보안 서비스를 고려한 이동 에이전트 모델과 클라이언트-서버 모델의 성능 비교

(A Performance Comparison of the Mobile Agent Model with the Client-Server Model under Security Conditions)

한 승 완 † 정 기 문 †† 박 승 배 ††† 임 형 석 ††††

(Seung-Wan Han) (Ki-Moon Jeong) (Seung-Bae Park) (Hyeong-Seok Lim)

요 약 분산 컴퓨팅 환경에서 프로세스 사이의 상호 협력을 위한 통신으로 원격 프로시져 호출이 전통적으로 사용되고 있다. 분산 응용이 더욱 복잡해짐에 따라 최근 이동 에이전트 패러다임이 등장하였다. 이처럼 다양한 상호 협력을 위한 통신 패러다임이 등장함에 따라 각 패러다임의 성능에 대한 평가와 비교 연구가 이루어지고 있다. 그러나 기존의 연구에서 성능 평가를 위해 사용한 성능 모델들은 보안 서비스를 위한 평가 요소를 고려하고 있지 않기 때문에 실제 분산 환경을 제대로 반영하지 못한다. 분산 환경은 개방되어 있으므로 정보의 노출이나 도청과 같은 공격에 있어서 상당히 취약하다. 이러한 분산 환경에서 안전하게 작업을 수행하기 위해서는 여러 가지 공격으로부터 응용 프로그램이나 정보를 보호하기 위한 보안 서비스가 고려되어야 한다.

본 논문에서는 상호 협력을 위한 통신 패러다임 중 원격 프로시져 호출과 이동 에이전트의 성능을 평가하고 비교한다. 분산 응용 프로그램을 안전하게 수행하기 위해 고려해야 하는 보안 서비스에 관하여 알아보고, 이러한 보안 서비스를 적용한 새로운 성능 모델을 제시한다. N개의 데이타베이스 서버에서 사용자가 필요한 정보를 검색하는 작업을 Petri Net으로 모델링하고, 각 파라미터에 수치 값을 할당해서 수행 속도를 측정하여 두 패러다임의 성능을 비교한다.

본 논문에서 안전한 통신을 위하여 보안 서비스를 적용한 두 성능 모델의 비교 결과는 다음과 같다. 원격 프로시져 호출은 연산 비용이 높은 암호화 메커니즘을 포함하는 통신 횟수와 통신량이 많기 때문에 실행 시간이 급격하게 증가하지만, 이동 에이전트 패러다임은 통신 횟수와 통신량을 줄일 수 있으므로 실행 시간이 완만하게 증가하는 것을 살펴볼 수 있다.

키워드 : 분산 컴퓨팅, 이동 에이전트, 보안 서비스, 성능 모델, 성능 평가

Abstract The Remote Procedure Call(RPC) has been traditionally used for Inter Process Communication(IPC) among processes in distributed computing environment. As distributed applications have been complicated more and more, the Mobile Agent paradigm for IPC is emerged. Because there are some paradigms for IPC, researches to evaluate and compare the performance of each paradigm are issued recently. But the performance models used in the previous research did not reflect real distributed computing environment correctly, because they did not consider the evaluation elements for providing security services. Since real distributed environment is open, it is very vulnerable to a variety of attacks. In order to execute applications securely in distributed computing environment, security services which protect applications and information against the attacks must be considered.

· 본 논문은 한국과학재단의 특정기초연구(98 0102 1101-3) 연구비의 지원에 의한 것임.

† 비 회 원 : 한국전자통신연구원 정보보호연구본부 연구원
hansw@etri.re.kr

†† 비 회 원 : 한국정보보호진흥원 연구원
kmjeong@kisa.or.kr

††† 종신회원 : 초당대학교 컴퓨터과학과 교수
sbpark@chodang.ac.kr

†††† 종신회원 : 전남대학교 전산학과 교수
hslim@chonnam.chonnam.ac.kr

논문접수 : 2001년 10월 10일
심사완료 : 2002년 2월 18일

In this paper, we evaluate and compare the performance of the Remote Procedure Call with that of the Mobile Agent in IPC paradigms. We examine security services to execute applications securely, and propose new performance models considering those services. We design performance models, which describe information retrieval system through N database servers, using Petri Net. We compare the performance of two paradigms by assigning numerical values to parameters and measuring the execution time of two paradigms.

In this paper, the comparison of two performance models with security services for secure communication shows the results that the execution time of the Remote Procedure Call performance model is sharply increased because of many communications with the high costly cryptography mechanism between hosts, and that the execution time of the Mobile Agent model is gradually increased because the Mobile Agent paradigm can reduce the quantity of the communications between hosts

Key words : Distributed Computing, Mobile Agent, Security Service, Performance Model, Performance Evaluation

1. 서 론

전자 상거래, 인터넷 정보 검색과 같은 분산 응용 프로그램이 인터넷 사용의 증가와 분산 컴퓨팅 기술의 발전에 따라 점차 증가하고 있다. 이러한 분산 응용 프로그램에서 프로세스 사이의 상호협력을 위한 통신(Inter Process Communication)에는 전통적으로 원격 프로시져 호출(Remote Procedure Call)에 기반한 클라이언트-서버 모델이 사용되어 왔다[1,2,3]. RPC는 개념적으로 단순하고 구현이 간단하기는 하지만 낮은 대역폭과 통신량이 많은 네트워크 환경에는 적합하지 않다. 분산 환경과 응용 프로그램이 갈수록 복잡해지고, 사용자들은 더 많은 편리함을 추구하면서 새로운 형태의 상호 협력 메커니즘이 요구되었다. 그 결과 이동 에이전트(Mobile Agent)가 등장하게 되었다[4,5].

이동 에이전트는 이종의 네트워크 환경에서 호스트 사이를 자율적으로 이동하면서 사용자가 원하는 작업을 수행하는 소프트웨어이다[6,7]. 이동 에이전트는 이동성과 자율성이라는 특성을 가지고 호스트간의 통신량을 줄임으로써 네트워크 부하를 줄일 수 있고, 클라이언트와 서버간의 비동기성을 증가시켜 클라이언트와 서버 사이의 지속적인 연결을 요구하지 않으며, 사용자 중심적인 기능을 수행하기에 용이하다.

최근 연구에 따르면 이동 에이전트가 전통적인 클라이언트-서버 패러다임에 비해 정보검색, 네트워크 관리, 이동 컴퓨팅과 같은 응용 분야에서 유용함을 보이고 있다[5,6,7]. 또한 네트워크 부하와 실행시간에 영향을 주는 파라미터를 사용하여 IPC 패러다임의 성능을 비교, 분석하는 연구가 진행중이다[8,9,10,11]. 그러나 기존의 연구는 실제 분산 환경을 제대로 반영하지 못하고 있다. 실제 분산 컴퓨팅 환경은 인증된 사용자로의 가장(Masquerade), 정보 노출(Information Disclosure), 비인가된 사용자로부터의 무결성 침해(Integrity Violation)와 같은 많은 공격에 취약하기 때문이다[12]. 분산 컴퓨팅 환경에서 응용 프로그램을 안전하게 실행하기 위해서는 이러한 공격에 대응할 수 있는 적절한 조치가 취해져야 한다. 그리고 보안 서비스를 제공하기 위해 사용되는 연산은 다른 연산에 비해 속도가 매우 느리기 때문에 시스템의 성능에 큰 영향을 미친다. 그러나 지금까지 IPC 패러다임의 평가를 위해 수행된 연구들은 분산 환경에서 보안 서비스를 제공하기 위해 필요한 인자를 포함하지 않았다. 그러므로, 실제 분산환경에서 IPC 패러다임의 성능을 정확히 분석하기 위해서는 보안 서비스가 고려된 새로운 모델이 제시되어야 한다. 본 논문에서는 전통적인 클라이언트-서버 모델인 RPC와 이동 에이전트 패러다임에 보안 서비스를 고려한 성능 모델을 제시하고 두 모델을 비교, 분석하여 보안 서비스가 성능에 미치는 영향을 살펴볼 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 제시된 IPC 패러다임의 성능 모델과 평가에 관한 연구에 대해서 살펴보고, 3장에서는 본 논문에서 제시한 성능 모델에서 고려되는 보안 서비스에 대해 알아본다. 그리고 4장에서는 Petri Net을 사용하여 보안 서비스를 고려한 각 IPC 패러다임을 모델링하고 성능을 평가 비교한다. 마지막으로 5장에서 결론과 향후 연구방향에 대해서 기술한다.

2. 관련 연구

지금까지 IPC 패러다임의 성능 모델 및 평가에 관한 연구로는 RPC 패러다임, REV(Remote EValuation) 패러다임, 이동 에이전트 패러다임 등의 성능 모델을 모델링하고 성능 분석을 수행한 연구와 Java의 RMI로 구현된 클라이언트 서버 패러다임과 이동 에이전트의 성능

을 비교 평가한 연구가 있다. 각각을 살펴보면 다음과 같다.

M. Straßer와 M. Schwehm은 [11]에서 RPC와 이동 에이전트의 성능분석을 위해 두 가지 모델을 제시하고 있다. 이 두 모델은 호스트간의 상호작용을 모델링한 것과, 질의를 하는 호스트와 여러 호스트들 사이의 순차적인 상호작용을 모델링한 것이다. 이 논문에서는 에이전트의 선택도(selectivity of agent)라는 파라미터를 사용하는데, 이것은 원격 처리를 통해서 응답 메시지의 크기를 줄일 수 있는 에이전트의 능력을 표현하고 있다. 성능 실험 결과 두 호스트간 한번의 상호 작용이 발생했을 때에는 에이전트의 선택도값과 응답 메시지의 크기에 따라서 네트워크 부하나 실행시간이 큰 영향을 받는 것을 보여준다. RPC의 경우 에이전트의 선택도와 무관하기 때문에 네트워크 부하나 실행 시간에 변화가 없는 반면, 이동 에이전트의 경우에는 에이전트의 선택도값이 커지면 네트워크 부하나 실행 시간이 감소함을 보였다. 그리고 응답 메시지의 크기가 증가하는 경우, RPC 모델에서는 네트워크 부하나 실행 시간이 급격한 증가를 보이지만 이동 에이전트 모델에서는 완만한 증가함을 보였다. 호스트들 사이에 순차적인 상호작용이 발생했을 때는 순수한 RPC 모델이나 순수한 이동 에이전트 모델보다는 두 패러다임이 적절하게 섞인 모델이 최적의 성능을 나타냄을 보여준다.

[10]에서는 RPC 패러다임, REV 패러다임, 이동 에이전트 패러다임을 Petri Net으로 모델링하고 WebSPN을 사용하여 성능을 분석하였다. 이 논문에서는 세 가지 패러다임의 성능을 비교하기 위해서 N개의 데이타베이스 서버에서 필요한 정보를 탐색하는 모델을 제시하고 있다. 또한 동일한 데이타베이스 서버에서 연속적인 탐색 작업을 수행하는 정도를 표현하기 위해 Redo라고 하는 확률을 고려한 트랜지션을 사용하고 있다. 이 논문에서 이동 에이전트는 Redo 확률이 작을 때는 RPC 패러다임이나 REV 패러다임에 비해서 수행 속도가 늦지만, Redo 확률이 커짐에 따라 점차 두 패러다임보다 더 나은 성능을 보여준다. 결론적으로 이동 에이전트 패러다임은 네트워크의 속도, 응용의 특성과 같은 외부적 요인에 많이 의존함으로써 항상 최적의 성능을 보장하지는 않지만 통신의 서브 시스템이 충분히 빠르고 이동 컴퓨팅 또는 불안정한 네트워크 환경에서는 적합한 패러다임이 될 수 있음을 보여주고 있다

L. Ismail과 D. Hagimont는 [9]에서 JAVA 환경에서 RMI로 구현된 클라이언트-서버 패러다임과 이동 에이전트의 성능을 비교 평가하고 있다. 이 논문에서도 다른 논문과 마찬가지로 이동 에이전트의 성능은 응용의 특징에 따라 차이는 있지만 이동 에이전트의 데이타의 크기가 커지거나, 서버의 복잡한 작업을 요구할 때에는 다른 패러다임에 비하여 향상된 성능을 나타냄을 보여주고 있다.

지금까지 살펴본 기존의 연구들은 IPC 패러다임의 성능 모델을 모델링하고 평가하는데 보안 서비스를 위한 평가 요소들을 고려하지 않고 있다. 본 논문에서는 실제 분산 환경을 제대로 반영하기 위하여 보안 서비스가 고려된 IPC 패러다임의 성능 모델을 제시하고 비교 분석을 통하여 보안 서비스가 IPC 패러다임의 성능에 미치는 영향을 살펴볼 것이다.

3. 성능 모델에서 고려하는 보안 서비스

분산 컴퓨팅 환경에서 발생할 수 있는 대표적인 위협으로는 인증된 사용자로의 위장, 호스트간의 통신에 대한 도청에 의한 정보의 노출, 비인가된 사용자로부터의 무결성 침해 등이 있을 수 있다[13]. 이러한 위협을 제거하기 위한 보안 서비스로는 인증 서비스(Authentication Service), 비밀성 서비스(Confidentiality Service), 무결성 서비스(Integrity Service), 접근 통제 서비스(Access Control Service), 회계 감사 서비스(Audit Trail Service)[14] 등이 있다.

인증 서비스는 어떤 사용자가 자기 신분을 증명할 수 있는 수단을 제공하는 서비스로써 다른 보안 서비스를 제공하기 위한 바탕이 되는 서비스이다. 비밀성 서비스는 호스트간에 주고받는 정보를 암호화함으로써 공격자의 도청에 의한 정보 노출을 방지하는 서비스이고, 무결성 서비스는 공격자로부터 호스트간에 사용되는 정보의 훼손을 막는 서비스이다. 접근 통제 서비스는 비인가된 사용자로부터 시스템 자원의 불법적인 접근을 제어하는 서비스이고, 회계 감사 서비스는 보안에 관련된 사건을 감시·추적함으로써 시스템을 보호하는 서비스이다. 이러한 보안 서비스 중에서 접근 통제 서비스와 회계 감사 서비스는 호스트간의 통신에는 직접적으로 관련되지 않고 단지 시스템을 보호하기 위한 서비스이다. 본 논문에서는 개방 환경에서 호스트간의 상호 협력을 위한 통신에 대해 발생할 수 있는 위협을 제거하는 보안 서비스만을 고려한다.

본 논문에서는 호스트의 상호 인증 서비스를 제공하기 위하여 Secure RPC[14,15,16]에서 사용된 인증 프로토콜을 사용한다. 인증 프로토콜을 사용하기 위해 각 호스트의 공개키 목록을 가지고 있고, 신뢰할 수 있고, 외부의 공격으로부터 안전한 키 분배 센터(Key Distribution

Center)가 있다고 가정한다. 그리고 각 호스트는 동기화된 시간을 사용한다고 가정한다. 사용된 키 분배 및 상호 인증 프로토콜은 다음과 같다.

<키 분배 및 상호 인증 프로토콜>

[단계 1] $A \rightarrow KDC : ID_B$

- A는 키 분배 센터(KDC)에 B의 공개키를 요구하기 위해서 B의 ID를 전송한다.

[단계 2] $KDC \rightarrow A : S_{KDC}(ID_B , PK_B)$

- 키 분배 센터는 B의 공개키를 찾아서 자신의 비밀키로 서명하여 A에게 전송한다.

[단계 3] $A \rightarrow B : E_{CK}(SK), E_{SK}(TTL), E_{SK}(T, TTL+1)$

- CK(Common Key) : Diffie-Hellman의 키 공유 방식을 이용하여 A의 비밀키와 B의 공개키를 이용하여 생성한다.
- SK(Session Key) : A와 B의 통신에 사용될 세션키는 A에 의해서 생성된다.
- T : 상호 인증을 위해 사용되는 타임스탬프
- TTL : 타임스탬프의 유효기간을 나타내는 Time To Live.
- A는 세션키를 공통키로 암호화하고, 인증에 사용할 타임스탬프와 타임스태프의 유효기간을 나타내는 TTL을 세션키로 각각 암호화하여 B에게 전송한다.

[단계 4] $B \rightarrow KDC : ID_A$

- A의 메시지를 전송 받은 B는 키 분배 센터에 A의 공개키를 요청한다.

[단계 5] $KDC \rightarrow B : S_{KDC}(ID_A , PK_A)$

- 키 분배 센터는 A의 공개키를 찾아서 자신의 비밀키로 서명하여 B에게 전송한다.

[단계 6] $B \rightarrow A : E_{SK}(T-1)$

- Diffie-Hellman의 키 공유 방식을 사용하여 A의 공개키와 B의 비밀키를 이용하여 공통키를 생성한다.
- 생성된 공통키를 사용하여 세션키를 복호화한다. 세션키를 사용하여 타임스탬프와 TTL 값을 복호화한 후 그 값을 통해서 세션키의 유효성을 검사한다. 새로운 타임스탬프를 A에게 전송하기 위하여 세션키를 사용하여 암호화한다.

[단계 7] $A : D_{SK}(T-1)$

- 새로운 타임스탬프 값을 복호화하여 확인한 후 상호 인증을 마친다.

비밀성과 무결성 서비스를 제공하기 위해서, RPC인 경우에는 프로시져를 호출하는 요청 메시지와 프로시져 수행 후 되돌아가는 응답 메시지에 대해 암호화와 서명이 사용된다. 그러나 이동 에이전트인 경우 추가적인 고려사항이 필요하다. 이동 에이전트는 소프트웨어이기 때문에 프로그램 코드자체, 에이전트의 상태, 그리고 각 호스트 방문시 작업을 처리하고 얻게 되는 에이전트 데이타로 구성되어 있다. 또한 이동 에이전트 패러다임에서 보안은 악의적인 에이전트로부터 호스트를 보호하는 것과 악의적인 호스트로부터 에이전트를 보호하는 것으로 크게 생각해 볼 수 있다[12,17]. 악의적인 에이전트로부터 호스트를 보호하는 방법은 호스트에서 적절한 접근 통제와 같은 방법을 이용하면 가능하지만 악의적인 호스트로부터 에이전트를 보호하는 일은 해결하기 힘든 문제이다. 악의적인 호스트는 코드를 분석하고 소스를 알아낼 수 있고, 이동 에이전트에 포함되어 있는 상태값이나 데이타값을 불법적으로 변경시킬 수도 있다[18]. 이러한 상황에서 이동 에이전트를 보호하기 위해서는 호스트를 방문할 때마다 코드에 대한 코드 소유자의 서명검증 작업을 통해 초기 작성된 코드에 대한 무결성을 제공해야 한다. 또한 각 호스트 방문시 발생하는 데이타와 상태에 대한 비밀성과 무결성을 제공하기 위하여 에이전트의 이동시에 암·복호화, 서명·검증 등이 이루어져야 한다.

4. 보안 서비스를 고려한 성능 모델과 성능 평가

4.1 성능 모델

그림 1은 RPC에 보안 서비스가 고려된 Secure RPC의 정보 검색 과정을 보여주고 있다. 클라이언트는 데이타베이스 서버 1부터 데이타베이스 서버 N까지 순차적으로 방문하면서 정보 검색을 수행한다. 데이타베이스 서버로부터 정보 검색 결과를 가지고 와서 클라이언트에서 사용자의 요구에 적합한 형태로 재처리한다. 키 분배 센터는 정보 검색이 수행되기 이전, 클라이언트와 데이타베이스 서버간의 상호 인증시 공개키를 분배하기 위하여 클라이언트와 데이타베이스 서버와 통신한다.

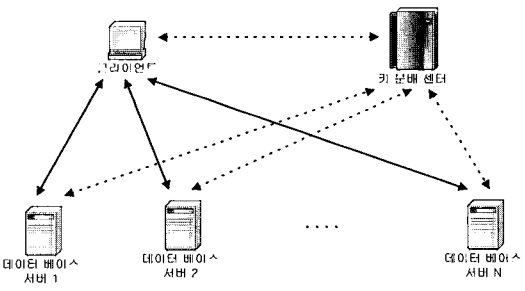

그림 1 Secure RPC의 정보 검색

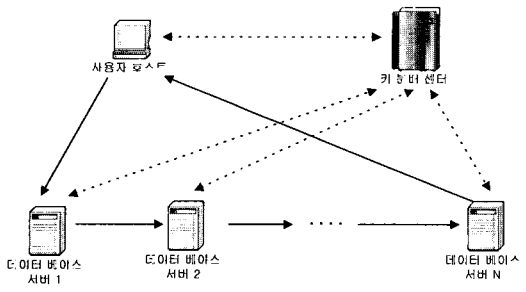

그림 2 Secure MA의 정보 검색

그림 2는 이동 에이전트에 보안 서비스가 고려된 Secure MA의 정보 검색 과정을 보여주고 있다. 키 분배 센터의 역할은 Secure RPC와 같다. 그러나 정보 검색을 수행하는 방식은 클라이언트와 데이타베이스 서버 간 지속적인 통신이 설정되어야 하는 RPC 패러다임과는 달리 사용자 호스트에서 출발한 이동 에이전트는 자율적으로 각 데이타베이스 서버를 이동하면서 사용자가 원하는 작업을 수행한 후 N번째 데이타베이스 서버에서 최종적인 결과만을 사용자 호스트에게 되돌려준다.

정보 검색 작업은 동일 데이타베이스 서버에서 여러 번의 검색 작업을 수행하기도 하고, N번째 데이타베이스 서버까지 모두 방문되고 사용자가 필요한 정보를 획득하면 종료한다. 본 논문에서는 두 패러다임의 정보 검색 작업을 Petri Net을 사용하여 모델링한다.

4.1.1 Secure RPC 성능 모델

그림 3은 Petri Net으로 모델링한 Secure RPC 성능 모델이다. 성능 모델의 동작 과정과 사용된 트랜지션과 플레이스의 역할을 살펴보면 다음과 같다. 클라이언트가 데이타베이스 서버의 정보를 검색하기 위해서 통신을 하고자 할 때, 먼저 클라이언트는 데이타베이스 서버와 상호 인증을 수행하기 위하여 키 분배 센터에 데이타베이스 서버의 공개키를 요청한다(ReqDBSKey). 키 분배 센터는 데이타베이스 서버의 공개키를 찾아내고, 자신의 비밀키로 서명을 해서 클라이언트에게 전송한다(ResDBSKey). 그리고 클라이언트는 키 분배 센터로부터 받은 자료를 통해 데이타베이스 서버에게 인증을 요청한다(ReqAuth). 데이타베이스 서버는 클라이언트와 상호 인증을 수행하기 위하여 클라이언트의 공개키를 키 분배 센터에 요청하고(ReqClientKey), 키 분배 센터는 요청에 따라 클라이언트의 공개키를 전송한다(ResClientKey). 데이타베이스 서버는 상호 인증 프로토콜에 따라 인증 요청에 대한 응답을 하고(ResAuth), 클라이언트가 응답을 확인하면 상호 인증이 이루어진다(End Auth).

인증 후에는 클라이언트와 데이타베이스 서버간에 통신이 이루어진다. 클라이언트는 정보 검색 요청 메시지에 자신의 비밀키로 서명을 하고, 인증시에 만들어진 세션키를 사용하여 암호화한다(SignEncrypt Req). 서명과 암호화가 이루어진 요청메시지는 데이타베이스 서버로 전송되고(Transfer Req), 데이타베이스 서버에서 복호화와 서명검증이 이루어진다(DecryptVerify Req). 데이타베이스 서버는 클라이언트의 요청에 따라 자료를 검색하고(Search), 검색 결과를 안전하게 클라이언트에게 전송하기 위해서 응답 메시지에 자신의 비밀키로 서명하고, 세션키로 암호화한다(SignEncrypt Rep). 응답 메시지는 클라이언트에게 전송되고(Transfer Rep), 클라이언트가 이용하기 위하여 복호화와 서명검증이 이루어진다(Decrpty Verify Rep). 데이타베이스 서버로부터 받은 정보 검색 결과를 사용자의 요구에 맞게 재처리하고(Filter), 다시 한번 동일한 데이타베이스로 정보 검색 요청 메시지를

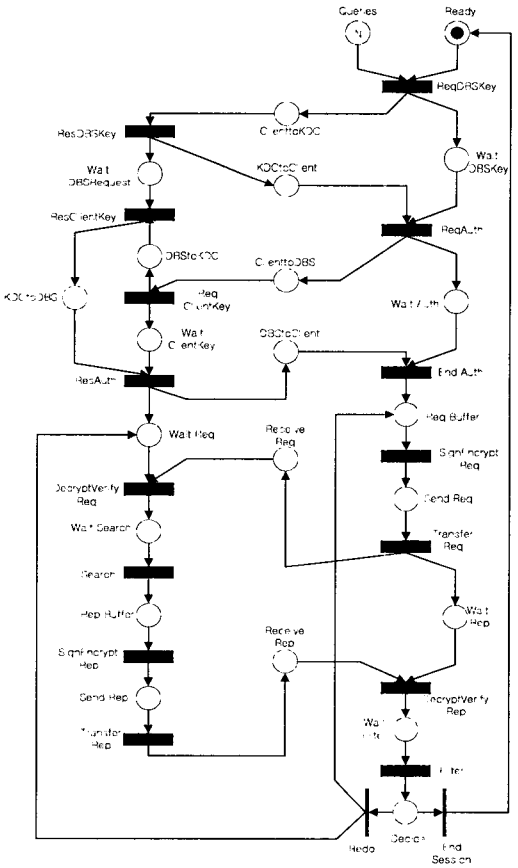

그림 3 Petri Net으로 모델링한 Secure RPC 성능 모델

보낼지(Redo), 아니면 다음 데이타베이스 서버를 이용할 것인지(End Session)를 결정하게 된다. 확률을 고려한 Redo 트랜지션이 선택되면, 클라이언트와 데이타베이스 서버 사이에는 이미 상호 인증이 되어있으므로 비밀성과 무결성 서비스를 제공하는 작업만 해주면 된다. 이 과정은 N번째의 데이타베이스 서버를 방문할 때까지 반복된다. Queries 플레이스는 N개의 데이타베이스 서버를 나타내고 있고, Ready 플레이스는 Queries 플레이스에 있는 토큰을 하나씩 발화시키기 위하여 사용된다.

4.1.2 Secure MA 성능 모델

그림 4는 Petri Net으로 모델링한 Secure MA 성능 모델이다. 성능 모델의 동작 과정과 사용된 트랜지션과 플레이스의 역할을 살펴보면 다음과 같다. 정보 검색을 위해서 이동 에이전트가 데이타베이스 서버로 이동하기 전에 사용자 호스트와 첫 번째 데이타베이스 서버 사이에 상호 인증이 이루어져야한다. 사용자 호스트는 첫 번째 데이타베이스 서버의 공개키를 키 분배 센터에 요청하고(ReqDBSKey1), 키 분배 센터는 첫 번째 데이타베이스 서버의 공개키를 전송해 준다(ResDBSKey1). 그

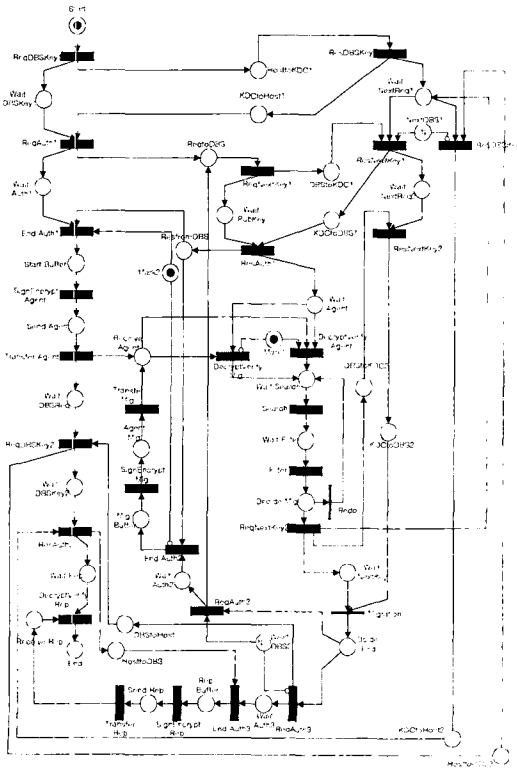

그림 4 Petri Net으로 모델링한 Secure MA 성능 모델

리고 사용자 호스트는 첫 번째 데이타베이스 서버에게 인증을 요청한다(ReqAuth1). 첫 번째 데이타베이스 서버는 사용자 호스트의 공개키를 키 분배 센터에 요청하고(ReqNextKey1), 키 분배 센터는 사용자 호스트의 공개키를 보내준다(ResNextKey1). 첫 번째 데이타베이스 서버는 인증 요청에 대한 응답을 하고(ResAuth1), 사용자 호스트는 인증 확인을 수행한다(End Auth1). 상호 인증시 만들어진 세션키를 사용해 이동 에이전트는 암호화되고, 사용자 호스트의 비밀키로 서명되어(Sign Encrypt Agent) 첫 번째 데이타베이스 서버로 이동한다(Transfer Agent).

데이타베이스 서버에 도착한 이동 에이전트는 복호화되고 서명확인 작업을 거친 후(DecryptVerify Agent), 정보 검색 작업을 수행하고(Search), RPC 패러다임과는 달리 사용자의 요구에 맞는 재처리를 데이타베이스 서버에서 수행한다(Filter). 작업을 수행한 후 이동 에이전트는 정보 검색 작업을 같은 데이타베이스 서버에서 계속 수행할 지(Redo), 아니면 다음 데이타베이스 서버로 이동할지를 결정한다. 그리고 데이타베이스 서버간의 상호 인증을 하기 위하여 다음 데이타베이스 서버의 공개키를 키 분배 센터에게 요청한다(ReqNextKey2). 키 분배 센터는 요청된 서버의 공개키를 전송해 준다(ReqNextKey2).

본 논문의 성능 모델에서 이동 에이전트가 데이타베이스 서버간을 이동할 때에는 에이전트 코드, 상태값, 데이타를 모두 포함하고 있지만, 마지막 데이타베이스 서버에서 사용자의 요청 작업이 모두 완료되면 사용자가 요구한 작업의 결과값만을 사용자 호스트로 전송한다. 따라서 그 두 가지 작업을 구분하기 위하여 Decide End 플레이스와 Next DBS2 플레이스가 사용된다. Next DBS2 플레이스의 토큰 $N-1$은 남아 있는 방문해야 할 데이타베이스 서버의 개수를 의미한다. 데이타베이스 서버간의 통신시에는 인증을 위하여 ReqAuth2 트랜지션이 선택되고, 마지막 데이타베이스 서버와 사용자 호스트 사이의 통신을 위해서는 ReqAuth3 트랜지션이 선택되어 진다. ReqAuth2 트랜지션이 선택되어진 경우 연속적인 인증 절차는 ReqNextKey1, ResNextKey1, ResAuth1, End Auth2 트랜지션으로 이어진다. Mark2 플레이스는 End Auth2 트랜지션이 선택되어지도록 하는 역할을 하고 있다. 상호 인증이 완료된 후 이동 에이전트는 암호화와 서명작업이 이루어진다(SignEncrypt Mig). 그리고나서 다음 데이타베이스 서버로 이동한다(Transfer Mig). 이동한 에이전트는 복호화와 서명검증 작업을 거치고(DecryptVerify Mig) 정보 검색 작업을 수행한다. Mark1 플레이스는 DecyrptVerify Mig 트랜지션이 선택되어지도록 하는 역

할을 하고 있다. 이러한 절차는 N번째 데이타베이스 서버에 도착할 때까지 반복한다.

마지막 데이타베이스 서버에서 검색 결과가 사용자 호스트로 전송될 때, 즉 ReqAuth3 트랜지션이 선택되어진 경우는 인증 요청이 사용자 호스트로 보내진다. 사용자 호스트는 마지막 데이타베이스 서버의 공개키를 요구하고(ReqDBSKey2), 키 분배 센터는 공개키를 보내준다(ResDBSKey2). 이때 NextDBS1 플레이스는 ResDBSKey2 트랜지션이 선택되어지도록 조정하는 역할을 하고 있다. ResAuth2와 End Auth3 트랜지션을 통해 인증 과정이 끝난 후, 정보 검색 결과는 서명되고 암호화되어 사용자 호스트에게 보내진다(SignEncrypt Rep, Transfer Rep). 사용자 호스트로 전송된 검색 결과는 복호화와 서명검증 절차가 끝난 후 사용자가 이용할 수 있게 된다(DecryptVerify Rep).

4.1.3 파라미터

Secure RPC와 Secure MA 성능 모델을 분석하기 위하여 다음과 같은 보안 서비스가 고려된 파라미터를 사용하였다.

- D_{Id} : 인증 프로토콜에 참여하는 각 호스트의 ID 크기를 나타낸다.
- D_{PK} : 인증 프로토콜에 참여하는 각 호스트의 공개키의 크기를 나타낸다.
- D_{SK} : 인증 프로토콜에서 호스트간의 통신에 사용되는 세션키의 크기를 나타낸다.
- D_T : 인증 프로토콜에서 사용되는 타임스탬프의 크기를 나타낸다.
- D_{TTL} : 인증 프로토콜에서 사용되는 Time To Live의 크기를 나타낸다.
- D_{Req} : Secure RPC 모델에서 클라이언트에서 데이타베이스 서버에 요청하는 메시지에 암호화를 통한 패딩과 서명이 부가된 크기. 실제 요청하려는 메시지에 비해 서명과 패딩은 크기가 매우 작으므로 여기에서 크기는 0이라고 가정한다.
- D_{Rep} : Secure RPC와 Secure MA 모델에서 데이타베이스 서버에서 클라이언트에 보내는 응답 메시지에 암호화를 통한 패딩과 서명이 부가된 크기. 최소값과

표 1 Secure RPC 성능 모델의 각 트랜지션에 할당된 수식

트랜지션	형태	수식
ReqDBSKey, ReqClienKey	Deterministic	$\frac{D_{Id}}{R_{Th}}$
ResDBSKey, ResClientKey	Exponential	$R_{Se}+\frac{(D_{Id}+D_{PK})}{R_S}+\frac{(D_{Id}+D_{PK})}{R_{Th}}$
ReqAuth	Deterministic	$T_{GK}+T_{DH}+\frac{(D_{Id}+D_{PK})}{R_V}+\frac{(D_{SK}+D_T+2D_{TTL})}{R_E}+\frac{(D_{SK}+D_T+2D_{TTL})}{R_{Th}}$
ResAuth	Deterministic	$T_{DH}+\frac{(D_{Id}+D_{PK})}{R_V}+\frac{D_T}{R_E}+\frac{D_T}{R_{Th}}+\frac{(D_{SK}+D_T+2D_{TTL})}{R_D}$
End Auth	Deterministic	$\frac{D_T}{R_D}$
SignEncrypt Req	Deterministic	$\frac{D_{Req}}{R_S}+\frac{D_{Req}}{R_E}$
Transfer Req	Deterministic	$\frac{D_{Req}}{R_{Th}}$
DecryptVerify Req	Deterministic	$\frac{D_{Req}}{R_D}+\frac{D_{Req}}{R_V}$
Search	Exponential	R_{Se}
SignEncrypt Rep	Uniform	$\frac{D_{Rep}}{R_S}+\frac{D_{Rep}}{R_E}$
Transfer Rep	Uniform	$\frac{D_{Rep}}{R_{Th}}$
DecryptVerify Rep	Uniform	$\frac{D_{Rep}}{R_D}+\frac{D_{Rep}}{R_V}$
Filter	Exponential	R_{Rf}

최대값이 있는 정규분포를 따른다고 가정한다. 실제 요청크기에 비해 서명과 패딩의 크기는 매우 작으므로 여기에서는 크기를 0이라고 가정한다.

- D_{Code} : 이동 에이전트의 코드 크기를 나타낸다.
- D_{State} : 이동 에이전트의 상태 크기를 나타낸다.
- D_{Data} : 데이타베이스 서버에서 정보 검색 작업과 정보 가공 작업을 처리하고 나서 구해지는 이동 에이전트의 데이타 크기. RPC 패러다임과는 달리 이동 에이전트는 다양한 작업을 데이타베이스 서버에서 실행할 수 있다. 본 논문에서 이동 에이전트는 데이타베이스 서버에서 정보 검색 뿐만 아니라 검색 결과를 재처리한다. 검색 결과에 대한 재처리 과정이 많을수록 데이타의 크기가 줄어든다고 가정한다. 본 논문에서는 이동 에이전트가 데이타베이스 서버에서 자율적으로 수행하는 재처리 작업의 정도를 σ로 표현하기로 한다. 가정에 따라 이동 에이전트 데이타의 크기는 RPC의 응답 메시지에 비해서 그 크기가 줄어든다고 볼 수 있다. 따라서 크기는 $D_{Rep}(1-\sigma)$ {σ | 에이전트의 선택도 $(0\le\sigma\le1)$}로 볼 수 있다. 정규분포를 따른다고 가정한다.
- R_{Se} : 데이타베이스 서버를 방문해서 정보를 검색하는 처리율. 고정된 값으로 지수분포를 따른다고 가정한다.
- R_{Rf} : 데이타베이스 서버에서 검색된 정보를 사용자의 요구에 맞게 가공하는 작업의 처리율. RPC인 경우 클라이언트로 응답 메시지가 전송된 후 가공 작업은 수행되고, 이동 에이전트인 경우 각 데이타베이스

표 2 Secure MA 성능 모델의 각 트랜지션에 할당된 수식

트랜지션	형태	수식
ReqDBSKey1, ReqDBSKey2, ReqNextKey1, ReqNextKey2	Deterministic	$\frac{D_{Id}}{R_{Th}}$
ResDBSKey1, ResDBSKey2, ResNextKey1, ResNextKey2	Exponential	$R_{Se}+\frac{(D_{Id}+D_{PK})}{R_S}+\frac{(D_{Id}+D_{PK})}{R_{Th}}$
ReqAuth1, ReqAuth2, ReqAuth3	Deterministic	$T_{GK}+T_{DH}+\frac{(D_{Id}+D_{PK})}{R_V}+\frac{(D_{SK}+D_T+2D_{TTL})}{R_E}+\frac{(D_{SK}+D_T+2D_{TTL})}{R_{Th}}$
ResAuth1, ResAuth2	Deterministic	$T_{DH}+\frac{(D_{Id}+D_{PK})}{R_V}+\frac{D_T}{R_E}+\frac{D_T}{R_{Th}}+\frac{(D_{SK}+D_T+2D_{TTL})}{R_D}$
End Auth1, End Auth2, End Auth3	Deterministic	$\frac{D_T}{R_D}$
SignEncrypt Agent	Deterministic	$\frac{(D_{Code}+D_{State})}{R_S}+\frac{(D_{Code}+D_{State})}{R_E}$
Transfer Agent	Deterministic	$\frac{D_{Code}+D_{State}}{R_{Th}}$
Decryptverify Agent	Deterministic	$\frac{(D_{Code}+D_{State})}{R_D}+\frac{(D_{Code}+D_{State})}{R_V}$
Search	Exponential	R_{Se}
Filter	Exponential	R_{Rf}
SignEncrypt Mig	Uniform	$\frac{(D_{Data}+D_{State}+D_{Code})}{R_S}+\frac{(D_{Data}+D_{State}+D_{Code})}{R_E}$
Transfer Mig	Uniform	$\frac{D_{Data}+D_{State}+D_{Code}}{R_{Th}}$
DecryptVerify Mig	Uniform	$\frac{(D_{Data}+D_{State}+D_{Code})}{R_D}+\frac{(D_{Data}+D_{State}+D_{Code})}{R_V}$
SignEncrypt Rep	Uniform	$\frac{D_{Data}}{R_S}+\frac{D_{Data}}{R_E}$
Transfer Rep	Uniform	$\frac{D_{Data}}{R_{Th}}$
DecryptVerify Rep	Uniform	$\frac{D_{Data}}{R_D}+\frac{D_{Data}}{R_V}$

서버에서 정보 검색 작업 후 수행된다. 고정된 값으로 지수분포를 따른다고 가정한다.

- R_E : 메시지의 보호를 위한 암호화 작업의 처리율. 각 호스트나 서버의 성능에 관계없이 일정하다고 가정한다.
- R_D : 암호화된 메시지에 대한 복호화 작업의 처리율. 각 호스트나 서버의 성능에 관계없이 일정하다고 가정한다.
- R_S : 메시지의 보호를 위해서 메시지에 대한 서명 작업의 처리율. 각 호스트나 서버의 성능에 관계없이 일정하다고 가정한다.
- R_V : 서명된 메시지에 대해서 서명 검증 작업의 처리율. 각 호스트나 서버의 성능에 관계없이 일정하다고 가정한다.
- R_{Th} : 네트워크로 메시지가 이동하는 전송처리율. 네트워크 상태에 관계없이 일정하다고 가정한다.
- T_{GK} : 세션키를 생성하는 소요되는 시간을 나타낸다.
- T_{DH} : Diffie-Hellman의 키 교환 방식을 적용하여 공통키를 만들어내는데 소요되는 시간을 나타낸다.

본 논문에서 각 성능 모델의 성능 평가는 전체 작업의 수행 시간을 측정하여 이루어진다. 작업 수행 시간을 측정하기 위해서 시간이 고려되는 트랜지션에 각 트랜지션의 역할에 따라서 수식이 할당되어야 한다. 표 1은 Secure RPC 성능 모델의 트랜지션에 할당되는 수식을 나타내고, 표 2는 Secure MA 성능 모델의 트랜지션에 할당되는 수식을 보여주고 있다.

4.1.4 수식에 의한 표현

본 논문에서는 각 성능 모델의 성능평가를 위하여 수행 시간을 측정해야 한다. 작업의 수행 시간 측정을 위하여 각 성능 모델은 수식으로 표현하면 다음과 같이 할 수 있다. 아래 수식에서 사용되는 p는 연속된 정보 검색 작업이 동일한 데이타베이스 서버에서 수행될 확률(각 모델에서 Redo 트랜지션이 선택되어질 확률)을 나타낸다.

Secure RPC와 Secure MA 성능 모델에 공통으로 사용되는 키 분배 및 상호 인증을 위한 프로토콜에 소요되는 시간은 다음과 같이 수식으로 표현할 수 있다. 인증 프로토콜의 단계 1과 단계 4는 통신에 참여하는 상대 호스트의 공개키를 키 분배 센터에 요청하는 단계인데 수행 시간은 수식 1과 같다. 수식 2는 인증 프로토콜의 단계 2와 단계 5로써 키 분배 센터에서 각 호스트로 상대방의 공개키를 전송하는데 소요되는 시간을 나타낸다. 수식 3은 인증 프로토콜의 단계 3을 처리하는데 소요되는 시간을, 수식 4는 인증 요청을 처리하고 응답하는 인증 프로토콜의 단계 6을 수행하는데 소요되는 시간을 나타낸다. 수식 5는 인증 프로토콜의 단계 7로써 상호 인증이 완료되는데 소요되는 시간을 나타낸다.

$$T_{ReqPK} = \frac{D_{Id}}{R_{Th}} \tag{1}$$

$$T_{ResPK} = R_{Se} + \frac{(D_{Id}+D_{PK})}{R_S} + \frac{(D_{Id}+D_{PK})}{R_{Th}} \tag{2}$$

$$T_{ReqAuth} = T_{GK} + T_{DH} + \frac{(D_{Id}+D_{PK})}{R_V} + \frac{(D_{SK}+D_T+2D_{TTL})}{R_E} + \frac{(D_{SK}+D_T+2D_{TTL})}{R_{Th}} \tag{3}$$

$$T_{ResAuth} = T_{DH} + \frac{(D_{Id}+D_{PK})}{R_V} + \frac{D_T}{R_E} + \frac{D_T}{R_{Th}} + \frac{(D_{SK}+D_T+2D_{TTL})}{R_D} \tag{4}$$

$$T_{EndAuth} = \frac{D_T}{R_D} \tag{5}$$

Secure RPC 인 경우 수행시간은 정보 검색 작업 시간과 보안 서비스를 제공하기 위한 시간으로 나누어 볼 수 있는데, 데이타베이스 서버에서 정보 검색 작업이 수행될 때마다 다음과 같은 시간이 소요된다.

$$T_{RPCTh} = \frac{D_{Req}}{R_{Th}} + \frac{D_{Rep}}{R_{Th}} + \frac{1}{R_{Se}} + \frac{1}{R_{Rf}} \tag{6}$$

클라이언트와 데이타베이스 서버 사이에 통신을 하기 위하여 키 분배 및 상호 인증이 수행되어야 한다. 소요되는 시간은 수식 1부터 수식 5를 고려하여 다음과 같이 나타난다.

$$T_{RPCAuth} = 2T_{ReqPK} + 2T_{ResPK} + T_{ReqAuth} + T_{ResAuth} + T_{EndAuth} \tag{7}$$

요청 메시지와 응답 메시지를 보호하기 위해서 암호화와 복호화 그리고 메시지에 대한 서명과 서명 검증 작업이 요구된다. 비밀성과 무결성 서비스를 제공하기 위한 소요시간은 다음과 같다.

$$T_{RPCSec} = \frac{D_{Req}}{R_S} + \frac{D_{Req}}{R_E} + \frac{D_{Req}}{R_D} + \frac{D_{Req}}{R_V} + \frac{D_{Rep}}{R_S} + \frac{D_{Rep}}{R_E} + \frac{D_{Rep}}{R_D} + \frac{D_{Rep}}{R_V} \tag{8}$$

N개의 데이타베이스 서버를 방문하고, 동일한 데이타베이스 서버에서 연속적인 작업을 수행할 확률 p를 고려할 경우, Secure RPC 성능 모델을 수행하는데 소요되는 시간은 다음과 같이 나타낼 수 있다.

$$T_{SRPC} = N \cdot T_{RPCAuth} + N\left(\frac{1}{1-p}\right)(T_{RPCTh} + T_{RPCSec}) \tag{9}$$

이동 에이전트인 경우를 살펴보면, 사용자 호스트에서 정보 검색 작업을 위한 이동 에이전트가 생성되었을 때

에는 에이전트의 코드와 에이전트 초기 상태만을 가지고 있다. 따라서 이동 에이전트의 초기 데이타의 크기는 다음과 같다.

$$D_{Init} = D_{State} + D_{Code} \quad (10)$$

이동 에이전트가 데이타베이스 서버사이에서 이동 중에는 에이전트의 코드와 상태뿐만 아니라 각 데이타베이스 서버에서 정보 검색 작업 수행 후 얻어진 결과 데이타를 포함하고 있으므로 이동중인 이동 에이전트의 크기는 다음과 같다.

$$D_{Mig} = D_{Code} + D_{State} + D_{Data} \quad (11)$$

Secure MA 성능 모델의 수행시간은 Secure RPC 성능 모델처럼 정보 검색 작업 수행시간과 보안 서비스 수행시간으로 나누어 볼 수 있다. D_{Mig} 의 N-1회의 이동과 D_{Init}와 D_{Data}의 한번의 이동, 동일한 데이타베이스 서버에서 연속적인 작업을 수행할 확률 p를 고려했을 때 정보 검색 작업 수행시간은 다음과 같다.

$$T_{MATh} = \frac{D_{Init}}{R_{Th}} + \frac{D_{Data}}{R_{Th}} + (N-1)\left\{\left(\frac{1}{R_{Se}} + \frac{1}{R_{Rf}}\right)\left(\frac{1}{1-p}\right) + \frac{D_{Mig}}{R_{Th}}\right\} \quad (12)$$

키 분배 및 상호 인증을 하는데 소요되는 시간은 다음과 같이 Secure RPC와 동일하다.

$$T_{MAAuth} = 2T_{ReqPK} + 2T_{ResPK} + T_{ReqAuth} + T_{ResAuth} + T_{EndAuth} \quad (13)$$

이동 에이전트의 초기값에 대한 서명, 암호화, 복호화가 필요하고, 에이전트 코드에 대한 무결성을 제공하기 위하여 N개의 데이타베이스 서버에서 각각 코드에 대한 서명검증이 필요하다. 그리고 에이전트가 이동 중에는 각 데이타베이스 서버에서 발생한 상태와 작업을 처리한 결과를 가지고 있는 데이타의 보호를 위해 암호화, 복호화, 서명과, 서명검증 작업이 수행되어야 한다. 마지막으로 최종 검색 결과 데이타가 사용자 호스트로 보내질 때 그 데이타의 보호를 위한 보안 서비스가 제공되어야 한다. 결국 비밀성과 무결성 서비스를 수행하기 위한 시간은 다음과 같다.

$$T_{MASec} = \frac{D_{Init}}{R_S} + \frac{D_{Init}}{R_E} + \frac{D_{Init}}{R_D} + N\frac{D_{Code}}{R_V} + \frac{D_{Data}}{R_S} + \frac{D_{Data}}{R_E} + \frac{D_{Data}}{R_D} + \frac{D_{Data}}{R_V} + (N-1)\left\{\frac{(D_{State}+D_{Data})}{R_S} + \frac{(D_{State}+D_{Data})}{R_E} + \frac{(D_{State}+D_{Data})}{R_D} \frac{(D_{State}+D_{Data})}{R_V}\right\} \quad (14)$$

N+1회의 상호 인증 시간과 수식 12와 수식 13의 합이 결국 Secure MA의 수행 시간이 된다.

$$T_{SMA} = T_{MATh} + (N+1)T_{MAAuth} + T_{MASec} \quad (15)$$

4.2 성능 평가 및 비교

4.2.1 파라미터에 적용된 수치 값

실제적인 성능 평가를 위해서는 각 패러다임을 모델링할 때 사용된 각 파라미터에 실제적인 수치 값을 대입해야 한다. 본 논문에서 쓰이는 파라미터 중 D_{Req}, D_{Rep}, D_{Code}, D_{State}, R_{Se}, R_{Rf}의 수치 값은 IPC 패러다임의 성능을 분석한 [10]에서 사용한 수치 값을 참조하였고 보안 서비스와 관련된 수치 값, 즉 R_E, R_D, R_S, R_V, T_{GK}, T_{DH} 등은 소프트웨어로 구현된 암호화 메커니즘에 다양한 크기의 데이타 값을 적용한 결과의 평균값을 사용하였다. 보안 서비스를 제공하기 위해 사용된 암호화 메커니즘은 다음과 같다.

- DSA : 무결성 서비스를 제공하기 위해, 서명과 검증시 사용된다. 본 논문에서는 1024비트 키를 사용하고, 공개키로 사용되는 p는 1024비트, q는 160비트, 베이스로 쓰이는 g는 1024비트가 사용되었다. 또한 속도향상을 위하여 랜덤 변수를 생성하는 작업은 미리 수행되었다고 가정한다[19].
- IDEA : 비밀성 서비스를 제공하고, 인증 프로토콜의 암호화와 복호화 작업이 필요한 곳에서 사용된다. 본 논문에서는 128비트 키를 사용하였다. 이 키는 두 호스트간의 세션키로 사용되기 위하여 생성된다.
- Diffie Hellman 키 공유 방식 : 인증 프로토콜에서 두 호스트간의 공통키를 생성하기 위해서 사용된다.

표 3 성능 모델에 사용되는 파라미터의 수치 값

파라미터	의 미	값
N	데이타베이스 서버의 수	10
D_{Id}	호스트의 ID 크기	8Byte
D_{PK}	공개키의 크기	128Byte
D_{SK}	세션키의 크기	16Byte
D_T	타임스탬프의 크기	8Byte
D_{TTL}	TTL의 크기	4Byte
D_{Req}	요청 메시지의 크기	1KByte
D_{Rep}	응답 메시지의 크기	(1KByte, 1MByte)
D_{Code}	이동 에이전트의 코드 크기	39KByte
D_{State}	이동 에이전트의 상태 크기	4KByte
R_{Se}	정보 검색 처리율	4 회/s
R_{Rf}	정보 가공 처리율	4 회/s
R_{Th}	네트워크 처리율	1000kbps
R_E	암호화 처리율	1.6MByte/초
R_D	복호화 처리율	1.6MByte/초
R_S	서명 처리율	4.36MByte/초
R_V	서명검증 처리율	2.03MByte/초
T_{GK}	세션키 생성 시간	33.123 초
T_{DH}	키 공유 처리 시간	2.366 초

본 논문에서는 1024비트 키를 사용한다.

본 논문에서는 각 암호화 메커니즘의 수행시간을 측정하기 위하여 JAVA1.3을 사용하였고, 암호화 메커니즘을 위해서 JCE1.2.1[20], ABA JCE[21]를 사용하였다. 사용된 시스템은 인텔 펜티엄III 450Mhz, 256M RAM, 윈도우 98 운영체제를 이용하였다. 각 파라미터에 적용된 수치 값은 표 3과 같다.

4.2.2 각 패러다임에 대한 성능 평가

본 논문에서는 표 3에서 설정한 파라미터의 수치 값을 성능 모델을 표현한 수식에 적용함으로써 RPC와 이동 에이전트 패러다임을 비교 평가한다. 그림 5는 에이전트의 선택도인 σ의 값이 0.1일 때, 동일한 데이타베이스 서버에서 연속적인 작업을 수행할 확률(Redo 트랜지션이 선택될 확률) p에 따른 두 패러다임의 작업 수행 시간의 변화를 나타낸다. 그리고, 그림 6은 σ의 값이 0.9일 때 확률 p에 따른 두 패러다임의 작업 수행 시간의 변화를 보여주고 있다. σ는 이동 에이전트의 선택도로서 이동 에이전트가 호스트에서 자율적으로 다양한 작업을 수행하는 정도를 나타내고 있다. σ 값의 증가는 이동 에이전트가 데이타베이스 서버에서 검색한 정보를 가공하여 그 결과 값의 크기를 감소시키는 것을 의미하고 이것은 호스트 사이의 통신 부하를 줄이는 역할을 한다.

σ의 값이 0.1일 때는 확률 p의 값이 0.4이상일 때 이동 에이전트의 작업 수행 시간이 RPC 보다 작아지는 것을 볼 수 있다. σ의 값이 0.9일 때는 이동 에이전트의 작업 수행 시간이 RPC 보다 항상 더 작은 것을 알 수 있다. 동일한 데이타베이스 서버에서 연속적으로 정보 검색 작업을 수행할 경우 RPC 패러다임인 경우, 클라이언트와 해당 데이타베이스 서버 사이에 지속적인 정보 교환이 이루어져야 한다. 반면 이동 에이전트 패러다임인 경우, 이동 에이전트는 해당 데이타베이스 서버에서 자율적으로 사용자 대신 사용자의 작업을 수행하기 때문에 사용자 호스트와 데이타베이스 서버 사이에 추가적인 통신이 이루어질 필요가 없다. 보안 서비스를 제공하는 암호화 메커니즘은 비용이 많이 드는 연산이다.

따라서 확률 p의 값이 커짐에 따라 클라이언트와 서버 사이에 통신 횟수와 통신량이 증가하는 RPC 패러다임의 경우 연산 비용이 큰 보안 서비스가 지속적으로 요구되므로 수행 시간이 급격하게 증가한다. 반면 이동 에이전트 패러다임의 경우 확률 p의 값이 증가하더라도 이 값에 비례해서 통신 횟수와 통신량이 증가하지 않으므로 수행 시간에 큰 변화가 없다.

그림 7은 확률 p를 0.3으로 했을 때 이동 에이전트 선택도 σ의 변화에 따른 두 패러다임의 수행 시간의 변화를 보여 주고 있다. 여기에서 σ는 이동 에이전트의 특징을 표현하고 있다. 이동 에이전트는 각각의 데이타베이스 서버에서 정보를 검색한 후, 검색된 정보를 사용자 대신 자율적으로 가공해서 사용자의 목적에 부합한 결과만을 만들어 낸다. 이렇게 크기가 작아진 결과 값은 결국 네트워크 상에서 두 호스트간의 통신량을 감소시킨다. 따라서 그림 7에서 σ가 증가함에 따라 이동 에이전트 패러다임의 작업 수행 시간이 감소하는 것을 볼

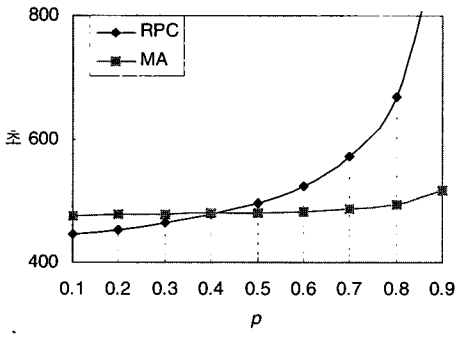

그림 5 σ = 0.1일 때 확률 p에 따른 실행 시간의 변화

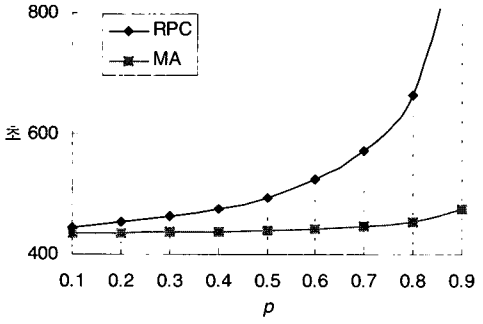

그림 6 σ = 0.9일 때 확률 p에 따른 실행 시간의 변화

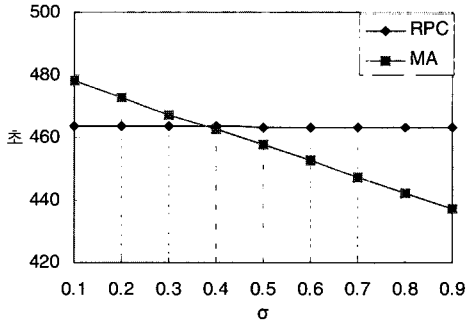

그림 7 p = 0.3일 때 σ에 따른 실행시간의 변화

수 있다. 하지만 RPC 패러다임인 경우 자율성이라는 특징이 없기 때문에 α값의 영향을 전혀 받지 않는 것을 그림 7에서 확인할 수 있다.

이동 에이전트 패러다임에서는 이동 에이전트가 호스트 사이를 이동할 때에만 통신이 이루어지고 각 호스트에서 처리해야 할 작업은 자율적으로 이동 에이전트가 처리하는 반면에 RPC 패러다임에서는 클라이언트가 요청한 작업을 수행하기 위하여 동일한 호스트와도 여러 번 통신을 해야만 한다. 그리고 분산 환경에서 안전한 통신을 위하여 보안 서비스를 적용하면 연산 비용이 큰 암호화 메커니즘을 사용해야한다. 그러므로 분산 환경의 안전한 통신 메커니즘으로 안전한 RPC 패러다임을 이용할 경우 고비용의 암호화 연산을 수반하는 통신 횟수와 통신량이 많아져서 실행 시간이 급격하게 증가하지만, 안전한 이동 에이전트 패러다임에서는 통신 횟수와 통신량의 증가가 크지 않으므로 실행 시간이 완만하게 증가함을 두 패러다임의 성능 분석을 통하여 살펴볼 수 있었다. 또한 이동 에이전트의 자율성이 실행 시간을 감소시키는 것도 살펴볼 수 있었다.

5. 결 론

본 논문에서는 분산된 데이타베이스 서버에서 정보를 검색하는 두 패러다임을 Petri Net을 이용하여 각각 모델링하였다. 고려된 보안 서비스는 인증 서비스, 비밀성 서비스, 무결성 서비스이다. 또한 두 패러다임의 차이를 보이기 위하여 동일한 데이타베이스 서버에서 연속적인 작업을 수행할 확률과 이동 에이전트가 데이타베이스 서버에서 자율적으로 작업을 수행하는 정도를 나타내는 에이전트의 선택도를 성능 모델에 적용하였다. 두 성능 모델의 수행 시간을 비교한 결과는 다음과 같다. 보안 서비스는 두 호스트 사이에 통신이 이루어질 때마다 수행되어야 하기 때문에, 동일한 데이타베이스 서버에서 연속적인 작업을 수행할 확률이 커짐에 따라 클라이언트와 데이타베이스 서버간의 통신을 지속적으로 유지해야하는 RPC 패러다임에서는 작업 수행 시간이 동일 데이타베이스 서버에서 연속적으로 작업을 수행할 확률에 비례하여 급격하게 증가한다. 반면 이동 에이전트는 방문한 데이타베이스 서버에서 자율적으로 작업을 수행하므로 사용자 호스트와 통신을 유지할 필요가 없다. 따라서 보안 서비스가 RPC 패러다임에 비하여 적게 적용되기 때문에 작업 수행 시간은 동일 데이타베이스 서버를 이용할 확률에 많은 영향을 받지 않는다. 에이전트 선택도는 단지 이동 에이전트 패러다임에만 영향을 미치는 요소이다. 에이전트의 선택도 값이 커지는 것은 정보 검색 결과를 이동 에이전트가 자율적으로 재처리하여 결과값의 크기를 감소시키는 것을 의미하기 때문에 에이전트의 선택도 값에 비례하여 이동 에이전트 데이타 크기가 작아진다.

실제로 이동 에이전트가 분산 환경에서 사용되기 위해서는 위에서 언급한 보안 서비스이외에도 더 많은 서비스가 제공되어야 한다. 특히 호스트가 이동 에이전트를 공격하는 것을 예방하기 위해서 에이전트를 보호하는 일은 아직까지도 해결하기 힘든 문제로 남아있다. 본 논문에서는 아직 까지 해결하지 못한 보안 문제를 포함시키지는 않았다.

향후 연구 과제로는 여러 IPC 패러다임의 더욱 정확한 성능 평가를 위하여 실제 분산 환경을 더욱 잘 표현할 수 있는 모델을 개발하고, 그 모델에 적용할 파라미터에 대한 연구가 필요하다. 보안 서비스를 적용할 경우 각 패러다임에 동일한 정도의 보안 서비스를 제공하기 위해서 각 패러다임의 보안 문제점과 문제를 해결할 수 있는 보안 서비스에 대한 연구가 더욱 필요하다.

참 고 문 헌

[1] A.D. Birrell and B.J. Nelson, "Implementing Remote Procedure Calls," ACM Transactions on Computer Systems 2(1), pp.39-59, February 1984.

[2] R. Srinivasan, "RPC: Remote Procedure Call Protocol Specification Version 2," RFC-1831, August 1995.

[3] J.W. Stamos and D.K. Gifford, "Remote Evaluation," ACM Transactions on Programming Languages and Systems, Vol.12, No.4, pp.537-565, October 1990.

[4] J. Baumann, F. Hohl, N. Radouniklis, K. Rothermel, and M. Strasser, "Communication Concepts for Mobile Agents," Proceedings of the 1 st International Workshop on Mobile Agents, Berlin (D), Lecture Notes in Computer Science, No. 1219, Springer-Verlag (D), pp. 123-135, April 1997.

[5] R. Gray, D. Kotz, S. Nog, D. Rus, and G. Cybenko, "Mobile agents for mobile computing," Technical Report PCS-TR96-285, Department of computer Science, Dartmouth College, Hanover, 1996.

[6] C.G. Harrison, D.M. Chess, and A. Kershenbaum, "Mobile Agents: Are they a good idea?," Research Report, IBM Research Division T.J. Watson Research Center, March 1995.

[7] J. White, "Mobile Agents White Paper," General Magic, Sunnyvale, CA, USA, 1998, URL

http://www.generalmagic.com/technology/techwhitepaper.hmtl.
[8] T-H Chia and S. Kannapan, "Strategically Mobile Agents," Proceedings of First International Workshop on Mobile Agents97, pp.149-161, April 1997.
[9] L. Ismail and D. Hagimont, "A Performance Evaluation of the Mobile Agent Paradigm," Proceedings of the 1999 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, pp.306-313, November 1999.
[10] A. Puliafto, S. Riccobene, and M. Scarpa, "An analytical comparison of the client-server, remote evaluation and mobile agents paradigms," Proceedings of First International Symposium on Agents Systems and Applications, pp.248-292, October 1999.
[11] M. Straßer and M. Schwehm, "A Performance Model for Mobile Agent Systems," Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA97), Vol.II, CSREA, pp.1132-1140, 1997.
[12] W. Jansen and T. Karygiannis, "Mobile Agent Security," NIST Special Publication 800-19, October 1999.
[13] W. M. Farmer, J. D. Guttman, and V. Swarup, "Security for mobile agents: Issues and requirements," In National Information Systems Security Conference. National Institute of Standards and Technology, October 1996.
[14] J. Barkley et al, "Security in Open Systems," NIST Special Publication 800-7, July 1994.
[15] A.D. Birrell, "Secure Communication Using Remote Procedure Calls," ACM Transactions on Computer Systems 3(1), pp.1-14, February 1985.
[16] A. Chiu, "Authentication Mechanism for ONC RPC," RFC-2695, September 1999.
[17] C.F. Tschudin, "Mobile agent security," In M. Klusch, editor, Intelligent Information Agents, Springer, July 1999.
[18] T. Sander and C.F. Tschudin, "Protecting Mobile Agents Against Malicious Hosts," in G. Vigna(Ed.), Mobile Agents and Security, Springer-Verlag, Lecture Notes in Computer Science No. 1419, 1998.
[19] B. Schneier, Applied Cryptography, Second Edition, pp.483-495, John Wiley & Sons, Inc., 1996.
[20] JCE1.2.1, http://java.sun.com/products/jce/index.html.
[21] ABA JCE, http://www.openjce.org.

한 승 완

1994년 2월 전남대학교 자연과학대학 전산학과(이학사). 1994년 3월 ~ 1996년 2월 전남대학교 자연과학대학 전산통계학과(이학석사). 1996년 3월 ~ 2001년 8월 전남대학교 자연과학대학 전산통계학과(이학박사). 2001년 12월 ~ 현재 한국전자통신연구원 선임연구원. 관심분야는 네트워크 보안, 암호화 이론, 계산 이론, 알고리즘

정 기 문

1999년 2월 전남대학교 전산학과 이학사 1999년 3월 ~ 2001년 8월 전남대학교 전산학과 이학석사. 2001년 7월 ~ 현재 한국정보보호진흥원 연구원. 관심분야는 정보보호, 취약점 분석평가, 해킹방법론

박 승 배

1989년 2월 전남대학교 자연과학대학 전산통계학과(이학사). 1990년 3월 ~ 1992년 2월 전남대학교 자연과학대학 전산통계학과(이학석사). 1992년 3월 ~ 1996년 8월 전남대학교 자연과학대학 전산통계학과(이학박사). 1996년 3월 ~ 1999년 9월 초당대학교 컴퓨터과학과 전임강사. 1999년 10월 ~ 현재 초당대학교 컴퓨터과학과 조교수. 관심분야는 보안, 암호, 그래프 이론

임 형 석

1983년 서울대학교 컴퓨터공학과 졸업(학사). 1985년 한국과학기술원 전산학과 졸업(석사). 1993년 한국과학기술원 전산학과(박사). 1996년 8월 ~ 1997년 7월 미국 퍼듀대학교 전산학과 방문교수. 현재 전남대학교 전산학과 교수. 관심분야는 계산이론, 알고리즘, 병렬 및 분산처리, 암호이론임