONA사와 Inprise사가 참여하여 연구 중에 있다.

본 논문에서는 비보호, 비밀성, 무결성, 비밀성/무결성과 같은 단순한 QoP 파라미터를 지원하는 기존의 QoP 모델에 광범위한 CORBA 응용영역에서 요구하는 암호화 알고리즘과 호환이 가능하도록 새로운 QoP 모델을 제안한다. 제안된 QoP 모델은 기존의 QoP 기능에 광범위한 CORBA의 응용영역간에 전송되는 데이타에 대한 다양한 무결성과 비밀성을 지원하기 위해 보안 문맥 객체에 개선된 QoP 정보가 추가되었다. 그리고, 추가된 QoP 정보를 관리할 수 있는 인터페이스와 실행 중에 사용되는 암호화 알고리즘의 변경 및 암호화 강도를 제어할 수 있는 기능도 개선되었다. 또한, 추가된 QoP 정보가 클라이언트와 타겟간에 동일하게 유지하도록 하기 위해서 QoP 정보 협정 모듈과 공통 QoP 정보 생성 모듈들도 첨부되었다. 한편, 기존의 QoP 모델이 단순히 암호화를 할 것인가? 또는, 암호화의 용도가 무엇인가? 에 대해 초점을 두고 있지만, 제안된 모델에서는 CORBA에서 요구한 RFP에 바탕을 두고 광범위한 CORBA의 응용 영역들간의 서로 다른 보안 정책을 어떻게 통합하고 유지시킬 수 있느냐? 에 초점을 맞추고 있다.

구현 결과 클라이언트와 타겟에 개선된 QoP 관리 인터페이스를 이용하여 동일한 QoP 정보의 생성이 가능하였고, 암호화 알고리즘 변경 또는 QoP 레벨 변경 시에도 메시지의 암/복호화가 가능하였다. CORBA 응용 시스템 구현 시 개발자로 하여금 요구한 QoP 기능을 쉽게 구현할 수 있도록 제안한 모델을 CORBA IDL로 설계하였으며, CORBA 응용 영역간에 서로 다른 암호화 알고리즘 정책 설정 시 암호화 알고리즘을 명확히 구별하기 위해 ISO/IEC 8824 (ASN.1)에서 정의된 OID(Object Identifier)를 사용하여 암호화 알고리즘의 호환성을 가능케 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고, 3장에서는 본 논문에서 제안된 암호화 알고리즘의 관리 기능과 제어 기능이 개선된 QoP 모델을 설명한다. 4장에서는 모델 구현을 통한 기능 실험 결과에 대해 살펴본 후, 5장에서 결론을 내린다.

## 2. 관련 연구

본 장에서는 기존의 QoP 관련연구, 그리고 본 논문에 참조하고 있는 OMG의 RFP 요구사항에 대해서 살펴본다.

### 2.1 기존의 QoP 관련연구

GSS-API(Generic Secure Service-API)는[2,8,9] 암호를 잘 아는 사용자는 물론 그렇지 못한 사용자들에게도 보안 서비스를 제공하는 보안 인터페이스 규격으로써 하부 메커니즘 (underlying mechanism)이나 프로토콜에 독립적으로 동작한다. 전형적인 GSS-API 호출자(caller)는 통신 프로토콜이며, 이는 인증, 무결성, 그리고 선택적인 비밀성 보안 서비스들을 제공받아 자신의 통신을 보호하기 위해서 GSS-API와 인터페이싱 한다. GSS-API 사용자는 [그림 1]과 같이 자신의 로컬 GSS-API 구현에서 제공되는 토큰을 원격 시스템상의 등위에게 전달하고, 상대 등위는 수신된 토큰을 자신의 지역 GSS-API에게 전달한다.

신임장 (credential)은 다른 프로세스로부터 자신의 신분을 확인하기 위한 전역식별자 (global identity)이다. 신임장 역시 로컬 GSS-API 내의 데이타 구조를

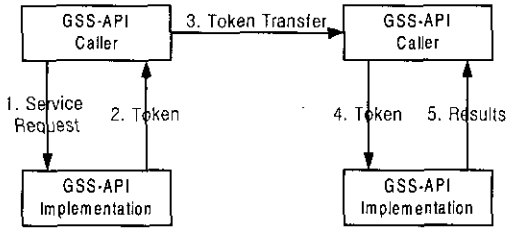


그림 1 GSS-API의 개략적인 데이타 전달 과정

갖는다. 보안 문맥은 향후 보안 서비스를 위해 요구되는 공유 상태 정보를 담고 있는 데이타 구조이다. 시스템간의 보안 서비스를 제공받기 위해서는 반드시 신임장 설정이 선행되어야 한다. 보안 문맥이 설정되는 동안 두 시스템에서는 서로를 인증한다. GSS-API는 하부 보안 메커니즘의 사용을 적극 권장하고 있다. GSS-API는 신임장을 설정하는 단계, GSS_Init_sec_context()와 GSS_Accept_ sec_context()를 이용하여 시스템들간에 보안 문맥을 설정하는 단계, GSS_Sign()과 GSS_Verify()를 이용하여 데이타 발신처 인증 및 데이타 무결성 서비스를 제공하는 단계, 그리고 GSS_Seal()과 GSS_Unseal()을 이용해서 메시지를 암/복호화 하는 단계들로 구성된다. GSS-API는 분산 네트워크 환경에서 일관된 보안 서비스를 지원하기 위한 API이며, CORBA는 GSS-API를 참조, 수용하고 있기 때문에 그 구조와 기능이 유사하다. CORBA에서 초기에 제안된 QoP와 GSS-API의 QoP는 동일하지만 GSS-API는 CORBA처럼 광범위한 분야에서 좀더 다양한 보안 특성을 지원하기 위한 구조와 기능은 가지고 있지 않다.

QoP 인터페이스를 지원하는 메커니즘으로서 IPSEC은 네트워크 계층의 보안 프로토콜은 인증, 무결성, 접근 제어등의 암호화를 지원한다 [6,7]. [그림 2]는 IPSEC의 개념적 모델이며, 이때 사용되는 키 관리 알고리즘은 공개키 암호화 기술을 근간으로 하고 있다. 설정된 IP 인증 헤더와 IP 캡슐화 보안 Payload의 형태는

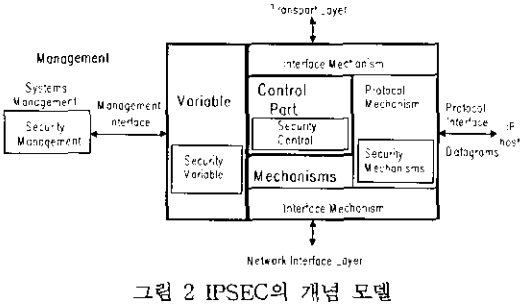


그림 2 IPSEC의 개념 모델

암호화 알고리즘에 무관하게 구성된다. IPSEC은 IPv4와 IPv6에서 지원되는데, 이 둘은 각각 따로 제공될 수도 있지만 서로 혼합되어 제공될 수도 있다. IPSEC에서의 QoP는 각각 비밀성과 무결성을 지원하는 암호화 알고리즘들의 집합을 정의하여 양단간의 정책 협정 시 두 시스템에서 사용할 암호화 알고리즘을 설정한다. CORBA의 RFP는 응용 레벨에서의 개선된 QoP를 지원을 위한 요구사항이지만, IPSEC은 네트워크 레벨에서의 QoP를 위한 요구사항이다. 따라서, 그 구조는 유사하지만 용도 차이로 인한 기능의 차이가 존재한다. 다시 말하면, CORBA는 서로 다른 플랫폼 또는 언어로 만들어진 응용들간에 전송되어지는 정보들에 대한 QoP 기능을 지원하지만 IPSEC은 각 응용마다의 특성을 고려하지 않는다.

커버로스[10]는 MIT에서 Athena 프로젝트를 수행하면서 개발된 인증 방법으로 QoP 인터페이스를 지원한다. 기존의 인증 방법은 사용자가 시스템에 로그인하여 패스워드를 입력하고, 입력된 패스워드가 맞으면 사용자에 대한 인증을 완료하는 형태이었다. 그러나, 클라이언트-서버 형태에서는 사용자의 패스워드가 네트워크를 경유하게 되므로 침입자들에 의하여 패스워드가 도용될 요소가 있다. 그래서 커버로스에서는 [그림 3]과 같이 티켓 승인 서버를 두어 티켓 하나를 할당받고, 이 티켓으로 서비스를 제공받고자 하는 서버에게 인증을 받게 하였다.

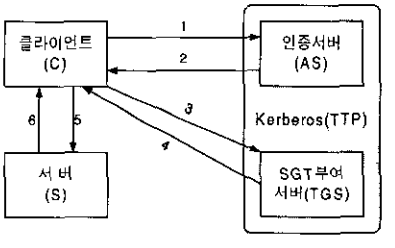


그림 3 커버로스의 인증 절차

커버로스는 인증 시스템의 일종으로, 현재 존재하는 인증 시스템(SESAME 등)들이 표준으로 삼고 있다. 사용자와 서버간의 인증을 위한 제 3자, 즉 신뢰된 인증 서버를 이용한다. 중요한 점은 클라이언트와 서버간에 이용할 암호화 알고리즘의 선택과 키 분배를 어떻게 할 것 인가이다. 커버로스는 이러한 보안 인터페이스를 GSS-API 구조를 이용하고 있기 때문에 QoP에 대한 구조와 기능이 GSS-API와 유사하다.

### 2.2 CORBA 보안 서비스

CORBA 보안 서비스는 OMG의 OMA(Object Man-

agement Architecture)내에 보안 기능을 제공할 수 있는 공통 객체 서비스이며, 정보의 무결성과 비밀성을 지원하기 위해서 데이타 암호화 기능과 QoP 기능을 기술하고 있다[3].

- OMG의 QoP

OMG의 QoP 정보 설정은 [그림 4]와 같으며, Object 클래스의 set_deafult_qop() 인터페이스를 통해 보안 문맥 객체 내에 있는 QoP 정보를 설정하며, 보안 연계 설정 과정을 통하여 QoP 정보가 클라이언트와 타겟에 동일하게 형성된다. QoP를 적용한 암/복호화는 보안 문맥 객체에서 제공하는 protected_message()와 reclaimed_mess-age() 인터페이스를 통하여 메시지 암/복호화가 수행된다.

[그림 5]는 QoP 기능을 이용한 암호화 오퍼레이션을 나타내며, 암호화할 메시지(라인 2)와 사용자 요구에 따라 보안 문맥에 설정된 QoP 레벨(라인 3)에 맞는 암호화 알고리즘으로 메시지를 암호화한 후, 암호화된 메시지(라인 4)와 토큰(라인 5)을 전달한다.

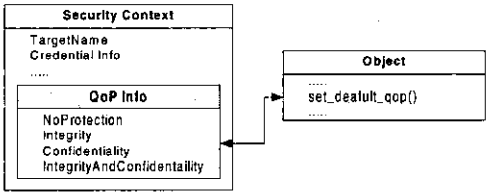


그림 33 QoP 정보 설정

```
1 : void protected_message(
2 :     in Security::Opaque message
3 :     in Security::QoP qop,
4 :     out Security::Opaque text_buffer,
5 :     out Security::Opaque in_token,
    )
```

그림 5 QoP 기능을 이용한 암호화 오퍼레이션

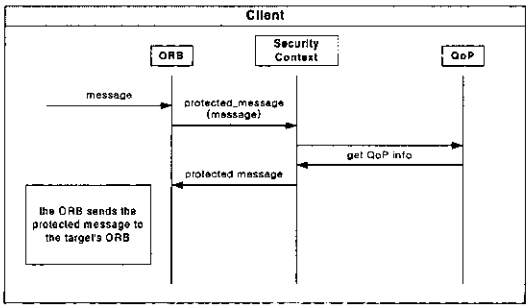


그림 6 QoP를 적용한 메시지 암호화

[그림 6]은 QoP를 적용한 메시지 암호화 과정을 보여준다. 메시지 암호화는 암호화할 메시지와 사용자 요구에 따라 보안 문맥에 설정된 QoP 정보에 맞는 암호화 알고리즘으로 메시지를 암호화한 후, 암호화된 메시지와 토큰을 타겟에 전달한다.

```
1 : void reclaimed_message(
2 :     in Security::Opaque text_buffer
3 :     in Security::Opaque in_token,
4 :     out Security::QoP qop,
5 :     out Security::Opaque message,
    )
```

그림 7 QoP기능을 이용한 복호화 오퍼레이션

[그림 7]은 QoP를 이용한 메시지 복호화 오퍼레이션을 나타내며, 암호화된 메시지(라인 2)와 토큰(라인 3)을 입력받아 토큰에서 암호화된 메시지에 대한 정보를 얻고 암호화된 메시지를 복호화한 후, 보안 문맥에 설정된 QoP 레벨(라인4)과 복호화된 메시지(라인 5)를 전달한다.

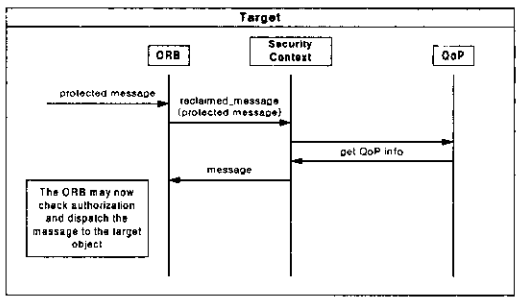


그림 8 QoP를 적용한 메시지 복호화

[그림 8]과 같이 QoP를 적용한 메시지 복호화는 암호화된 메시지와 토큰을 입력받아 토큰에서 암호화된 메시지에 대한 정보를 얻고 설정된 QoP 정보를 참조하여 암호화된 메시지를 복호화한 후, ORB에 복호화된 메시지를 전달한다.

기존의 QoP 모델을 가지는 CORBA의 응용에서 사용자가 요구하는 특정 암호화 알고리즘과 비도를 제어하기 위해서는 특정 암호화 알고리즘 정보를 교환할 수 있는 기능과 암호화 알고리즘에 대한 추가/삭제 기능, 암호화/복호화에 사용되는 암호화 알고리즘을 실행 중 변경 기능, 사용자가 요구한 비도 변경 기능이 필요하

다.

- QoP 개선 요구사항

전술 한바와 같이, 기존의 QoP 유형은 비보호, 무결성, 비밀성, 무결성 및 비밀성과 같은 유형들로 나누어지며, 시스템 개발자는 전송할 메시지에 대한 안전성을 이러한 유형에 따라 설정한다. 각각의 유형에 대한 구체적인 메시지 보호와 관련되어서 어떤 암호화 알고리즘을 사용할 것인가? 키는 어떻게 관리할 것인가? 에 대한 정책은 개발자가 전담하여 설정한다.

그러나, 전자상거래, 재무, 통신, CORBAMed와 같은 광범위한 CORBA의 응용 영역들은 정보가 전송되는 동안 더 높은 비밀성과 데이타의 무결성을 보장할 수 있도록 CORBA 보안 서비스의 개선을 요구하고 있다[3]. 이러한 응용분야에서 CORBA 트랜잭션들의 안전한 전송을 위해 처리되어지는 일련의 과정 (통신의 초기설정, 실행 및 완료)은 상대방에 대한 정보의 신뢰와 암호화 알고리즘의 사용 정도이다. 하지만, 정보를 보다 안전하게 전송하기 위해서는 특정 암호화 알고리즘들을 관리할 수 있는 기능과 정보의 특성 또는 목적에 따라 암호화 강도를 다르게 제어할 수 있는 기능이 필요하다. 예를 들면, 암호화 알고리즘에 대한 정책이 다른 경우 메시지 암호화 시 서로 다르게 암/복호화 과정을 수행하여서는 정상적으로 시스템이 동작을 할 수가 없으며, 메시지에 대해 적절한 보안 적용이 불가능하다. 이를 위해서는 클라이언트와 타겟간에 암호화 알고리즘 정보의 관리와 제어를 할 수 있고, 동일한 암호화 알고리즘 정책을 설정할 수 있는 인터페이스가 요구된다.

따라서, 광범위한 CORBA 응용에서 보다 안전한 전송을 지원하기 위하여 기존의 QoP 모델에 암호화 알고리즘의 관리 기능과 제어기능을 개선 또는 강화하려는 연구가 진행중이다. 이러한 개선 요구들을 수용하여 OMG에서는 QoP 개선 요구사항들을 기술한 RFP를 발표하였고, 이에 대한 내용은 다음과 같다.

암호화 알고리즘의 관리 기능은 첫째, CORBA 응용하에서 가용한 QoP 암호화 알고리즘 관련 정보 지원이다. 이는 가용한 암호화 알고리즘이 서로 다른 응용간에 각각의 암호화 알고리즘에 대한 정보를 주고받기 위한 기능으로써 보안 협정을 할 때 사용된다. 둘째, QoP 레벨에서 가용한 기본/교체 암호화 알고리즘 설정 지원이다. 이는 초기에 보안 협정 없이 처음 사용되는 암호화 알고리즘을 설정하고 또한 QoP 정책에 따라 교체되어 지는 암호화 알고리즘을 사전에 설정하기 위한 기능이다. 셋째, QoP 레벨을 위해 가용한 기본/교체 암호화 알고리즘 집합에 대한 추가 및 삭제 지원이다. 이는 기본 암호화 알고리즘들의 집합과 교체 암호화 알고리즘들의 집합을 관리함으로써 보다 다양한 암호화 알고리즘을 지원할 수 있는 기능이다. 두 번째와 세 번째 기능은 사용자가 시스템을 운영할 때 QoP 정책설정에 사용된다.

한편, 암호화 알고리즘의 제어기능으로는 첫째, 실행중에 가용한 특정 QoP 레벨을 변경할 수 있는 기능이다. 이는 전송되는 데이타의 유형에 따라 서로 다른 비도를 설정하여 각각의 데이타에 대한 알맞은 비밀성을 부여할 수 있도록 한다. 둘째, 실행 중에 기본 암호화 알고리즘을 교체 암호화 알고리즘으로 변경할 수 있어야 한다. 이는 전송되는 데이타의 특성(텍스트, 이미지)에 따라 서로 다른 암호화 알고리즘을 설정하여 전송 데이타에 가장 적합한 비밀성을 유지할 수 있도록 한다.

## 3. 개선된 QoP 관리 및 제어 모델

### 3.1 시스템 모델

본 논문에서 제안하는 QoP 관리 및 제어 모델은 기존의 QoP 모델을 개선하였기 때문에 기존의 QoP 모델과 유사한 구조를 가진다[그림 9]. QoP 정보 연계 블록은 보안 서비스 모듈과 Message Interceptor 모듈로 구성된다.

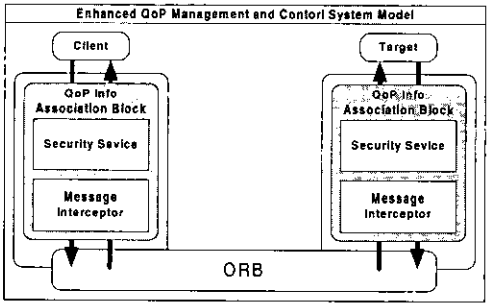


그림 9 개선된 QoP 관리 및 제어 모델을 위한 시스템 모델

제안된 모델은 암호화 알고리즘과 암호화 알고리즘 집합에 대한 정보의 관리(설정, 획득, 추가, 삭제) 기능과 암/복호화에 사용되는 알고리즘의 제어(암호화 알고리즘 변경) 기능이 개선된다. 그리고, 공통 QoP 정보 생성 모듈에 의해 생성된 QoP 정보를 기반으로 Message Interceptor 모듈을 통해 전송되는 모든 메시지에 대한 암/복호화를 수행한다.

### 3.2 개선된 QoP 관리 및 제어 모델

개선된 QoP 관리 및 제어 모델의 구조와 모듈간 연관

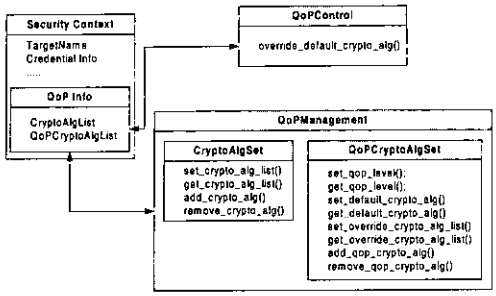


그림 10 개선된 QoP 관리 및 제어 모델의 구조

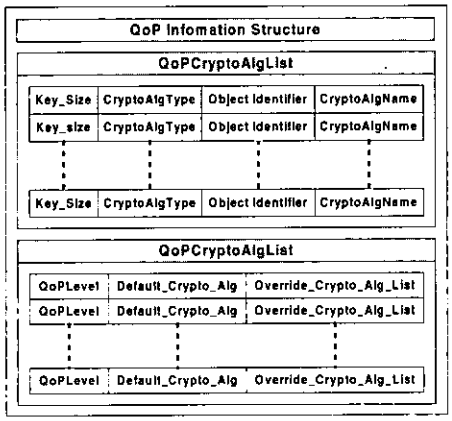


그림 11 QoP 정보 구조

관계는 [그림 10]에 나타나 있는데, 이는 보안 문맥 객체에 있는 QoP 정보를 관리하기 위한 QoP 관리 모듈과 qop_level의 변경과 암호화 모드의 변경을 지원하는 QoP Control 모듈로 구성된다. QoP 관리 모듈은 시스템에서 지원하는 암호화 알고리즘의 리스트를 관리하기 위한 CryptoAlgSet 인터페이스와 QoP 암호화 알고리즘의 리스트를 관리하기 위한 QoPCryptoAlgSet으로 구성된다.

QoP 정보는 암호화 레벨과 암호화 알고리즘 리스트, QoP 암호화 알고리즘 리스트로 구성되어 있으며 [그림 11]은 그 구조를 나타낸다. 암호화 레벨은 메시지 암호화 시 초기 암호화 비도를 나타내며, 메시지 암/복호화 시 암호화 레벨을 요청 파라미터로 전달받아 암/복호화를 수행한다. 암호화 알고리즘 리스트는 암호화 알고리즘 클래스들의 집합으로 구성되어 있으며, 클라이언트 시스템 또는 타겟 시스템에 설치된 암호화 알고리즘 정보를 지원한다. 암호화 알고리즘 클래스는 암호화에 사용되는 키 크기, 암호화 알고리즘의 유형(Integrity, Confidentiality), 암호화 알고리즘의 호환성을 부여하기 위한 ISO/IEC 8824(ASN.1)에 정의된 OID와 암호화 알고리즘 이름으로 구성된다. QoP 암호화 알고리즘 리스트는 QoP 암호화 알고리즘 클래스들의 집합으로 구성되어 있으며, 클라이언트와 타겟 간에 전송되는 메시지에 대한 QoP 기능을 지원하는 암호화 알고리즘 정보를 지원한다. QoP 암호화 알고리즘 클래스는 암호화 비도를 선택적으로 지원하기 위한 QoPLevel, 초기 메시지 암호화에 사용될 기본 암호화 알고리즘, 메시지의 특성에 따라 기본 암호화 알고리즘을 교체할 수 있는 교체 암호화 알고리즘 리스트로 구성된다

QoP 관리 모듈은 시스템에서 지원하는 암호화 알고리즘 리스트와 QoP 암호화 알고리즘 리스트의 관리를 위한 인터페이스들의 집합이며, 그 구조는 [그림 12]와 같다. QoP 관리 모듈은 보안 문맥 객체 내에 있는 QoP 정보의 암호화 알고리즘 리스트와 QoP 암호화 알고리즘 리스트를 관리하기 위해 각각 CryptoAlgSet 인터페이스와 QoPCryptoAlgSet 인터페이스를 제공한다

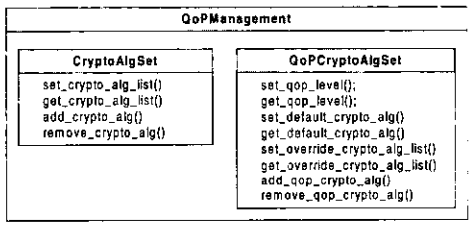


그림 12 QoP 관리 모델의 구조

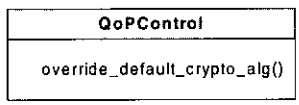


그림 13 QoP 제어 모듈

CrytpoAlgSet 인터페이스는 알고리즘 리스트의 설정을 위한 set_crypto_alg_list(), 정보 획득을 위한 get_rypto_alg_list()와 생성된 리스트에 알고리즘 추가/삭제를 위한 add_crypto_alg() /remove_crypto_alg() 인터페이스로 구성된다. QoPCryptoAlgSet 인터페이스는 암호화 알고리즘의 비도 정보인 QoPLevel 관리를 위한 set_qop_level()/get_qop_level() 인터페이스, 기본 암호화 알고리즘의 관리를 위한 set_default_crypto_alg()/ et_default_crypto_alg() 인터페이스, 교체 암호화 알고리즘의 관리를 위한 set_override_crypto_alg_list()/ get_verride_crypto_alg_list() 인터페이스로 구성된다

QoP 제어 모듈은 [그림 13]과 같이 기본 암호화 알고

리즘을 교체 암호화 알고리즘으로 변경하기 위한 override_default_crypto_alg() 인터페이스로 구성된다. QoP 제어 모듈은 보안 문맥 객체에 접근하여 요구한 변경 정보를 설정한다.

기본 암호화 알고리즘 변경 과정은 QoP 암호화 알고리즘 리스트의 교체 암호화 알고리즘 리스트에서 변경 가능 여부를 판단한 후 요구한 교체 암호화 알고리즘을 기본 암호화 알고리즘으로 변경한다. 제어 기능은 응용 프로그램이 실행 중에 암호화 알고리즘 특성에 따라 다른 암호화 알고리즘의 적용을 가능하도록 한다. 제어 기능은 응용 개발자에 의해서 추가도 가능하다. 이 경우에는 설계된 제어 인터페이스를 상속받아 새로운 제어 인터페이스를 추가해야 한다.

### 3.3 QoP 정보 협정

개선된 QoP 정보 협정 과정은 QoP 정보를 공유하기 위해 반드시 요구되며, 기존의 보안 문맥 설정 과정과 유사하다[그림 14]. 기존의 보안 문맥 설정 과정에 다음과 같은 과정이 추가된다. 타겟은 클라이언트로부터 전송 받은 QoP 정보를 자신의 QoP 정보와 공통 QoP 정보 생성 모듈을 이용하여 새로운 공통 QoP 정보를 생성하고, 자신의 보안 문맥 객체에 설정한다. 클라이언트에게 공통 QoP 정보를 전송한다. 클라이언트는 타겟에 의해 생성된 공통 QoP 정보를 받아, 자신의 보안 문맥 객체에 설정한다.

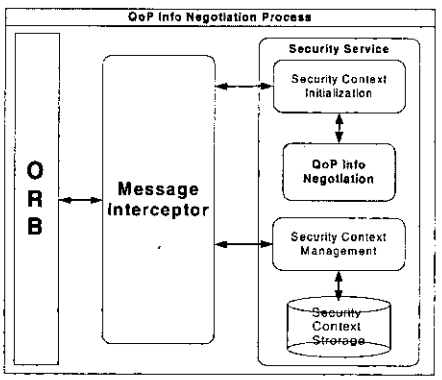


그림 14 QoP 정보 협정 과정

공통 QoP 정보 생성 모듈은 두 개의 QoP 정보 즉, 자신의 QoP 정보와 전송 받은 QoP 정보에서 새로운 공통 QoP 정보를 생성한다. 이 모듈은 타겟에서 수행이 되며, 공통 QoP 레벨 생성, 공통 암호화 알고리즘 리스트 생성, QoP 암호화 알고리즘 리스트 생성으로 나누어진다. [그림 15]는 공통 QoP 정보 생성 모듈을 나타낸다.

공통 QoP 레벨 생성시에는 클라이언트와 타겟에서 지원하는 QoP 암호화 알고리즘의 초기 QoP 레벨을 선택한다. 공통 암호화 알고리즘 리스트 생성은 클라이언트와 타겟에서 공통으로 지원하는 암호화 알고리즘의 리스트 생성시 이루어진다. 공통 QoP 암호화 알고리즘 리스트 생성은 QoP 레벨별로 정렬된 리스트에서 공통 기본 암호화 알고리즘을 먼저 선택하고, 공통 교체 암호화 알고리즘 리스트를 생성한다.

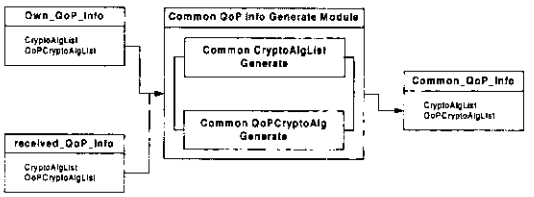


그림 15 공통 QoP 정보 생성 모듈

QoP 정보 협정 과정을 살펴보기 위하여 [그림 16]과 같은 예를 고려하자. Qi는 QoP 레벨을 의미하며, 대문자 A~G까지는 암호화 알고리즘을 간략히 표기 한 것이다. 클라이언트의 암호화 알고리즘은 A, B, C, D, E이며, 타겟의 암호화 알고리즘은 B, C, D, E, F, G이다.

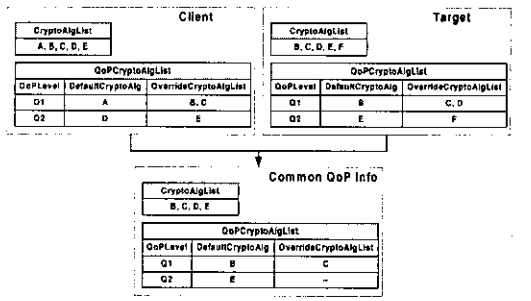


그림 16 QoP 정보 협정 과정 예

클라이언트와 타겟의 QoP 암호화 알고리즘 정책은 먼저, 클라이언트와 타겟 시스템에서 공통으로 지원할 수 있는 암호화 알고리즘 리스트를 설정한다. 공통 QoP 암호화 알고리즘 리스트를 설정하기 위해 클라이언트와 타겟에서 지원하는 QoP 레벨별로 분류한다. 분류된 클라이언트와 타겟의 QoP 암호화 알고리즘 리스트를 QoP 레벨 단위로 [그림 17]과 같은 순서로 기본 암호화 알고리즘과 교체 암호화 알고리즘 리스트를 설정한다.

Step1: 클라이언트의 기본 암호화 알고리즘과 타겟의 기본 암호화 알고리즘을 비교하여 동일할 경우 공통 기본 암호화 알고리즘으로 설정한다.

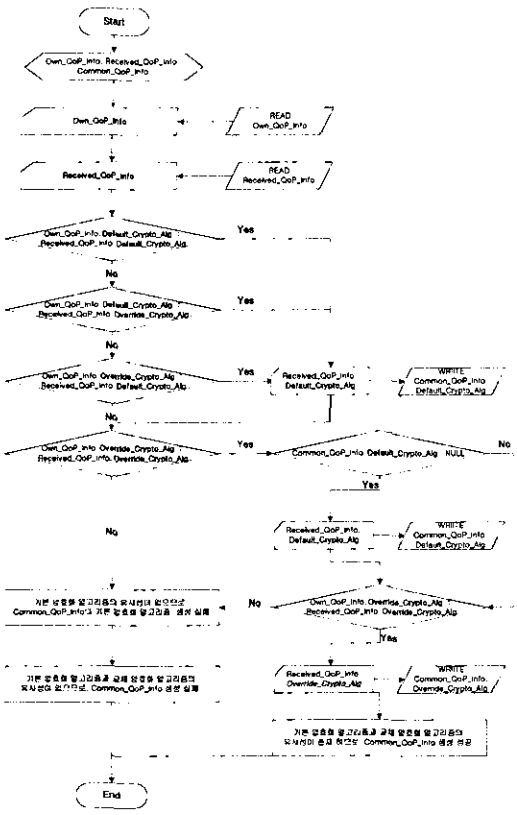


그림 17 QoP 정보협정모듈 알고리즘

Step2: Step 1에서 기본 암호화 알고리즘의 설정을 하지 못했을 경우, 클라이언트의 기본 암호화 알고리즘과 타겟의 교체 암호화 알고리즘 리스트를 비교하여 동일한 경우 공통 기본 암호화 알고리즘으로 설정한다.

Step 3: Step 2에서 실패한 경우, 타겟의 기본 암호화 알고리즘과 클라이언트의 교체 암호화 알고리즘 리스트를 비교하여 같을 경우 공통 기본 암호화 알고리즘으로 설정한다.

Step 4: Step 3에서 실패한 경우, 클라이언트의 교체 암호화 알고리즘 리스트와 타겟의 교체 암호화 알고리즘 리스트를 각각 비교하여 동일한 암호화 알고리즘을 기본 암호화 알고리즘으로 한다.

Step 5: Step 4에서 실패한 경우, 현재의 QoP 레벨은 암/복호화 기능을 지원할 수 없으므로, 현재 QoP 레벨이 삭제된다.

Step 6: Step 1과 Step 2, Step 3, Step 4에서 기본 암호화 알고리즘을 설정한 경우, 클라이언트의 교체 암호화 알고리즘 리스트와 타겟의 교체 암호화 알고리즘 리스트를 비교하여 동일한 암호화 알고리즘들을 공통 교체 암호화 알고리즘 리스트로 설정한다.

Step 7: Step 1부터 Step 6을 모두 수행하여 공통 QoP 정보를 생성할 수 없는 경우, 공통 QoP 정보 협정 과정 결과를 실패로 처리한다.

### 3.4 개선된 QoP를 적용한 암/복호화

개선된 QoP를 적용한 암/복호화는 기존 CORBA 메시지 보호 서비스의 암/복호화 과정과 유사하다. 단지, 보안 문맥 객체의 관리 인터페이스인 protected_message()과 reclaimed_message()와 파라미터에 QoP 레벨이 추가된다. 클라이언트의 암호화 과정은 [그림 18]과 같으며, QoP_Level 파라미터가 추가된 protected_message() 인터페이스를 통하여 수행된다. protected_message() 인터페이스의 QoP_Level 값은 보안 문맥 객체에 설정된 QoP 암호화 알고리즘 리스트에서 기본 암호화 알고리즘 검색을 위한 인덱스 값이다. 메시지 암호화 시 사용자가 요구한 QoP_Level 값을 이용하여 검색된 기본 암호화 알고리즘 정보를 기반으로 외부 보안 서비스(암호화 알고리즘 패키지)를 사용하여 전달된 메시지를 암호화한다. 암호화된 메시지는 Message Interceptor 모듈을 지나 ORB의 통신 모듈을 통해 타겟에 전송된다. 타겟의 암호화 과정은 클라이언트의 암호화 과정과 동일하다.

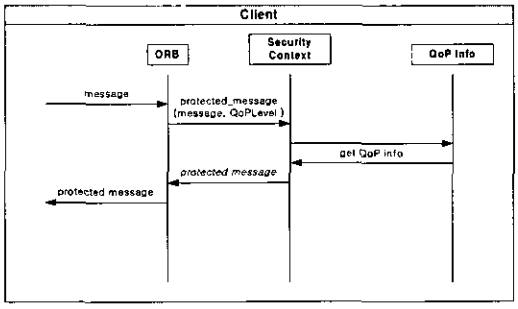


그림 18 개선된 QoP를 적용한 암호화

타겟의 복호화 과정은 [그림 19]와 같으며, 암호화 과정과 마찬가지로 QoP 레벨 파라미터가 추가된 reclaimed_message()를 통하여 수행된다. ORB의 통신 모듈을 통해 Message Interceptor 모듈을 거쳐 암호화된 메시지는 미리 설정된 QoP_Level 값을 이용하여 QoP 암호화 알고리즘 리스트에서 복호화에 필요한 기본 암호화 알고리즘 정보를 획득한다. 획득된 QoP 암호화 알고리즘 정보를 기반으로 암호화 알고리즘 패키지를 선택하고 암호화된 메시지를 복호화 한다. 클라이언트의 복호화 과정은 타겟의 복호화 과정과 동일하다.

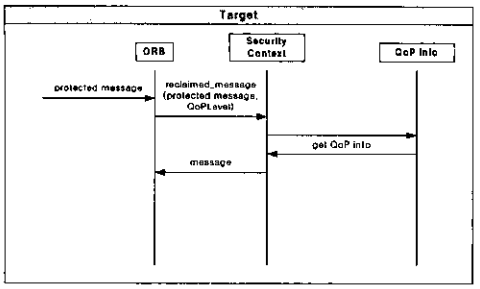


그림 19 개선된 QoP를 적용한 복호화

## 4. 구현 및 시험

QoP 관리 및 제어 모델의 기능 시험을 수행하기 위한 환경은 [그림 20]과 같다. 먼저, 물리적으로 기본적인 분산 시스템 환경을 구성하기 위해 LAN 환경에서 네트워크로 연결된 두 대의 WinNT4.0 워크스테이션을 기반으로 하며, 시험 환경의 논리적 구성은 WinNT #1 시스템이 응용 클라이언트 그리고 WinNT #2 시스템은 타겟(서버)이 위치하도록 하였다. 또한 응용 클라이언트와 타겟 시스템은 각기 기본적인 ORB로써 IONA사의 OrbixWeb 31c를 이용하였다.

기능 시험을 위한 응용 프로그램의 구성은 타겟 프로그램과 클라이언트 프로그램으로 이루어진다. 클라이언트 응용 프로그램은 보안 문맥 초기화 과정을 거친 후에, 입력된 메시지를 암호화하여 타겟에 전송하고 타겟은 이를 받아 복호화 하여 원본 메시지와 암호화 메시지를 화면에 조합하여 출력하도록 작성하였다. 또한, QoP 제어 기능 시험은 클라이언트에서 기본 암호화 알고리즘 변경을 통하여 수행된다. 입력된 메시지를 암호화하여 전송하고 타겟은 이를 복호화하여 원본 메시지를 화면에 출력한 후, 클라이언트에서 전송한 원본 메시지와 타겟에서 복호화된 메시지를 비교하였다.

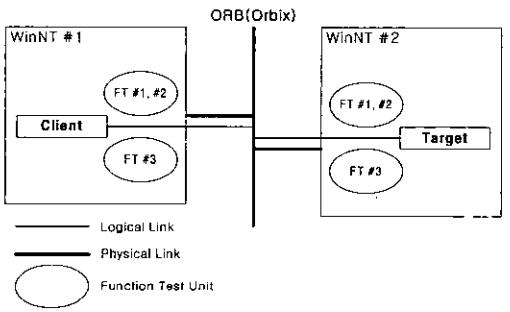


그림 20 기능 시험 모델

모의시험은 [그림 20]에서 FT(Function Test) 개수만큼 시행하였으며, 각기 시험의 수행 내용 및 그 결과는 [표 1]과 같다.

표 1 기능 시험 수행 항목

| 분류 | 시험 대상 기능 | 시험 방법 | 시험 결과 |
|---|---|---|---|
| FT #1 | • QoP 정보 협정 기능 | • 클라이언트와 서버 각각에 생성된 공통 QoP 정보 객체의 확인 | • QoP 정보 협정 완료 |
| FT #2 | • 암/복호화 기능 | • 클라이언트에서 메시지를 송부하기 전에 보낼 메시지의 내용과 타겟에서 암호화한 메시지를 출력 장치에 출력한 메시지와 비교 | • 출력된 문자열 판독 불가 |
| FT #3 | • QoP 제어 기능: 기본 암호화 알고리즘 변경 | • 클라이언트 응용 프로그램에서 기본 암호화 알고리즘 변경을 수행한 후 메시지를 암호화하여 타겟에 전송, 타겟에서 복호화한 메시지와 클라이어트 메시지와 비교 | • 메시지 복호화 가능 |

QoP 정보 협정 기능의 결과는 [그림 21]과 같다. 클라이언트의 QoP 정보를 받은 타겟이 공통 QoP 정보 생성 모듈을 수행하여 공통 QoP 정보를 정확하게 생성하였다. 메시지 암/복호화 기능의 결과는 [그림 22]와 같다. 클라이언트에서 QoP 레벨 2로 암호화하여 전송한 메시지에서 원본 메시지의 판독은 불가능하다. QoP 제어 기능의 결과는 [그림 23]과 같다. 클라이언트에서 암호화 알고리즘 변경을 수행 후 암호화된 메시지와 타겟에서 복호화된 메시지가 일치하였다. 이것은 클라이언트에서 암호화 알고리즘을 변경 수행 시 클라이언트의 QoP 정보를 변경하고 타겟에 변경 사항을 전달하여 동일한 QoP 정보가 설정되었기 때문이다.

시험 결과 보안 문맥 설정 기능(QoP 정보 협정), 암호화 기능, QoP 제어 기능(기본 암호화 알고리즘 변경)이 올바르게 동작하였다. 모니터링 프로그램을 수행하여 보안 문맥 객체 메시지와 시스템들 간의 메시지를 조사, 분석한 결과 클라이언트와 타겟에 설정된 보안 문맥 설정 기능과 암호화 기능이 정상적으로 수행됨을 알 수 있었다

## 5. 결 론

광범위한 CORBA 응용 영역에서는 기존의 QoP 유형만으로는 실제 시스템에 사용되는 암호화 알고리즘에 대한 정보를 알 수 없기 때문에 QoP의 개선을 요구하고 있으며, 기존의 CORBA 보안 서비스에서 QoP 기능 지원 시 암호화 알고리즘 정책의 차이로 인하여 네트워

크를 통해 전송되는 데이타의 무결성과 비밀성을 보장할 수 없었다. 이를 해결하기 위해서 암호화 알고리즘 정책을 공유할 수 있도록 암호화 알고리즘에 대한 호환성을 부여해야 하며, 암호화 알고리즘 정보의 관리와 제어를 할 수 있는 인터페이스가 요구된다.

본 논문에서는 CORBA 환경에서 QoP 기능 지원 시 암호화 알고리즘에 대한 호환성을 부여할 수 있는 개선된 QoP 관리 및 제어 모델을 제안하였다. 제안된 QoP 모델은 OMG에서 발표한 RFP의 요구사항에 맞추어 설계되었다. 또한, 광범위한 CORBA의 응용영역간에 전송되는 데이타에 대한 다양한 무결성과 비밀성을 지원하기 위해 보안 문맥 객체에 QoP 정보를 추가하고, 추가된 QoP 정보를 관리할 수 있는 인터페이스와 실행 중에 사용되는 암호화 알고리즘의 변경 및 암호화 강도를 제어할 수 있는 기능을 개선하였으며, 추가된 QoP 정보가 클라이언트와 타겟간에 동일하게 유지하도록 하기 위해서 QoP 정보 협정 모듈과 공통 QoP 정보 생성 모듈을 첨가하였다.

```
/* Client QoP Info */

CryptoAlgList :
        [1] : 40, 1, "DES40", "DES"
        [2] : 64, 1, "DES64", "DES"
        [3] : 64, 1, "RC4", "RC4"
        [4] : 64, 1, "RC5", "RC5"
        [5] : 128, 1, "IDEA", "IDEA"
QoPCryptoAlgList :
        [1] : 1, CryptoAlgList[1],
        [2] : 2, CryptoAlgList[2], CryptoAlgList[3];CryptoAlgList[4];
        [3] : 3, CryptoAlgList[5];
```

```
/* Server QoP Info */

CryptoAlgList:
        [1] : 64, 1, "DES64", "DES"
        [2] : 64, 1, "RC4", "RC4"
        [3] : 64, 1, "RC5", "RC5"
        [4] : 128, 1, "IDEA", "IDEA"
        [5] : 128, 1, "DESede", "3DES"
QoPCryptoAlgList :
        [1] : 2, CryptoAlgList[1], CryptoAlgList[2];CryptoAlgList[3];
        [2] : 3, CryptoAlgList[4], CryptoAlgList[5];
```

```
/* Common QoP Info */

CryptoAlgList:
        [1] : 64, 1, "DES64", "DES"
        [2] : 64, 1, "RC4", "RC4"
        [3] : 64, 1, "RC5", "RC5"
        [4] : 128, 1, "IDEA", "IDEA"
QoPCryptoAlgList :
        [1] : 2, CryptoAlgList[1], CryptoAlgList[2];CryptoAlgList[3];
        [2] : 3, CryptoAlgList[4], null;
```

그림 21 QoP 정보 협정 기능 시험 결과

```
/* Client Encrypt Execution */

Used QoPLevel : 2
Used QoPCryptoAlg :
        [1] : 64, 1, "DES64", "DES"

Plain Text :
        사용자로 부터 입력 받은 메시지는 암호화 과정을 수행후 타겟으로
        전송되며 복호화 과정을 거쳐 사용자가 보낸 메세지를 확인한다.

Encrypt Text :
        ????}? ¥c??•??l????QM?RR?j-@$???`??m#¯¯??'
        \?W$aG?0?c\•?{?!?????J5m`?L?xc[??a?b]?\???5
        ?n[????? ??[??F? ?m\3£?Y??•G?D??^
```

```
/* Target Encrypt Execution */

Used QoPLevel : 2
Used QoPCryptoAlg :
        [1] : 64, 1, "DES64", "DES"

Encrypt Text :
        ????}? ¥c??•??l????QM?RR?j-@$???`??m#¯¯??'
        \?W$aG?0?c\•?{?!?????J5m`?L?xc[??a?b]?\???5
        ?n[????? ??[??F? ?m\3£?Y??•G?D??^

Decrypt Text :
        사용자로 부터 입력 받은 메시지는 암호화 과정을 수행후 타겟으로
        전송되며 복호화 과정을 거쳐 사용자가 보낸 메세지를 확인한다.
```

그림 22 암/복호화 기능 시험 결과

```
/* Client Encrypt Execution */

Used QoPLevel : 2
Used QoPCryptoAlg :
        [1] : 64, 1, "RC4", "RC4"

Plain Text :
        클라이언트에서 기본 암호화 알고리즘 변경을 수행한 후 타겟에 변
        경 정보를 전송한다. 타겟은 전송받은 변경 정보를 이용하여 요구한
        기본 암호화 알고리즘으로 QoP정보를 재설정한다.
Encrypt Text :
        {C@175b15a6??K6-
        ???3?P???????Jg??zP? ¯•J•??????l?KQ??l???????>2? l?~?•
        ?4???o$??•=?/?` UbE/?R ]??b'?z'V7???`o:J??ezcwp £
        9?¯???Q]???•C???¥Z?• ¸!A$`e ?l??•??<?¸Z
```

```
/* Target Encrypt Execution */

Used QoPLevel : 2
Used QoPCryptoAlg :
        [1] : 64, 1, "RC4", "RC4"

Encrypt Text :
        {C@175b15a6??K6-
        ???3?P???????Jg??zP? ¯•J•??????l?KQ??l???????>2? l?~?•
        ?4???o$??•=?/?` UbE/?R ]??b'?z'V7???`o:J??ezcwp £
        9?¯???Q]???•C???¥Z?• ¸!A$`e ?l??•??<?¸Z

Decrypt Text :
        클라이언트에서 기본 암호화 알고리즘 변경을 수행한 후 타겟에 변
        경 정보를 전송한다. 타겟은 전송받은 변경 정보를 이용하여 요구한
        기본 암호화 알고리즘으로 QoP정보를 재설정한다.
```

그림 23 기본 암호화 알고리즘 변경 기능시험 결과

구현 결과 클라이언트와 타겟에 개선된 QoP 관리 인터페이스를 적용하여 동일한 QoP 정보의 생성이 가능케 되었으며 암호화 알고리즘 변경 또는 QoP 레벨 변경 시에도 메시지의 암/복호화가 가능하게 되었다. 또한, 안전한 CORBA 응용 시스템 구현 시 개발자로 하여금

요구된 QoP 기능을 쉽게 구현할 수 있도록 제안된 모델을 CORBA IDL로 설계하였으며, CORBA 응용 영역간에 서로 다른 암호화 알고리즘 정책 설정 시 ISO/IEC 8824 (ASN.1)에서 정의된 OID를 사용하여 암호화 알고리즘의 호환성을 가능케 하였다.

향후 연구로는 제안된 QoP 모델이 OMG의 RFP의 요구사항에 따라 기존의 QoP 모델에서 지원할 수 없는 암호화 알고리즘에 대한 호환성을 지원하였으나, 현재 공개되어 있지 않은 다른 QoP 모델과의 성능 비교가 필요하다. 또한, 제안된 QoP 관리 및 제어 모델을 다른 보안 메커니즘에도 적용시키기 위한 연구도 필요하다.

## 참 고 문 헌


[1] OMG Security Working Group, OMG White Paper on Security, Issue: 1.0, 1994. 4, http://www.omg.org/docs/orbos /98-11-23

[2] J. Linn, Generic Security Service Application Program Interface (GSS-API), RFC-1508, 1993. 9

[3] OMG(Object Management Group), Quality of Protection Management and Control, Request for Proposal, 1998. 11, http://www.omg.org/docs/orbos /98-11-23

[4] OMG(Object Management Group), CORBA Security Service, Revision 1.2, 1998. 1, http://www.omg.org/docs/ptc /98-01-02

[5] OMG(Object Management Group), CORBA Messaging, Joint Revised Submission, 1998. 5, http://www.omg.org /docs/orbos/98-05-08

[6] C. Adams, Independent Data Unit Protection Generic Security Service Application Program Interface (IDUP-GSS-API), IETF RFC-2479, 1998. 11

[7] C. Adams, The Simple Public-Key GSS-API Mechanism (SPKM), IETF RFC-2025, 1996. 10

[8] J. Linn, Generic Security Service Application Program Interface, Version 2 (GSS-API 2), RFC-2078, 1997. 1

[9] E. Baize, The Simple and Protected GSS-API Negotiation Mechanism, RFC-2487, 1998. 12

[10] J. Linn, The Kerberos Version 5 GSS-API Mechanism, RFC-1964, 1996. 6

[11] J. Raisbeck, A. Hatwalne, CORBA Security and its Application to the VRE Project, 1998. 5, http://www.u.arizona.edu /~hatwalne/ece678/hw/ 678prj.htm

[12] D, Chizmadia, A Quick Tour of the CORBA Security Service, OMG Security SIG, 1998. 9



이 희 종

1998년 한신대학교 정보통신학과 학사. 2000년 경희대학교 전자계산공학과 석사. 2000년 3월 ~ 현재 네플(주) 연구원. 관심분야는 실시간 시스템, Security, CORBA.



이 승 룡

1978년 고려대학교 재료공학과 학사. 1987년 12월 Illinois Institute of Technology 전산학 석사. 1991년 12월 Illinois Institute of Technology 전산학 박사. 1992년 ~ 1993년 Governors State University 조교수. 1993년 ~ 현재 경희대학교 전자정보학부 (전자계산공학 전공) 부교수. 관심분야는 실시간 컴퓨팅, 멀티미디어 시스템



전 태 웅

1981년 서울대학교 계산통계학과 학사. 1983년 서울대학교 계산통계학과 석사. 1992년 Illinois Institute of Technology 전산과학 박사. 1983년 ~ 1987년 금성통신 연구소 주임연구원. 1992년 ~ 1995년 LG산전 연구소 책임연구원. 1995년 ~ 현재 고려대학교 자연과학부(전산학 전공) 부교수. 관심분야는 소프트웨어 테스트링, 소프트웨어 아키텍처, 객체지향 프레임워크, 실시간 소프트웨어 공학