□특집□

CORBA환경에서 개선된 Quality of Protection 모델

이 희 종† 이 승 룡††

◈ 목 차 ◈

1. 서 론

CORBA(Common Object Request Broker Architecture) 보안 서비스는 OMG(Object Management Group)의 OMA(Object Management Architecture)내에 보안 기능을 제공할 수 있는 공통 객체 서비스이며, 분산 환경 하에서 데이터 전송 시 사용자가 요구하는 수준의 비밀성 보장과 무결성 지원을 위해 비보호(no-protection), 무결성(integrity), 비밀성(confidentiality), 무결성 및 비밀성(integrity and confidentiality)등과 같은 파라미터를 갖는 QoP(Quality of Protection) 기능을 지원한다[1,3,4,5,11,12]. QoP란 보안 기능들의 집합이며 인증(authenti- cation), 비밀성, 무결성, 부인거부(non-repudiation) 등의 조합들로 구성되어 있다[2,6,7,8,9,10]. 이러한 QoP는 분산 환경에서 데이터 전송 시 사용자가 요구하는 수준의 무결성 및 비밀성을 지원할 수 있다.

그러나, 전자상거래, 재무, 통신, CORBAMed와 같은 광범위한 CORBA의 응용 영역들은 정보가 전송되는 동안 더 높은 비밀성과 데이터의 무결성을 보장할 수 있도록 CORBA 보안 서비스의 개선을 요구하고 있다[3]. 이러한 영역들에서 CORBA 트랜잭션들의 안전한 전송을 위해 처리되어지는 일련의 과정 (예를 들어 통신의 초기설정, 실행 및 완료)은 상대방에 대한 정보의 신뢰와 암호화 알고리즘의 사용 정도이다. 하지만, 정보를 보다 안전하게 전송하기 위해서는 특정 암호화 알고리즘들을 관리 할 수 있는 기능과 정보의 특성 또는 목적에 따라 암호화 강도를 다르게 제어할 수 있는 기능이 필요하다. 현재 OMG에서는 이러한 요구들을 수용하여 QoP 관리 및 제어 기능을 개선하기 위한 RFP(Request For Proposal)를 발표하였으며[3] 현재 IONA사와 Inprise사가 참여하여 연구 중에 있다.

본 고에서는 OMG의 RFP 요구사항을 기반으로 개선된 관리 및 제어 QoP 모델을 제시한다. 이는 무보호, 비밀성, 무결성, 비밀성/무결성과 같은 단순한 QoP 파라미터를 지원하는 기존의 QoP 모델에 광범위한 CORBA 응용영역에서 요구하는 암호화 알고리즘이 호환이 가능하도록 QoP 정보를 보안 문맥 객체에 추가되었다. 그리고, 추가된 QoP 정보를 관리하고 제어할 수 있는 인터페이스, 클라이언트와 타겟 간에 서로 다른 QoP 정책을 가질 경우 보안 문맥 연계 시 타겟에서 공통 QoP 정보를 생성하여 클라이언트와 동일한 QoP

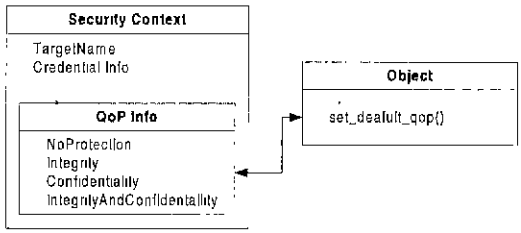
† 준회원 : 경희대학교 전자정보학부 전자계산공학전공
†† 정회원 : 경희대학교 전자정보학부 전자계산공과 부교수

정보를 갖도록 하는 기능도 추가된다. 개선된 QoP 관리 및 제어 모델은 특정 암호화 알고리즘들의 관리와 실행 중에 사용되는 암호화 알고리즘의 변경 제어와 암/복호화 시 비도 설정을 지원한다.

본 고의 구성은 다음과 같다. 2장에서는 기존의 CORBA QoP 모델에 대하여 살펴보고, 3장에서는 OMG에서 제안하고 있는 QoP 관리 및 제어 기능을 개선하기 위한 RFP를 간단히 소개한다. 4장에서는 제안한 암호화 알고리즘의 관리 기능과 제어 기능이 개선된 QoP 모델을 기술한 뒤, 5장에서 결론을 내린다.

2. CORBA에서의 QoP

CORBA 환경에서 QoP 유형은 무보호(noprotection), 무결성(integrity), 비밀성(confidentiality), 무결성 및 비밀성(integrity-and-confidentiality)과 같은 유형들로 나누어지며, 시스템 개발자는 전송할 메시지에 대한 안전성을 이러한 유형에 따라 설정한다. 각각의 유형에 대한 구체적으로 메시지 보호와 관련되어서 "어떤 암호화 알고리즘을 사용할 것인가?" "키는 어떻게 관리할 것인가?"에 대한 정책은 개발자가 전담하여 설정한다.



(그림 1) QoP 정보 설정

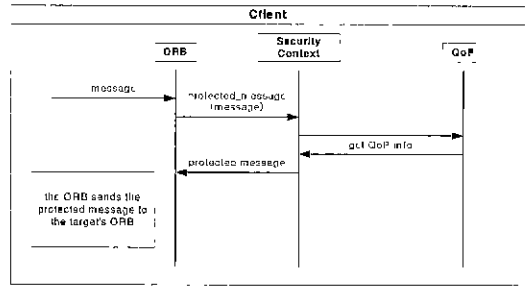
QoP 정보 설정은 (그림 1)과 같다. 여기서, Object 클래스의 set_deafult_qop() 인터페이스를 통해 보안 문맥 객체 내에 있는 QoP 정보를 설정하며, 보안 연계 설정 과정을 통하여 QoP 정보가 클라이언트와 타겟에 동일하게 형성된다.

QoP를 적용한 암/복호화는 보안 문맥 객체에서 제공하는 protected_message()와 reclaimed_message() 인터페이스를 통하여 메시지 암/복호화가 수행된다. QoP 기능을 이용한 암호화 오퍼레이션은 (그림 2)에서 보여주는데, 이는 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.

```
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 2) QoP 기능을 이용한 암호화 오퍼레이션

QoP를 적용한 메시지 암호화 과정은 (그림 3)에서 보여준다. 메시지 암호화는 암호화할 메시지와 사용자 요구에 따라 보안 문맥에 설정된 QoP 정보에 맞는 암호화 알고리즘으로 메시지를 암호화한 후, 암호화된 메시지와 토큰을 타겟에 전달한다.



(그림 3) QoP를 적용한 메시지 암호화

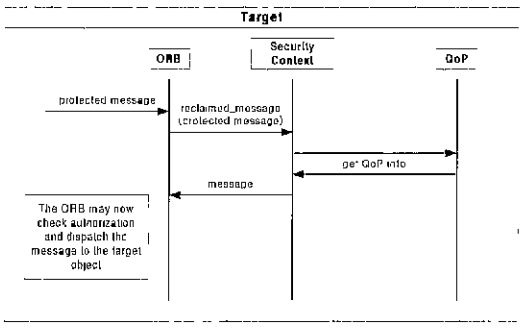
QoP를 이용한 메시지 복호화 오퍼레이션은 (그

림 4)에서 보여주는데, 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.

```
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 4) QoP 기능을 이용한 복호화 오퍼레이션

QoP를 적용한 메시지 복호화는 (그림 5)와 같이 암호화된 메시지와 토큰을 입력받아 토큰에서 암호화된 메시지에 대한 정보를 얻고 설정된 QoP 정보를 참조하여 암호화된 메시지를 복호화한 후, ORB에 복호화된 메시지를 전달한다. 기존의 QoP 모델에서 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 필요하다.



(그림 5) QoP를 적용한 메시지 복호화

3. QoP 개선 요구사항

광범위한 CORBA 응용 영역에서는 기존의 QoP 유형만으로는 실제 시스템에 사용되는 암호화 알고리즘에 대한 정보를 알 수 없기 때문에 현재의 QoP의 개선을 요구하고 있다. 예를 들면, 암호화 알고리즘에 대한 정책이 다른 경우 메시지 암호화 시 서로 다르게 암/복호화 과정을 수행하여 정상적으로 동작을 할 수가 없으며, 메시지에 대해 적절한 보안성의 적용이 불가능하다. 이를 위해서는 클라이언트와 타겟 간에 암호화 알고리즘 정보의 관리와 제어를 할 수 있고, 동일한 암호화 알고리즘 정책을 설정할 수 있는 인터페이스가 요구된다. 이러한 개선 요구들을 수용하여 OMG에서는 QoP 개선 요구사항들을 기술한 RFP를 발표하였고, 이에 대한 개괄적인 내용은 다음과 같다.

암호화 알고리즘의 관리 기능은, 첫째 CORBA 응용 하에서 가용한 QoP 암호화 알고리즘 관련 정보 지원 기능, 둘째 QoP 레벨에서 가용한 기본/교체 암호화 알고리즘 설정 지원 기능과, 셋째 QoP 레벨을 위해 가용한 기본/교체 암호화 알고리즘 집합에 대한 추가 및 삭제 지원 기능을 지원해야 한다. 첫 번째 기능은 가용한 암호화 알고리즘이 서로 다른 응용간에 각각의 암호화 알고리즘에 대한 정보를 주고받기 위한 것으로써, 보안 협정을 할 때 사용된다. 두 번째 기능은 초기에 보안 협정 없이 처음 사용되는 암호화 알고리즘을 설정하고 또한 QoP 정책에 따라 교체되어지는 암호화 알고리즘을 사전에 설정하기 위한 것이다. 세 번째 기능은 기본 암호화 알고리즘들의 집합과 교체 암호화 알고리즘들의 집합을 관리함으로써 보다 다양한 암호화 알고리즘을 지원하기 위한 것이다. 두 번째와 세 번째 기능은 사용자가 시스템을 운영할 때 QoP 정책설정에 사용된다.

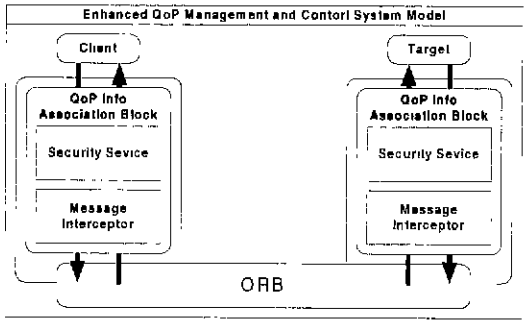
암호화 알고리즘의 제어기능은, 첫째 실행 중

에 가용한 특정 QoP 레벨을 변경할 수 있는 기능, 둘째 실행 중에 기본 암호화 알고리즘을 교체 암호화 알고리즘으로 변경할 수 있어야 한다. 셋째 제어기능은 전송되는 데이터의 유형에 따라 서로 다른 비도를 설정하여 각각의 데이터에 대한 알맞은 비밀성을 부여할 수 있도록 한다. 넷째는 전송되는 데이터의 특성(텍스트, 이미지)에 따라 서로 다른 암호화 알고리즘을 설정하여 전송 데이터에 가장 적합한 비밀성을 유지할 수 있도록 한다.

4. 개선된 QoP 관리 및 제어 모델

4.1 시스템 모델

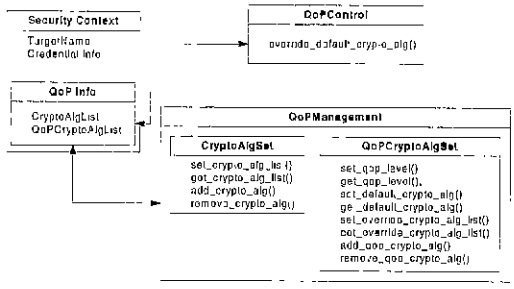
본 고에서 제안하는 개선된 QoP 관리 및 제어 모델은 기존의 QoP 모델과 유사한 구조를 가진다(그림 6). QoP 정보 연계 블록은 보안 서비스 모듈과 Message Interceptor 모듈로 구성된다. 암호화 알고리즘과 암호화 알고리즘 집합에 대한 정보의 관리(설정, 획득, 추가, 삭제) 기능과 암/복호화에 사용되는 알고리즘의 제어(암호화 알고리즘 변경) 기능이 개선된다. 공통 QoP 정보 생성 모듈에 의해 생성된 QoP 정보를 기반으로 Message Interceptor 모듈을 통해 전송되는 모든 메시지에 대한 암/복호화를 수행한다.



(그림 6) 개선된 QoP 관리 및 제어 모델을 위한 시스템 모델

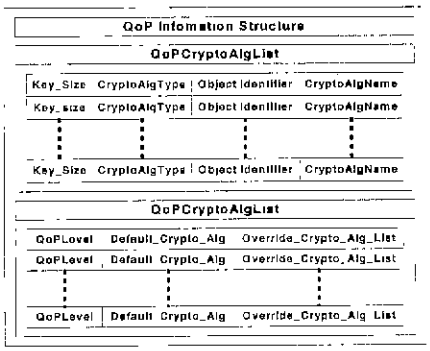
4.2 개선된 QoP 관리 및 제어 모델

개선된 QoP 관리 및 제어 모델의 구조와 그 연관관계는 (그림 7)에 나타나 있다. 보안 문맥 객체에 있는 QoP 정보를 관리하기 위한 QoP Management 모듈과 qop_level의 변경과 암호화 모드의 변경을 지원하는 QoP Control 모듈로 구성된다. QoP Management 모듈은 시스템에서 지원하는 암호화 알고리즘의 리스트를 관리하기 위한 CryptoAlgSet 인터페이스와 QoP 암호화 알고리즘의 리스트를 관리하기 위한 QoPCryptoAlgSet으로 구성된다.



(그림 7) 개선된 QoP 관리 및 제어 모델의 구조

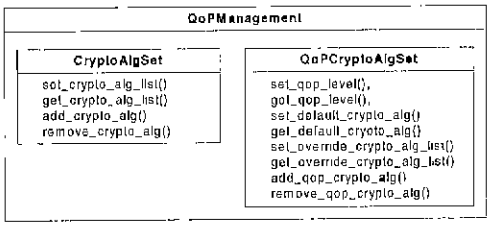
QoP 정보는 암호화 레벨과 암호화 알고리즘 리스트, QoP 암호화 알고리즘 리스트로 구성되어 있으며 (그림 8)은 그 구조를 나타낸다. 암호화 레벨은 메시지 암호화 시 초기 암호화 비도를 나타내며, 메시지 암/복호화 시 암호화 레벨을 요청 파라미터로 전달받아 암/복호화를 수행한다. 암호화 알고리즘 리스트는 암호화 알고리즘 클래스들의 집합으로 구성되어 있으며, 클라이언트 시스템 또는 타겟 시스템에 설치된 암호화 알고리즘 정보를 지원한다. 암호화 알고리즘 클래스는 암호화에 사용되는 키 크기, 암호화 알고리즘의 유형(Integrity, Confidentiality), 암호화 알고리즘의 호환성을 부여하기 위한 ISO/IEC 8824(ASN.1)에 정의된 OID(Object Identifier)와 암호화 알고리즘 이름으로 구성된다.



(그림 8) QoP 정보 구조

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 9)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.

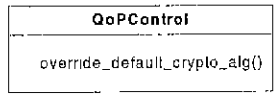


(그림 9) QoP 관리 모델의 구조

CrytpoAlgSet 인터페이스는 알고리즘 리스트의 설정을 위한 set_crypto_alg_list(), 정보 획득을 위한 get_crypto_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()/get_default_crypto_alg() 인터페이스, 교체 암호화 알고리즘의 관리를 위한 set_override_crypto_alg_list()/get_override_crypto_alg_list() 인터페이스로 구성된다.

QoP 제어 모듈은 (그림 10)과 같이 기본 암호화 알고리즘을 교체 암호화 알고리즘으로 변경하기 위한 override_default_crypto_alg() 인터페이스로 구성된다. QoP 제어 모듈은 보안 문맥 객체에 접근하여 요구한 변경 정보를 설정한다.



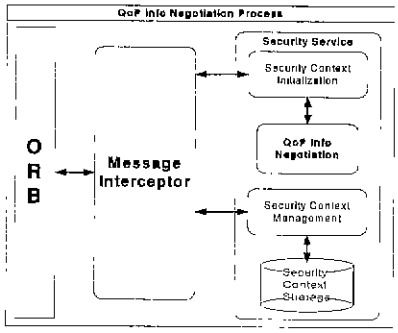
(그림 10) QoP 제어 모듈

기본 암호화 알고리즘 변경 과정은 QoP 암호화 알고리즘 리스트의 교체 암호화 알고리즘 리스트에서 교체 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 적용이 가능하도록 한다. 제어 기능은 응용 개발자에 의해서 추가도 가능하다. 단, 본 연구에서 설계된 제어 인터페이스를 상속받아 새로운 제어 인터페이스를 추가해야 한다.

4.3 개선된 QoP 정보 협정

QoP 정보 협정 과정은 QoP 정보를 공유하기

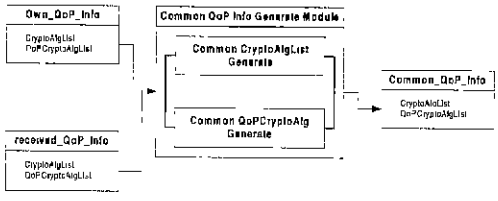
위해 반드시 요구되며, 기존의 보안 문맥 설정 과정과 유사하다(그림 11).



(그림 11) QoP 정보 협정 과정

기존의 보안 문맥 설정 과정에 다음과 같은 과정이 추가된다. (1) 타겟은 클라이언트로부터 전송 받은 QoP 정보를 자신의 QoP 정보와 공통 QoP 정보 생성 모듈을 이용하여 새로운 공통 QoP 정보를 생성하고 (2) 자신의 보안 문맥 객체에 설정한다. (3) 클라이언트에게 공통 QoP 정보를 전송한다. (4) 클라이언트는 타겟에 의해 생성된 공통 QoP 정보를 받아 (4) 자신의 보안 문맥 객체에 설정한다.

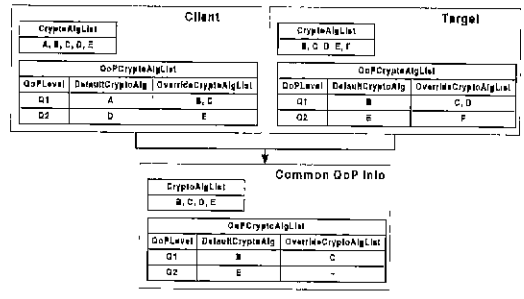
공통 QoP 정보 생성 모듈은 두 개의 QoP 정보 즉, 자신의 QoP 정보와 전송 받은 QoP 정보에서 새로운 공통 QoP 정보를 생성한다. 이 모듈은 타겟에서 수행이 되며, 공통 QoP 레벨 생성, 공통 암호화 알고리즘 리스트 생성, QoP 암호화 알고리즘 리스트 생성으로 나누어진다. (그림 12)는 공통 QoP 정보 생성 모듈을 나타낸다.



(그림 12) 공통 QoP 정보 생성 모듈

공통 QoP 레벨 생성은 클라이언트와 타겟에서 지원하는 QoP 암호화 알고리즘의 초기 QoP 레벨을 선택한다. 공통 암호화 알고리즘 리스트 생성은 클라이언트와 타겟에서 공통으로 지원하는 암호화 알고리즘의 리스트를 생성한다. 암호화 알고리즘 클래스의 OID로 알고리즘들로 분류하고 OID가 동일한 암호화 알고리즘들을 생성한다. 공통 QoP 암호화 알고리즘 리스트 생성은 QoP 레벨별로 정렬된 리스트에서 공통 기본 암호화 알고리즘을 먼저 선택하고, 공통 교체 암호화 알고리즘 리스트를 생성한다.

QoP 정보 협정 과정을 살펴보기 위하여 (그림 13)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. 클라이언트와 타겟의 QoP 암호화 알고리즘 정책은 (그림 13)과 같다.



(그림 13) QoP 정보 협정 과정 예

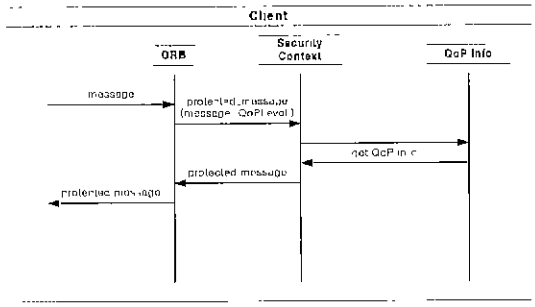
먼저, 클라이언트와 타겟 시스템에서 공통을 지원할 수 있는 암호화 알고리즘 리스트를 설정한다. 공통 QoP 암호화 알고리즘 리스트를 설정하기 위해 클라이언트와 타겟에서 지원하는 QoP 레벨별로 분류한다. 분류된 클라이언트와 타겟의 QoP 암호화 알고리즘 리스트를 QoP 레벨 단위로 다음과 같은 순서로 기본 암호화 알고리즘과 교체 암호화 알고리즘 리스트를 설정한다. Step1)

클라이언트의 기본 암호화 알고리즘과 타겟의 기본 암호화 알고리즘을 비교하여 동일할 경우 공통 기본 암호화 알고리즘으로 설정한다. Step2) 1)의 과정에서 기본 암호화 알고리즘의 설정을 하지 못했을 경우, 클라이언트의 기본 암호화 알고리즘과 타겟의 교체 암호화 알고리즘 리스트를 비교하여 동일한 암호화 알고리즘을 검색하여 동일한 경우 공통 기본 암호화 알고리즘으로 설정한다.

Step 3) 2)의 과정에 실패할 경우, 타겟의 기본 암호화 알고리즘과 클라이언트의 교체 암호화 알고리즘 리스트를 비교하여 같을 경우 공통 기본 암호화 알고리즘으로 설정한다. Step 4) 3)에서 실패할 경우, 클라이언트의 교체 암호화 알고리즘 리스트와 타겟의 교체 암호화 알고리즘 리스트를 각각 비교하여 동일한 암호화 알고리즘을 기본 암호화 알고리즘으로 한다. Step 5) 4)에서 실패할 경우, 현재의 QoP 레벨은 암/복호화 기능을 지원할 수 없으므로, 현재 QoP 레벨이 삭제된다. Step 6) 1)과 2), 3), 4)의 과정에서 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) 1)과 2), 3), 4), 5), 6)과정 모두 수행하여 공통 QoP 정보를 생성할 수 없는 경우, 공통 QoP 정보 협정 과정 결과를 실패로 처리한다.

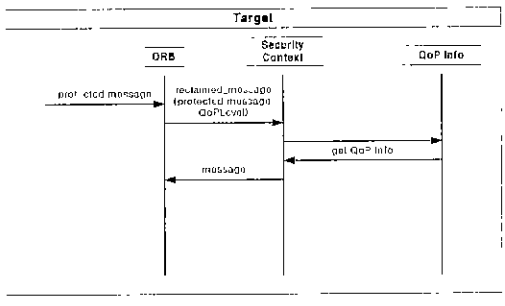
4.4 개선된 QoP를 적용한 암/복호화

기존 CORBA의 메시지 보호 서비스의 암/복호화 과정과 유사하다. 단지, 보안 문맥 객체의 관리 인터페이스인 protected_message()과 reclaimed_message()의 파라미터에 QoP 레벨이 추가된다. 클라이언트의 암호화 과정은 (그림 14)와 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 알고리즘을 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 이용하여 전달된 메시지를 암호화한다. 암호화된 메시지는 Message Interceptor 모듈을 지나 ORB의 통신 모듈을 통해 타겟에 전송된다. 타겟의 암호화 과정은 클라이언트의 암호화 과정과 동일하다.



(그림 14) 개선된 QoP를 적용한 암호화

타겟의 복호화 과정은 (그림 15)와 같으며, 암호화 과정과 마찬가지로 QoP 레벨을 파라미터가 추가된 reclaimed_message()를 통하여 수행된다. ORB의 통신 모듈을 통해 Message Interceptor 모듈을 지나온 암호화된 메시지는 미리 설정된 QoP_Level 값을 이용하여 QoP 암호화 알고리즘



(그림 15) 개선된 QoP를 적용한 복호화

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.

5. 결 론

광범위한 CORBA 응용 영역에서는 기존의 QoP 유형만으로는 실제 시스템에 사용되는 암호화 알고리즘에 대한 정보를 알 수 없기 때문에 현재의 QoP의 개선을 요구하고 있으며, 기존의 CORBA 보안 서비스에서 QoP 기능 지원 시 암호화 알고리즘 정책의 차이로 인하여 네트워크를 통해 전송되는 데이터의 무결성과 비밀성을 보장할 수 없다. 이를 해결하기 위해서는 암호화 알고리즘 정책을 공유할 수 있도록 암호화 알고리즘에 대한 호환성을 부여해야 하며, 암호화 알고리즘 정보의 관리와 제어를 할 수 있는 인터페이스가 요구된다.

본 고에서는 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. 기존의 QoP 기능에 광범위한 CORBA의 응용영역간에 전송되는 데이터에 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 추가하였다. 또한, 안전한 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.

참고문헌

[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 정보통신 공학과(이학사)
2000년 경희대학교 전자계산공학(석사)
2000년 현재 (주) 네플 근무
관심분야 : 실시간 시스템, 보안, 멀티미디어 시스템, CORBA, Java

이 승 룡

1978년 고려대학교 재료공학과 (공학사)
1986년 Illinois Institute of Technology 전산과학(석사)
1991년 Illinois Institute of Technology 전산과학(박사)
1992년-1993년 Governors State University, Illinois 조교수
1993년-현재 경희대학교 전자계산공학과 부교수
관심분야 : 실시간 시스템, 실시간 고장허용시스템, 멀티미디어 시스템, 실시간 CORBA, Java