다중 분산 웹 클러스터모델의 안전한 데이터 전송을 위한 상호 인증 프로토콜

이 기 준† · 김 창 원†† · 정 채 영†††

요 약

기존의 클러스터 시스템을 확장한 다중 분산 웹 클러스터 모델은 개방 네트워크상에 존재하는 다수의 시스템 노드들을 단일한 가상 네트워크로 구축하여 사용자로부터 요구되어지는 대규모 작업을 병렬 컴퓨팅 방식으로 처리하는 클러스터 시스템이다 구성된 특성상 다중 분산 웹 클러스터 모델은 불법적인 3자에 의해 내부 시스템 노드들이 노출되어 있으며, 각 시스템 노드간의 협조작업 시 고의적인 방해와 공격으로 정상적인 작업수행이 불가능할 가능성을 지니고 있다. 본 논문에서는 시스템 노드의 서비스 코드 블록의 등록, 요구, 협조 및 결과취합 시 해당 시스템노드의 인증을 위하여 키 분배방식을 통한 시스템 노드 상호 인증 프로토콜을 제시하며, 전체 시스템 노드의 대칭키를 안전하고 효율적으로 관리하며 분배하는 SNKDC를 설계한다. SNKDC는 시스템 노드가 작업수행 시 필요한 대칭키를 분배하며, 제공된 키를 기반으로 시스템 노드는 암호화된 패킷을 전송한다. 시스템 노드간의 주고받는 암호화 패킷은 3자에 의해 해독되거나 거짓 메시지를 통한 정보의 유출을 방지할 수 있다.

Mutual Authentication Protocol for Safe Data Transmission of Multi-distributed Web Cluster Model

Kee-Jun Lee† · Chang-Won Kim†† · Chai-Yeoung Jung†††

ABSTRACT

Multi-distributed web cluster model expanding conventional cluster system is the cluster system which processes large-scaled work demanded from users with parallel computing method by building a number of system nodes on open network into a single imaginary network. Multi-distributed web cluster model on the structured characteristics exposes internal system nodes by an illegal third party and has a potential that normal job performance is impossible by the intentional prevention and attack in cooperative work among system nodes. This paper presents the mutual authentication protocol of system nodes through key division method for the authentication of system nodes concerned in the registration, requirement and cooperation of service code block of system nodes and collecting the results and then designs SNKDC which controls and divides symmetrical keys of the whole system nodes safely and effectively. SNKDC divides symmetrical keys required for performing the work of system nodes and the system nodes transmit encoded packet based on the key provided. Encryption packet given and taken between system nodes is decoded by a third party or can prevent the outflow of information through false message.

키워드 : 클러스터 시스템(Cluster System), 분산 컴퓨팅(Distributed Computing), 키분배(key division method), 인증시스템(Authentication System)

1. 서 론

최근 하드웨어 기술의 성장과 네트워크 기술의 비약적인 발전은 고속의 클러스터 컴퓨팅 모델을 구축할 수 있는 기반을 제공하여 주었다. 클러스터 모델은 매우 저렴한 비용으로 주어진 작업을 병렬처리 할 수 있다는 장점을 지니고 있으며 [1], 구성방식과 적용분야에 따라 HPC (high performance cluster)[2], Bulk Storage 클러스터, Web/Internet 클러스터 [3], HA(high availability) 클러스터로 구분 지을 수 있다[4].

본 논문에서는 기존의 클러스터 시스템들을 기반으로 한 다중 분산 웹 클러스터 모델을 구성하고 구성된 시스템 노드들간의 안전한 자료전송을 위한 시스템 노드 상호 인증 프로토콜을 제안하고자 한다. 기존의 클러스터 모델은 고속의 지역 네트워크를 기반으로 일정 수준 이상의 시스템으로 구성되는데 반하여 다중 분산 웹 클러스터 모델은 개방화된 웹 상에 존재하는 저가(low price), 저속(low speed)의 다양한 시스템 노드를 대상으로 한다. 따라서 제안 모델에서는 복수개의 시스템 노드들을 단일한 가상 네트워크에 묶어놓은 서브 클러스터 그룹을 구성하고, 구축된 서브 클러스터 그룹을 기반으로 다중 분산 웹 클러스터 모델을 구축한다. 구축된 다중 분산 웹 클러스터 모델은 사용자가 요구하는 대규모의

† 정 회 원 : 조선대학교 일반대학원 전산통계학과
†† 정 회 원 : 조선대학교 일반대학원 전산통계학과
††† 종신회원 : 조선대학교 자연과학대학 수학·전산통계학부 교수
논문접수 : 2001년 8월 8일, 심사완료 : 2001년 9월 27일

작업을 부하분배 및 병렬 컴퓨팅 방식을 이용하므로 처리의 효율을 극대화시킬 수 있다.

다중 분산 웹 클러스터 모델은 개방화된 네트워크 환경을 기반으로 구성되었기 때문에, 구조적 특성상 불법적인 3자에 의해 내부의 시스템 노드들이 노출되어 있으며, 각 시스템 노드간의 협조작업을 진행할때 고의적인 방해와 공격으로 정상적인 작업 수행을 불가능하게 할 가능성을 지니고 있다[5]. 따라서 이러한 불법적인 공격에 대하여 시스템 노드들을 보호하고 인증받지 못한 사용자로부터의 정보유출과 불법적인 서비스 요구를 효과적으로 대응할 수 있는 보안 시스템이 필요하다[6, 7].

시스템 노드의 상호인증 프로토콜은 시스템 노드의 서비스 코드 블록의 등록, 요구, 시스템 노드간의 협조작업, 작업결과의 취합시 해당 시스템 노드가 다중 분산 웹 클러스터 모델에서 인증받은 시스템 노드인가를 확인한다. 이를 위하여 다중 분산 웹 클러스터 모델은 전체 시스템 노드의 대칭키를 분배하기 위한 시스템 노드 키 분배 센터(system node key distribution center : SNKDC)를 구축한다. 구축한 SNKDC는 시스템 노드가 작업 수행시 필요한 대칭키를 안전하게 분배하여 제공된 대칭키를 기반으로 시스템 노드는 암호화된 패킷을 전송한다. 따라서 각 시스템 노드들간에 주고받는 암호화된 패킷들은 제 3자에 의해 해독되거나 거짓 메시지를 통한 정보의 유출을 방지할 수 있다[8].

본 논문에서는 전체 시스템 노드의 대칭키를 안전하게 관리하며 분배하는 SNKDC의 설계와 안전한 키 분배 방식을 통한 상호인증 프로토콜을 제안하며, 분산 침입 탐지 시스템과 시스템 노드 인증 프로토콜을 통하여 분산 클러스터환경의 시스템 노드들을 안전하게 운영할 수 있는 방안을 제시한다.

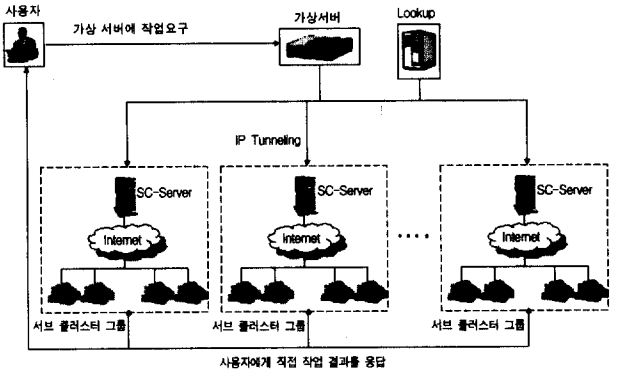
본 논문은 다음과 같이 구성되어 있다. 먼저 2장에서는 본 논문의 배경이 되는 다중 분산 웹 클러스터 모델에 대하여 기술하고 3장에서는 SNKDC의 설계와 이를 이용한 상호 인증 프로토콜에 대하여 기술한다. 그리고 4장의 실험의 고찰을 통하여 제안 프로토콜의 안정성을 검증하고 마지막 5장에서 결론은 맺는다.

2. 다중 분산 웹 클러스터 모델

기존의 클러스터 시스템은 고속의 지역 네트워크 기반으로 일정 수준이상의 시스템으로 구성되는데 반하여 다중 분산 웹 클러스터 모델은 웹 유휴자원을 이용한 저가, 저속의 시스템 노드를 대상으로 한다. 따라서 구성된 시스템 노드의 다양성으로 인하여 복수 개 시스템 노드들을 단일한 가상 네트워크에 묶어놓은 서브 클러스터 그룹으로 구성하고, 이중 한 시스템노드가 서브 클러스터 그룹을 대표하는 분산 서버(SC-Server)가 된다. 따라서 다중 분산 웹 클러스터 모델은 구축된 서브 클러스터 그룹들을 기반으로 형성되며 사용자가 요구하는 대규모의 작업을 부하분배 및 병렬 컴퓨팅 방식을 이용하므로 처리 효율의 극대화시킬 수 있다.

2.1 다중 분산 웹 클러스터 모델 구성

다중 분산 웹 클러스터 모델은 단일한 가상 네트워크에 묶여져 있는 서브 클러스터 그룹과 서브 클러스터 그룹의 집합체인 다중 분산 웹 클러스터 모델로 구분한다. 서브 클러스터 그룹은 일정 수량의 시스템 노드들을 동적(dynamic)으로 구성하고, 가상 서버로부터 전송되어온 사용자의 서비스 작업을 분산 처리할 수 있는 병렬 컴퓨팅 구조로 구성되어 있으며, 다중 분산 웹 클러스터 모델은 이러한 서브 클러스터 그룹들에 대한 작업의 지시와 수행결과의 통합을 수행한다.



(그림 1) 다중 분산 웹 클러스터 모델의 구성도

가상서버에 의해 묶여진 서브 클러스터 그룹은 네트워크상에 분산되어 있는 여러 시스템 노드들을 대표하는 SC-Server에 의해 하나의 노드로 그룹화 되어있고, 이를 외부에서 바라볼 때 서브 클러스터 그룹은 한 개의 노드로 구성된 단일 시스템으로 보여지게 된다. 따라서 가상서버와 SC-Server 간의 연결은 단일 네트워크 상에 묶여진 추상화된 내부 네트워크로, SC-Server와 시스템 노드간은 개방 네트워크상에 연결된 외부 네트워크로 인식하게 된다.

2.2 서브 클러스터 그룹의 구성

다중 분산 웹 클러스터 모델을 구성하는 시스템 노드들은 개방화된 네트워크상에서 분포되어있는 일정 수량의 시스템을 서브 클러스터 그룹으로 동적 구성하고 가상서버로부터 전송되어온 사용자의 서비스 요구를 분산 처리할 수 있는 병렬 컴퓨팅 구조로 구성한다.

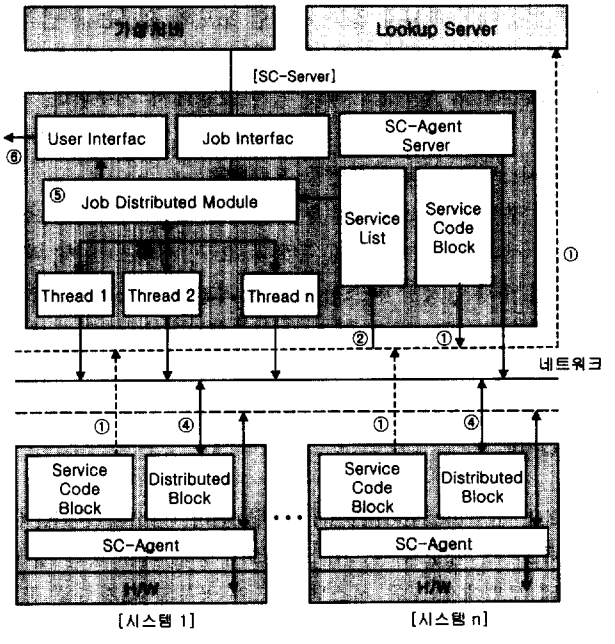
2.2.1. 서브 클러스터 그룹의 구성

가상서버의 서비스 수행요구에 의해 구성된 서브 클러스터 그룹은 내부의 노드중 한 개의 시스템 노드를 선출해 SC-Server의 임무를 부여한다. 따라서 SC-Server는 동적으로 구성되어지며, 다중 분산 클러스터 그룹을 구성하고 있는 전

체 노드는 모두 SC-Server가 될 수 있는 잠재적인 가능성을 지니고 있다.

(그림 2)는 구성된 서브 클러스터 그룹과 가상서버, 전체 시스템 노드의 서비스 모듈을 통합 관리하는 Lookup Server의 작업과정을 나타내고 있다. SC-Server와 각각의 시스템 노드에 구성되어있는 주요 모듈의 기능은 다음과 같다.

- Job Distributed Module : 요청한 작업 내용을 분석하고, 서비스 코드 목록의 수(서브 클러스터 노드의 수)에 비례한 쓰레드(thread)를 발생시켜, 서브 클러스터 그룹을 구성하고 있는 시스템 노드에서 요청된 작업을 분산 처리한다.
- Service List : Service List 모듈은 Job Distributed Module로부터 요청받은 서비스 코드 블록을 찾기 위하여 Lookup Server에 검색을 요청한다. Service List 모듈은 전송받은 서비스 코드 목록을 Job Distributed Module에 전달한다.
- Service Code Block : Service Code Block은 해당 시스템 노드에서 처리할 수 있는 서비스의 코드로 시스템 노드들은 Service Code Block을 Lookup Server에 등록함으로써 서브 클러스터 그룹에 참여할 수 있는 자격을 지니게 된다.
- Distributed Block : Distributed Block은 SC-Server에서 발생한 쓰레드에 의해 해당 시스템 노드에서 수행되는 서비스 코드 블록이다.
- SC-Agent : SC-Agent는 개방 네트워크상에 구성되어 있는 각 시스템 노드간의 분산 작업요청에 대한 상호인증작업과 불법적인 외부의 침입에 대하여 분산탐지 등의 보안 모듈로 만일 SC-Agent를 구성하고 있는 시스템 노드가 SC-Server로 선출된다면 이 SC-Agent는 SC-Agent Server의 역할을 수행한다.



(그림 2) 서브 클러스터 그룹

2.2.2 서브 클러스터 그룹의 작업과정

(그림 2)에서 구성된 서브 클러스터 그룹에 가상서버의 작업이 요청되었을 때 수행과정은 다음과 같다.

- 서브 클러스터 그룹을 구성하고 있는 각 시스템 노드들은 클러스터 그룹에 참여하기 위하여 자신의 서비스 코드 블록(service-code block)을 Lookup Server에 등록한다(①). Lookup Server에 등록되어진 코드블록은 Lookup Server의 Service Code Bank에 저장되어지며, 이후 SC-Server나 다른 시스템 노드의 작업 요청시 사용하게 된다.
- 가상 서버부터 해당 서브 클러스터 그룹에 서비스 작업이 요청되어지면 Job Interface 모듈은 서비스 요구 패킷을 분석하고 이중 작업 요청 영역을 Job Distributed Module에 전송한다. Job Distributed Module은 서비스 작업에 필요한 서비스 코드 목록을 전송받기 위하여 Service List 모듈을 통하여 Lookup Server로부터 서비스의 목록(시스템의 목록)을 전송 받는다(②).
- Lookup Server로부터 전송되어진 서비스 코드 목록은 Job Distributed Module로 전달되고, 수행해야할 작업의 내용이 서비스의 목록에 비례하여 분할되어진다. 분할되어진 각 작업의 내용은 각각 쓰레드에 의해 수행되고, 발생된 쓰레드는 서비스를 수행할 수 있는 시스템 노드의 Distributed Block과 원격 메소드 호출(remote method invocation)을 이용하여 서비스를 제공하는 시스템 노드에서 수행된다(③).
- 서브 클러스터 그룹을 구성하는 각 시스템 노드의 Distributed Block은 SC-Server에서 구동된 쓰레드로부터 처리할 데이터를 전송 받아 작업을 수행한다(④).
- 각 시스템 노드에서 수행되었던 작업의 결과는 메시지 전송방식에 의하여 SC-Server의 Job Distributed Module에 취합되어진다(⑤).
- SC-Server의 Job Distributed Module에 취합되어진 내용은 Job Interface 모듈에 의해 임시 저장한 사용자의 주소와 포트번호를 이용하여 작업을 요청한 사용자에게 직접 전송되어진다(⑥).

3. 시스템 노드의 상호인증 프로토콜

3.1 다중 분산 웹 클러스터 모델의 보안구조

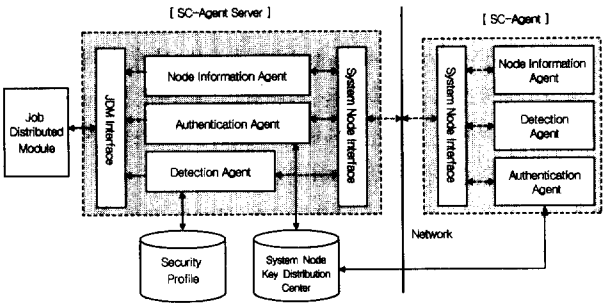
개방형 네트워크상의 시스템 노드들로 구성된 다중 분산 웹 클러스터 모델은 사용자에게 다양한 서비스와 고속의 연산기능을 제공하여주지만 구성된 특성상 불법적인 3자에 의해 내부 시스템 노드들이 노출될 가능성이 존재하며 또한 각 시스템 노드간의 협조 작업을 진행할 때 고의적인 방해와 공격으로 정상적인 서비스 수행을 불가능하게 할 수 있는 가능성을 지니고 있다. 따라서 다중 분산 웹 클러스터 모델은 이러한 불법적인 공격으로부터 시스템 노드들을 보호하고 인증받지 못한 사용자로부터의 정보의 유출과 불법적인 서비

스 요구에 효과적으로 대처할 수 있는 보안관리 시스템이 필요하다.

시스템 노드간의 상호인증 프로토콜은 서비스 코드 블록의 등록과 요구, 시스템 노드간의 협조작업, 작업 결과의 취합시 해당 시스템 노드가 다중 분산 웹 클러스터 모델에서 인증 받은 시스템 노드인가를 확인한다.

3.1.1 시스템 보안을 위한 SC-Agent 구조

SC-Agent는 서브 클러스터 그룹을 구성하는 시스템 노드의 Agent 모듈로 시스템 노드간의 상호인증과 불법적인 침입탐지를 위한 Agent로 구성되어 있다.



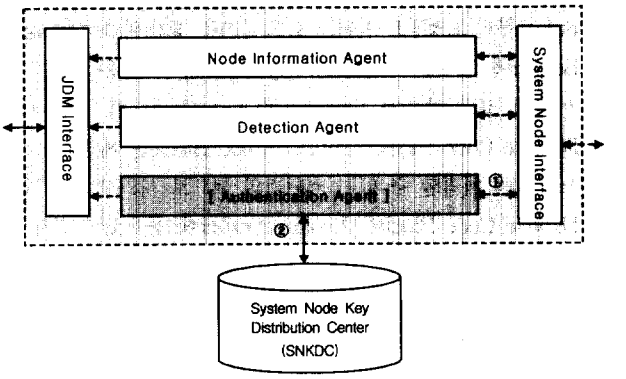
(그림 3) SC-Agent의 구조

(그림 3)에서 SC-Agent Server의 역할은 서브 클러스터 그룹내의 시스템 노드중에서 SC-Server로 선출된 시스템 노드의 SC-Agent가 수행하게 된다. SC-Agent Server는 구성된 시스템 노드내의 SC-Agent와 유기적으로 시스템의 정보와 인증, 침입탐지에 관한 정보를 교환한다. SC-Agent 모듈은 Node Information Agent, Detection Agent, Authentication Agent로 구성되며 이외에 Job Distributed Module과의 자료 전달을 위한 JDM Interface와 System Node Interface로 구성되어진다. 구성된 세 개의 Agent는 SC-Agent 모듈내에서 각각 독립적으로 운영되며, 이중 Detection Agent와 Authentication Agent는 다음과 같은 역할을 수행한다.

- Detection Agent(DA) : Detection Agent는 불법적인 시스템 접근을 탐지하기 위하여 다른 시스템 노드들로부터 요구되어진 작업패킷 검사모듈과 Job Distributed Module 내의 공유 메모리를 기반으로 각 시스템 노드에 있는 Detection Agent와 질의와 응답을 통해 침입을 탐지하는 모듈로 구성된다. 만일 시스템 노드상태에 대한 의심스러운 징후가 발생하였다면 구성된 전문가 영역인 보안 프로파일(security profile)에 따라 침입 여부를 결정한다.
- Authentication Agent(AA) : Authentication Agent는 구성된 서브 클러스터 그룹내의 각 시스템 노드들이 자신의 서비스 코드 블록의 등록이나 요구, 작업의 요청, 수행된 결과를 전송하는 등의 역할을 수행할 때 해당 시스템 노드가 서브 클러스터 그룹내의 정상적인 사용자인가를 인증하는 인증 에이전트이다. Authentication Agent는 인증을 위하여 각 시스템 노드의 공개키를 System Node Key Distribution Center(SNKDC)에 등록하고 이를 이용한 인증 프로토콜을 수행한다.

3.1.2 SC-Agent내의 Authentication Agent의 구조

SC-Agent 모듈내의 Authentication Agent는 서비스 코드 블록의 등록과 요구, 작업수행 등에 관한 작업패킷을 전달받을 때 상대방 시스템 노드가 다중 분산 웹 클러스터 모델의 적법한 시스템 노드인가를 인증하는 에이전트로 SNKDC를 통하여 시스템 노드의 인증 작업을 수행한다.



(그림 4) Authentication Agent의 동작방식

(그림 4)에서 Authentication Agent는 서비스 요구를 수행하는 시스템노드의 요구패킷이 들어왔을 경우 (①) 이에 대한 인증을 수행하기 위하여 System Node Key Distribution Center에 상대방 시스템 노드의 인증여부를 문의하고 이에 대한 인증프로토콜에 의해서 시스템 노드를 인증하게 된다(②). 만일 인증이 수락되지 않는 시스템 노드의 요구는 불법적인 시스템 침입으로 간주하여 이후 해당 시스템 노드의 요구 패킷은 Detection Agent의 패킷 검사 모듈에서 거부된다.

3.2 시스템 노드의 상호인증 프로토콜

다중 분산 클러스터 그룹을 구성하고 있는 각각의 시스템 노드는 서비스 코드 블록의 등록과 요구, 서비스 요청, 서비스 작업 수행시 해당 시스템 노드가 적법한 시스템 노드인지를 판별할 수 있는 상호 인증 작업이 필요하다. 다중 분산 클러스터 그룹에서는 시스템 노드간의 상호 인증을 위하여 Kerberos의 인증 시스템을 기반으로 한 상호인증 프로토콜을 사용한다.[9]

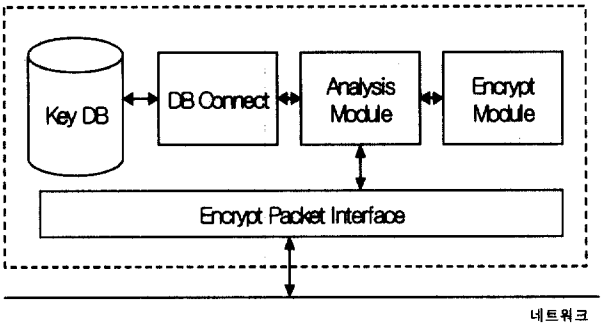
3.2.1 System Node Key Distribution Center(SNKDC)

SNKDC는 다수의 시스템 노드로 구성된 다중 분산 웹 클러스터 모델에서 모든 시스템 노드와 공통된 대칭키를 공유하는 신뢰할 수 있는 키 분배 서버이다. 임의의 두 시스템 노드간에 특정한 암호기법에 소요되는 키가 시스템 노드 키 분

배 센터를 통해서 해당 시스템 노드에 분배되고, 시스템 노드는 분배받은 키를 통해서 암호화 기법을 수행하게 된다. 시스템 노드의 입장에서는 클러스터 그룹내의 모든 시스템 노드들의 대칭키를 저장할 필요가 없고, 서비스 코드 블록의 등록, 요구, 수행, 작업요청시 SNKDC와의 공통된 대칭키를 이용하기만 하면 된다.

3.2.2.1 SNKDC의 구성

SNKDC는 키-암호화 키의 역할을 수행하는 대칭키를 기반으로 임의의 시스템 노드간의 상호 인증과 작업연계를 위한 키를 분배하여 준다.

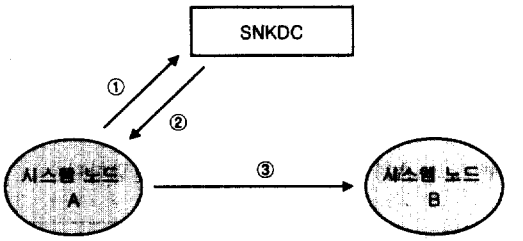


(그림 5) SNKDC의 구성

- Encrypt Packet Interface : 시스템 노드가 작업 대상 시스템 노드의 대칭키 요구를 위하여 암호화된 패킷을 작성할 때 SNKDC는 암호화된 패킷을 수신하기 위하여 Encrypt packet Interface 모듈을 이용한다. 전달받은 암호화된 패킷은 Analysis Module에 전달되어 인증과정을 거친 후 요구된 시스템 노드의 대칭키를 전송한다. 송수신시 사용되는 암호화 패킷은 인증 프로토콜에 의해 정의된 구조의 패킷을 사용한다.
- Analysis Module : Analysis Module은 Encrypt Packet Interface로부터 전달되어온 암호화 패킷을 분석하고 작업을 요청해온 시스템 노드의 인증 여부를 확인한다. 이때 사용되는 암호화 기법은 Encrypt Module을 이용한다. 시스템 노드의 인증이 확인되면 DB Connect를 이용하여 해당 시스템 노드의 요구를 수행한다. 시스템 노드에 전송되는 암호화 패킷 역시 인증 프로토콜에 의한 암호화작업에 의해 패킷화되어 전송된다.
- Encrypt Module : Encrypt Module은 암호화 패킷의 해독 및 작성을 위하여 사용되는 암호화 기법을 모듈화하였다. Encrypt Module내의 암호화 기법으로는 RSA, DES, El-Gamal, 타원곡선 암호, MD5가 있다.
- Key DB : Key DB는 다중 분산 웹 클러스터 모델을 구성하고 있는 전체 시스템 노드의 대칭키를 저장하고, 저장된 대칭키는 인증을 거친 시스템 노드에게 제공되어지며, 새로이 등록되는 시스템 노드의 대칭키를 저장한다.

3.2.2.2 키 분배 방식

(그림 6)은 시스템 노드A와 시스템 노드B가 SNKDC를 이용하여 상호간의 공통된 키를 공유하는 기본적인 과정을 보여주고 있다.



(그림 6) SNKDC를 이용한 키 분배

(그림 6)에서 시스템 노드 A는 시스템 노드 B와의 작업연계를 위하여 시스템 노드 B의 대칭키를 필요로 한다. 따라서 SNKDC에게 시스템 노드 B의 대칭키를 요구하고 (①), SNKDC는 시스템노드의 대칭키 요구를 수행하기 전에 먼저 시스템 노드 A의 인증여부를 확인한다. 만일 시스템 노드 A의 인증이 확인되지 않으면 요구된 작업의 요청을 수행되지 않고 인증이 확인되면 자신의 Key DB에서 시스템 노드 B의 대칭키를 검색한 후 이를 암호패킷화하여 전송한다(②). 시스템 노드 A는 SNKDC로부터 전송된 시스템 노드 B의 대칭키를 이용하여 작업을 요구한다(③).

3.2.2 상호인증 프로토콜

다중 분산 웹 클러스터 모델을 구성하고 있는 시스템노드들은 상호간의 인증여부를 확인하기 위하여 시스템 노드 상호인증 프로토콜을 사용한다.

〈표 1〉 시스템 노드 상호인증 프로토콜의 수식모음

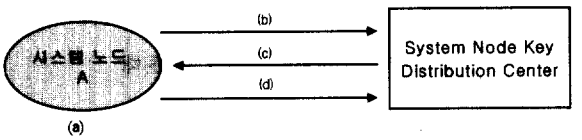
Key_A : 시스템 노드 A의 대칭키
$Public_{SNKDC}$: SNKDC의 공개키
$Public_{SNKDC}(Key_A)$: Key_A 를 $Public_{SNKDC}$ 로 암호화
$MD(Key_A)$: Key_A의 메시지 축약
$Private_{SNKDC}$: SNKDC의 개인키
$Private_{SNKDC}(Public_{SNKDC}(Key_A))$: SNKDC의 개인키로 해독
$Key_{Lookup\ Server}$: Lookup Server의 대칭키
$Time\ Stamp_A$: 시스템 노드 A의 Time Stamp
$Sesion\ Key_{Lookup\ Server}$: Lookup Server의 세션키
JOB : 작업내용
Key_B : 시스템 노드 B의 대칭키
$RESULT$: 작업결과

3.2.2.1 서비스 등록시 시스템 노드 인증

시스템 노드는 자신의 서비스 코드 블록을 Lookup Server에 등록하기 위하여 SNKDC를 이용한 인증과정을 수행한다. SNKDC는 서비스 코드 블록 등록을 요청하는 시스템 노

드의 등록여부를 확인하고 만일 등록되어 있지 않는 시스템 노드라면 새로이 Key DB에 저장하고 만일 갱신된 시스템 노드라면 수정된 대칭키 정보를 Key DB에 저장한다.

① 서비스 코드 블록의 등록을 요청하는 시스템 노드 A는 (그림 7)과 같은 절차를 수행한다. 먼저 시스템 노드 A는 자신의 대칭키를 생성하고, 시스템 노드 키 분배 센터(SNKDC)의 공개키를 요구한다. 전달받은 SNKDC의 공개키를 이용하여 시스템 노드 A의 대칭키, 대칭키의 축약메세지, 시스템 노드 A의 IP로 구성된 패킷을 전달한다.



(그림 7) SNKDC의 공개키 분배 및 시스템 노드 패킷전달

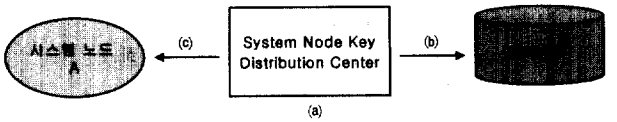
(a) 시스템 노드 A는 자신의 대칭키를 생성.
$Make\ \ Key_A$

(b) 시스템 노드 A는 SNKDC의 공개키를 요구.
$Request\ \ Public_{SNKDC}$

(c) SNKDC는 시스템 노드 A에 공개키를 전달.
$Reply\ \ Public_{SNKDC}$

(d) 시스템 노드 A는 SNKDC의 공개키를 이용하여 시스템 노드 A의 대칭키, 대칭키의 축약메시지, 시스템 노드 IP를 패킷화하여 전송.
$Send\ \{\ Public_{SNKDC}(Key_A),\ MD(Key_A),\ System\ IP\ \}$

② SNKDC는 시스템 노드 A로부터 전달받은 패킷을 분석하기 위하여 암호화된 대칭키를 해독하여 시스템 노드 A의 대칭키를 얻는다. 이후 암호화된 대칭키의 축약메세지와 전달된 축약메세지를 비교하여 시스템 노드 A의 전송여부를 확인한다. 만일 시스템 노드 A가 신규등록인 경우 Key DB에 시스템 노드 A의 주소(IP)와 대칭키를 저장하고 Lookup Server의 대칭키를 전달한다.



(그림 8) 시스템 노드 A의 인증 및 대칭키 전달

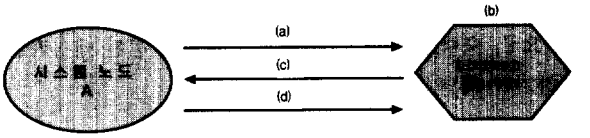
(a) - ① 암호화된 대칭키를 해독후 시스템 노드 A의 대칭키획득
$Private_{SNKDC}(Public_{SNKDC}(Key_A))$

(a) - ② 대칭키 축약값과 해독후 얻은 대칭키의 축약값 비교
$Compare\ \ MD(Key_A)\ ,\ MD(Private_{SNKDC}(Public_{SNKDC}(Key_A)))$

(b) 신규 등록된 시스템 노드인 경우 Key DB에 저장
$Store\ \ System\ Node\ IP\ \ ,\ Key_A$

(c) 시스템 노드 A에 Lookup Server의 대칭키를 전달
$Send\ \ Key_A(Key_{Lookup\ Server})$

③ 시스템 노드 A는 Lookup Server의 대칭키를 이용하여 Time Stamp를 암호화하여 Lookup Server에 전달한다. Lookup Server는 암호화된 Time Stamp를 대칭키로 확인한 후 시스템 노드와의 세션키를 전달하고, 시스템 노드 A는 Lookup Server와의 세션키를 통해 서비스 코드 블록을 등록한다.



(그림 9) 서비스 코드 블록의 등록

(a) time Stamp를 Lookup Server의 대칭키를 이용하여 암호화한 후 Lookup Server에 전송
$Send\ \{\ Key_{Lookup\ Server}(Time\ Stamp_A)\ ,\ Time\ Stamp_A\ ,\ Key_A\ \}$

(b) 암호화된 Time Stamp 해독후 시스템 노드 A의 Time Stamp와 비교
$Compare\ Key_{Lookup\ Server}(Key_{Lookup\ Server}(Time\ Stamp_A)),\ Time\ Stamp_A$

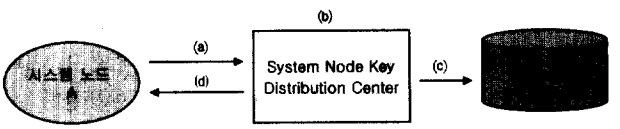
(c) 노드 A의 대칭키로 암호화한 Lookup Server의 세션키를 전달
$Reply\ \ Key_A(\ Sesion\ Key_{Lookup\ Server}\ )$

(d) 노드 A는 전달받은 세션키를 이용하여 서비스 코드 블록등록
$Send\ \{\ Key_A(Key_A(\ Sesion\ Key_{Lookup\ Server})),\ Service\ Code\ Block_A\ \}$

3.2.2.2 서비스 요구 시 시스템 노드 인증

만일 시스템 노드 A가 시스템 노드 B에게 작업을 요청하고 한다면 먼저 시스템 노드 A는 시스템 노드 B의 대칭키를 SNKDC에게 요구한다. SNKDC는 시스템 노드 A의 인증여부를 확인한 후 시스테 노드 B의 대칭키를 전송하고 시스템 노드 A는 시스템 노드 B의 대칭키를 이용한 작업요구 패킷을 전송하게 된다.

① 시스템 노드 A는 SNKDC에 시스템 노드 B의 대칭키를 요구하기 위하여. 시스템 노드 A의 대칭키, 대칭키의 축약메세지, 시스템 노드 A의 IP를 SNKDC의 공개키로 암호화한 패킷을 전달한다. SNKDC는 전달된 패킷을 분석하여 시스템 노드 A 인지를 확인한 후 시스템 노드 B의 대칭키 암호문을 전송한다.



(그림 10) 시스템 노드A의 요구패킷전달

(a) 시스템 노드 A는 SNKDC의 공개키를 이용하여 시스템 노드 A의 대칭키, 축약메세지, 시스템 노드 A의 IP, 시스템 노드 B의 IP로 구성된 대칭키 요구 패킷을 전달
$Send\ \{\ Public_{SNKDC}(Key_A),\ MD(Key_A),\ System\ IP_A,\ System\ IP_B\ \}$

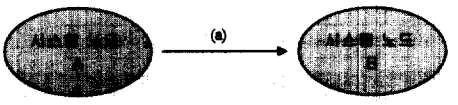
(b)-① 암호화된 대칭키를 해독하여 시스템 노드 A의 대칭키를 얻음
$Private_{SNKDC}(Public_{SNKDC}(Key_A))$

(b)-② 대칭키 축약값과 해독후 얻은 대칭키의 메시지 축약값을 비교
$Compare\ \ MD(Key_A),\ MD(Private_{SNKDC}(Public_{SNKDC}(Key_A)))$

(c) Key DB에 저장된 에 저장 시스템 노드 A의 정보와 비교
$Compare\ \{\ System\ Node\ IP\ ,\ Key_A\},\ \{\ System\ Node\ IP_{KeyDB},\ Key_{A\ KeyDB}\}$

(d) 시스템 노드 A의 인증이 확인되면 시스템 노드 B의 대칭키를 암호화하여 전달
$Reply\ \ Key_A(Key_B)$

② 시스템 노드 A는 SNKDC로부터 전달받은 시스템 노드 B의 대칭키를 이용하여 작업내용, Time Stamp, 시스템 노드 A의 대칭키로 구성된 작업요구 패킷을 작성하여 전송한다.

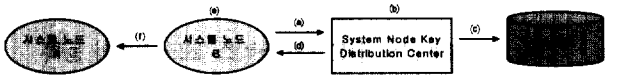


(그림 11) 작업요구패킷 전달

(a) 시스템 노드 A는 시스템 노드 B의 대칭키를 이용하여 작업내용, Time Stamp, 시스템 노드 A의 대칭키를 암호화하여 구성된 패킷을 시스템 노드 B에 전송

$Send\ \{\ Key_B(JOB),\ Key_B(Time\ Stamp_A),\ Key_B(Key_A)\ \}$

③ 시스템 노드 B는 SNKDC의 공개키를 입수한 후 전달받은 패킷을 분석하여 시스템 노드 A의 인증여부를 SNKDC에 요구한다. 시스템 노드 A의 인증이 확인되면 암호화된 작업내용을 해독한 후 작업을 시작한다.



(그림 12) 시스템 노드 A의 인증확인 및 작업수행

(a) 시스템 노드 B는 SNKDC의 공개키를 이용하여 시스템 노드 B의 대칭키, 축약메세지, 시스템 노드 B의 IP, 시스템 노드 A의 IP로 구성된 대칭키 요구 패킷을 전달

$Send\ \{\ Public_{SNKDC}(Key_B),\ MD(Key_B),\ System\,IP_B,\ System\,IP_A\ \}$

(b) - ① 암호화된 대칭키를 해독하여 시스템 노드 B의 대칭키를 얻음

$Private_{SNKDC}(Public_{SNKDC}(Key_B))$

(b) - ② 대칭키 축약값과 해독하여 얻은 대칭키의 축약값을 비교

$Compare\ \ MD(Key_B),\ MD(Private_{SNKDC}(Public_{SNKDC}(Key_B)))$

(c) Key DB에 저장된 애 저장 시스템 노드 A의 정보와 비교

$Compare\ \{\ System\,Node\,IP_B,\ Key_B\},\{\ System\,Node\,IP_{KeyDB},\ Key_{B\,KeyDB}\}$

(d) 시스템 노드 A의 인증이 확인되면 시스템 노드 B의 대칭키로 암호화하여 전달

$Reply\ \ Key_B(Key_A)$

(e) 노드 B는 노드 A의 대칭키를 비교하여 A의 인증 확인

$Compare\ \ Key_B(Key_A),\ \ Key_B(Key_A),$

(f) 시스템 노드 B는 암호화된 작업문을 해독하여 실행하고 Time Stamp를 시스템 A에 전송하여 작업시작여부를 알려준다.

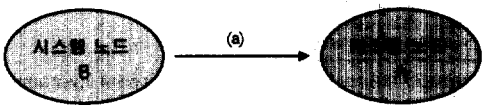
$Run\ \ Key_B(Key_B(JOB))$

$Reply\ \ Key_A(Key_B(Key_B(Time\ Stamp_A)))$

3.2.2.3 작업 취합시 시스템 노드 인증

작업수행을 요청받은 시스템 노드는 작업결과를 요청한 시스템 노드 또는 SC-Server의 공유 메모리로 전송하여야 한다. 만일 작업결과를 전송받는 시스템 노드 A에서는 작업을 전송하는 시스템 노드 B의 인증여부를 확인하여야 하고 전달된 작업의 내용의 수렴여부를 전달하여 주어야 한다.

① 시스템 노드 B는 시스템 노드 A의 대칭키를 이용하여 작업결과, Time Stamp, 시스템 노드 B의 대칭키로 구성된 작업결과 패킷을 작성하여 전송한다.

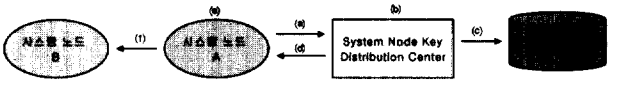


(그림 13) 작업결과 패킷 전달

(a) 시스템 노드 B는 시스템 노드 A의 대칭키를 이용하여 작업결과, Time Stamp, 시스템 노드 B의 대칭키를 암호화하여 구성된 패킷을 시스템 노드 A에 전송

$Send\ \{\ Key_A(RESULT),\ Key_A(Time\ Stamp_B),\ Key_A(Key_B)\ \}$

② 시스템 노드 A는 SNKDC의 공개키를 입수한 후 전달받은 작업결과 패킷을 분석하여 시스템 노드 B의 인증여부를 SNKDC에 요구한다. 시스템 노드 B의 인증이 확인되면 암호화된 작업 결과 내용을 해독한 후 작업을 저장하고, 시스템 노드 B에 수렴여부를 알려준다.



(그림 14) 시스템 노드 B의 인증확인 및 작업결과 저장

(a) 시스템 노드 A는 SNKDC의 공개키를 이용하여 시스템 노드 A의 대칭키, 축약메세지, 시스템 노드 A의 IP, 시스템 노드 B의 IP로 구성된 대칭키 요구 패킷을 전달

$Send\ \{\ Public_{SNKDC}(Key_A),\ MD(Key_A),\ System\,IP_A,\ System\,IP_B\ \}$

(b) - ① SNKDC는 암호화된 대칭키를 해독하여 노드 A의 대칭키획득

$Private_{SNKDC}(Public_{SNKDC}(Key_A))$

(b) - ② 대칭키 축약값과 해독하여 얻은 대칭키의 메시지 축약값 비교

$Compare\ \ MD(Key_A),\ MD(Private_{SNKDC}(Public_{SNKDC}(Key_A)))$

(c) Key DB에 저장된 에 저장 시스템 노드 A의 정보와 비교

$Compare\ \{\ System\,Node\,IP_A,\ Key_A\},\ \{\ System\,Node\,IP_{KeyDB},\ Key_{A\,KeyDB}\}$

(d) 노드 B의 인증이 확인되면 노드 A의 대칭키로 암호화하여 전달

$Reply\ \ Key_A(Key_B)$

(e) 노드 A는 노드 B의 대칭키를 비교하여 B의 인증을 확인

$Compare\ \ Key_A(Key_B),\ \ Key_A(Key_B),$

(f) 노드 A는 암호화된 작업결과을 해독하여 저장하고 Time Stamp를 시스템 노드 B에 전송하여 작업결과수렴여부를 알려준다.

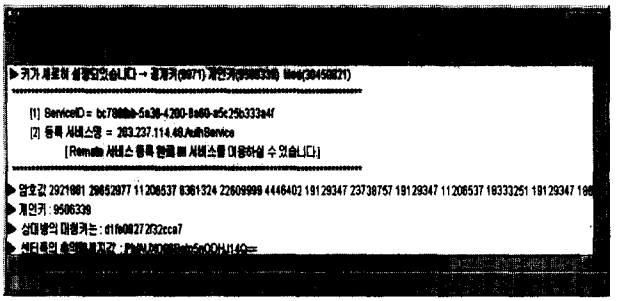
$Store\ \ Key_A(Key_A(JOB))$

$Reply\ \ Key_B(Key_A(Key_A(Time\ Stamp_B)))$

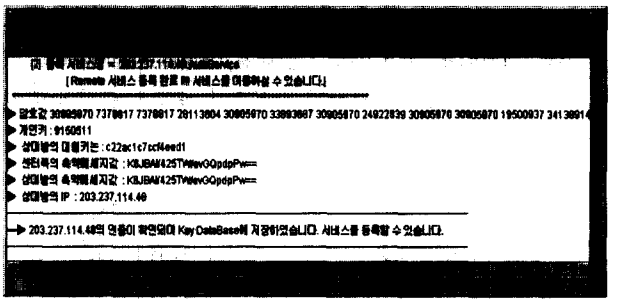
4. 실험 및 고찰

4.1 SNKDC

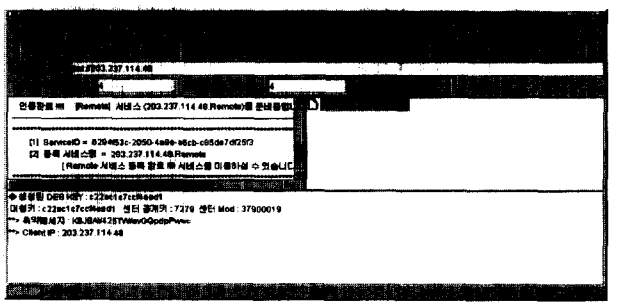
SNKDC는 다중 분산 클러스터 모델을 구성하고 있는 시스템 노드들의 공통된 대칭키를 공유하며 이를 분배해줄 수 있는 키 분배 서버이다. SNKDC는 임의의 시스템에서 동작하며 작업을 수행하는 시스템 노드들은 SNKDC를 통하여 해당 시스템 노드나 SC-Server, Lookup-Server의 접근이 가능하다. (그림 15)는 수행중인 SNKDC의 모습이다.



(그림 15) 수행중인 시스템 노드 키 분배 센터(SNKDC)



(a) 시스템 노드등록을 위한 SNKDC의 키 분배



(b) 시스템 노드의 인증과 서비스 코드 블록의 등록

(그림 16) SNKDC의 키분배와 서비스 코드 블록의 등록

4.2 수행 시나리오

시스템 노드의 인증작업을 수행하기 위하여 "203.237.114.48. Remote" 시스템 노드를 중심으로 서비스 코드의 등록과 작업의 요구, 작업결과의 수행과정을 시뮬레이션을 통하여 수행한다.

4.2.1 다중 분산 클러스터 그룹의 등록

시스템 노드가 다중 분산 클러스터 그룹에 등록하기 위해서는 자신의 서비스 코드 블록을 Lookup Server에 등록하여야 한다.

① "203.237.114.48.Remote"서비스 노드는 자신의 대칭키를 생성하고 SNKDC의 공개키를 요구한다.
② SNKDC는 자신의 공개키를 전달하고 "203.237.114.48.Remote" 서비스노드는 전달받은 공개키를 이용하여 자신의 대칭키, 축약메세지, 시스템 노드 IP를 패킷화하여 전송한다.
③ SNKDC는 전송된 패킷에서 "203.237.114.48.Remote"의 대칭키를 해독한 후 축약메세지와 비교하여 이상유무를 확인한다. "203.237.114.48.Remote"시스템 노드의 대칭키를 Key DB에 저장하고 Lookup Server의 대칭키를 전달한다.
④ "203.237.114.48.Remote"시스템노드는 Lookup Server의 대칭키를 이용하여 Time Stamp를 암호화한 후 Lookup Server에 전송하고, Lookup Server는 이를 해독하여 "203.237.114.48.Remote" 시스템 노드의 Time Stamp와 비교한다.
⑤ "203.237.114.48.Remote" 시스템 노드와 Lookup Server와의 상호인증이 확인되면 "203.237.114.48.Remote" 시스템 노드의 대칭키로 암호화한 Lookup Server의 세션키를 전달한다.
⑥ "203.237.114.48.Remote" 시스템 노드는 전달받은 세션키를 이용하여 자신의 서비스 코드블록을 Lookup Server에 등록한다.

(그림 16)의 (a)는 SNKDC에 의한 시스템 노드의 인증과 키 분배방식을 나타내고 있으며 (b)는 시스템 노드가 SNKDC의 인증 확인 후 자신의 서비스 코드블록을 Lookup Server에 등록하는 과정이다.

4.2.2 서비스 요구시 시스템 노드의 인증

다중 분산 클러스터 그룹의 부하 분산기에 의해 접수된 사용자의 작업이 해당 서브 클러스터 그룹에 전송되었을 때 SC-Server는 전달받은 작업을 분석하여 각 시스템 노드에 작업을 할당한다. 이때 SC-Server는 해당작업을 수행할 시스템 노드의 인증여부를 확인하여야 하며, 시스템 노드의 경우 SC-Server나 인증된 다른 시스템 노드로부터의 작업요구인지를 인증하여야 한다. 작성된 시나리오는 SC-Server가 "203.237.114.48. Remote" 시스템 노드에 작업을 요청할 때이다.

① SC-Server는 "203.237.114.48.Remote"시스템 노드에 작업을 요청하기 위해서 SNKDC에게 "203.237.114.48. Remote" 시스템 노드의 대칭키를 요구한다.
② SNKDC는 SC-Server의 인증을 확인한 후 "203.237.114.48.Remote"시스템 노드의 대칭키를 SC-Server의 대칭키로 암호화하여 전송한다.
③ SC-Server는 "203.237.114.48.Remote" 시스템 노드 대칭키를 이용하여 작업의 내용, Time Stamp, SC-Server의 대칭키를 암호화하여 전송한다.
④ "203.237.114.48.Remote" 시스템 노드는 전달된 작업패킷의 전달자를 확인하기 위하여 SNKDC에게 SC-Server의 인증을 요구한다.
⑤ SNKDC는 "203.237.114.48.Remote" 시스템 노드의 인증을 확인한 후 의뢰한 SC-Server의 인증을 확인하여 확인결과를 전달하여 준다.

⑥ "203.237.114.48.Remote"시스템 노드는 SC-Server의 인증이 확인되면 자신의 대칭키로 암호화된 작업요구사항을 해독한 후 SC-Server에게 작업시작 여부를 전달한다.

4.2.3 작업 결과 취합시 시스템 노드의 인증

시스템 노드에서 수행된 작업의 결과는 작업을 요청했던 시스템 노드에게 전송되어야 한다. 이때 작업 결과를 전송받는 시스템 노드의 경우에는 전송되는 자료에 대한 신뢰성을 확보하기 위하여 자료를 전송하는 시스템 노드에 대한 인증과정을 거쳐야 한다. 본 시나리오에서는 "203.237.114.48.Remote" 시스템 노드가 요청된 작업의 결과를 SC-Server에게 전송하는 과정이다.

① "203.237.114.48.Remote" 시스템 노드는 SC-Server로부터 요청된 작업을 완료한 후 작업결과와 시스템 노드의 주소를 SC-Server의 대칭키로 암호화하여 SC-Server에 전송한다.

② SC-Server는 SNKDC에게 자료를 전달한 "203.237.114.48.Remote" 시스템 노드의 인증을 요구하면 SNKDC는 SC-Server의 인증과정을 거친 후 "203.237.114.48.Remote" 시스템 노드에 대한 인증결과를 전달하여 준다.

③ "203.237.114.48.Remote" 시스템 노드의 인증이 확인되면 SC-Server는 자신의 대칭키로 암호화된 작업결과를 해독한 후 공유메모리 공간에 저장하고 작업 수렴여부를 "203.237.114.48.Remote" 시스템 노드에 알려준다.

5. 결 론

본 논문에서는 다중 분산 클러스터 그룹의 특성을 고려하여 다수의 시스템 노드들이 분산작업을 진행할 때 불법 침입자에 의한 고의적인 방해와 공격을 막기 위하여 시스템 노드 상호간의 인증방식을 적용하였다.

다중 분산 클러스터 모델을 구성하고 있는 시스템 노드는 서비스 코드 블록의 등록과 요청, 서비스 작업 수행시 해당 시스템 노드가 적법한 시스템 노드인지를 판별할 수 있도록 Kerberos의 인증시스템을 기반으로 한 상호인증 프로토콜을 수행한다 인증작업의 수행을 위하여 전체 시스템 노드의 대칭키를 공유하며, 신뢰할 수 있는 키 분배 서버인 SNKDC를 구축하였다. 임의의 두 시스템 노드간에 암호기법에 사용되는 키는 SNKDC를 통해 해당 시스템 노드에 분배되고, 시스템 노드는 분배받은 키를 통해 암호화 기법을 수행하게 된다. 시스템 노드의 입장에서는 전체 시스템 노드의 대칭키를 저장할 필요가 없고, 작업수행시 SNKDC와의 공통된 대칭키를 이용하기만 하면 된다. 또한 각 시스템 노드간에 전송되는 암호화 패킷은 불법적인 3자가 해독하거나 거짓 메시지의 작성을 통한 정보의 유출을 방지할 수 있었다.

향후 연구과제로 분산된 시스템 노드간의 협조작업시 다양한 질의와 응답을 효율적으로 수행하기 위한 프로토콜의 표준화의 연구와 함께 시스템 노드의 상호인증작업중 SNKDC의 장애로 인하여 전체 클러스터 모델이 중지되는 문제를 해결할 수 있는 고가용 인증방식에 대한 연구가 수행되어야 하리라 사료된다.

참 고 문 헌

[1] Rajkumar Buyya, "High Performance Cluster Computing : Architectures and Systems," Vol.1, 1999, Prentice Hall, New Jersey, USA.

[2] H, J. Siegel, S. Abraham, W. L. Bain, K. E. Batcher, T. L. Casavant, et al. "Report of the Purdue Workshop on Grand Challengers in Computer Architecture for the Support of High Performance Computing," Journal of Parallel and Distributed Computing 16 : 199-221, Nov. 1992.

[3] Thomas T. Kwan, Robert E. McGrath, and Daniel A, Reed, "NCSA's World Wild Web Server : Design and Performance," IEEE Computer, pp.68-74, November, 1995.

[4] Ralf S.Engelshall, "Load Balancing Your Web Site : Practical Approaches for Distributing HTTP Traffic," Web Techniques Magazine, Volume 3, Issue 5, http : //www.webtechniques.com, May, 1998.

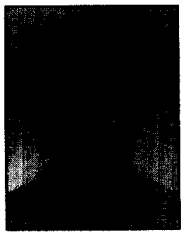
[5] Halsall, F. "Data Communications. Computer Networks and Open Systems," 4th Edition, Addison Wesley, 1996.

[6] R. Bird, et. al., "Systematic Design of Two party Authentication Protocols," Proc of Crypto'91, pp.44-61, 1992.

[7] W. Diffie, P.C.van Oorschot, and M.J,Wiener, "Authentication and Authenticated Key Exchanges," Designs, Codes, and Cryptography, Vol.2, pp.107-125, 1992.

[8] L.M.Kohnfelder, "Toward a Practical Public-Key Cryptosystem," B. Sc. Thesis, MIT Department of Electrical Engineering, 1978.

[9] RFC 1510, "The Kerberos Network Authentication Service(V5)," Internet Request for Comments 1510, Sep. 1993.



이 기 준

e-mail : cholee@shinbiro.com

1994년 조선대학교 전산통계학과(이학사)

1997년 조선대학교 일반대학원 전산통계학과 (이학석사)

1998년~현재 조선대학교 일반대학원 전산통계학과 박사과정

관심분야 : 신경망, 패턴인식, 인공지능, 분산 에이전트 시스템

김 창 원

e-mail : cwkim@dongkang.ac.kr

1981년 조선대학교 (공학사)

1985년 조선대학교 일반대학원 전산통계학과(이학석사)

1997년~현재 조선대학교 일반대학원 전산통계학과 박사과정

1987년~현재 동강대학 컴퓨터정보과 교수

1996년~현재 동강대학 전자계산소장

관심분야 : 인공지능, 신경망, 데이터베이스, 영상처리

정 채 영

e-mail : cyjung@mail.chosun.ac.kr

1983년 조선대학교 컴퓨터공학과(이학사)

1986년 조선대학교 일반대학원 전자과 전산전공(공학석사)

1989년 조선대학교 일반대학원 전기과 전산전공(공학박사)

1986년~현재 조선대학교 자연과학대학 수학·전산통계학부 부교수

관심분야 : 영상처리, 신경망, 데이터베이스, 멀티미디어 컨텐츠