STAS 시스템을 적용한 안전한 이동 에이전트 구조

김선영* · 조인준*

Architecture for Secure Mobile Agent through STAS System

Seon-Young Kim* · In-June Jo*

요 약

P2P 서비스를 이용하는 이동 단말의 증가 및 응용 분야 확산에 따라 이동 에이전트 기술이 P2P에 적용되어 다양한 분야에 혁신적인 서비스를 제공하고 있다. 그러나 이동 에이전트의 자유로운 이동성은 웜과 유사하게 동작하기 때문에 악의적인 공격자의 공격에 따라 빠르게 오염되는 문제가 제기되었지만 현재로서는 근본적인 해결책이 전무한 상황이다. 본 논문에서는 구조적인 P2P 환경에서 이동 에이전트의 안전성 검증을 제공하는 STAS(Security Tracking and Auditing Server) 시스템을 제안하였다. 이동 에이전트는 피어를 경유한 후 STAS에게 결과값을 전송함으로 보안감사 및 무결성을 검증받고, 이동 에이전트 생성자는 STAS로부터 최종 결과값을 획득한다. 이러한 방법을 통해서 이동 에이전트의 안전성 검증과 그 수행에 따라 발생할 이동 단말의 부하를 최소화할 수 있다.

ABSTRACT

As the mobile terminal which uses P2P service increases and it comes to be applied to many fields, mobile agent technology has been applied to P2P and its innovative services has been offered to various fields. However, free mobility of mobile agent technology works like worm, the problem which is contaminated by malicious attacker's attack quickly has appeared and fundamental solution has not been developed yet. This paper proposes STAS (Security Tracking and Auditing Server) system which can offer verification for security of mobile agent in structured P2P environments. Mobile Agent will send data value to STAS via peer so that STAS can verify secure audit and integrity and Mobile agent initiator will obtain the final value of the data from STAS. It can minimize overload of mobile terminal which is occurred by verification of mobile agent and its accomplishment.

키워드

STAS, mobile agent, security tracking, authentication

I. 서 론

인터넷 기술은 지속적으로 발전되고 있다. 특히 웹의 사용은 그 범위가 매우 다양하게 진화되고 있다. 이를 지원하는 인프라는 이동통신 기술, H/W 장비 기술, S/W 기술 등이다. 또한 웹에서 클라이언트/서버 구조는 모든 기능과 서비스를 중앙에서 관리하기 때문에 중앙서버에 대한 의존도가 높을수록 서버의 과부화가 문제시 된다. 이러한 문제에 대한 하나의 해결책이 제시되었다. 즉, 모든 호스트를 서버/클라이언트의 역할을 병행하는 서번트(Servent)로 구성하여 기존 클라이언트/서버의 한계점을 극복하고 유연한 오버레이 네트웍을 지원할 수 있는 P2P(Peer -to-Peer) 모델을 말한다. P2P는 하나의 망안에서 저렴한 비용으로 광대한 저장 용량과 처리 자원 공유

* 배재대학교 컴퓨터공학과

접수일자 : 2007. 8. 14

등의 많은 효과를 얻을 수 있다. 이런 P2P에서 동료노드 사이의 통신을 지원하기 위해 이동 에이전트(Mobile Agent) 기술이 검토되고 있다[1,2]. 즉, MA가 피어를 대신하여 정보를 수집·분석하고 결과를 얻어낸 뒤 복귀하는 시스템을 말한다. 이동 단말의 사용은 보편화 되고 서비스가 확대됨으로 P2P망에서 이동 단말의 사용은 증가 되었으나, 저전력과 소형화라는 특성 때문에 작업부하와 작업량에 제약을 받는 시스템이다. 이러한 시스템에서 MA를 이동 단말의 대리인으로 사용하는 아이디어는 아주 유용한 기술이다. 따라서 이동중에 가장 큰 문제점이면서 시급한 해결책으로 요구되었던 핸드오프라는 결점을 MA는 효과적으로 처리할 수 있다. 뿐만 아니라 그 밖의 활용분야로 전자상거래에서 역할은 전자화폐, 개인 정보 등 중요한 정보를 전송한다.

그러나 MA의 보안문제는 초보적 연구수준에 머물러 있다. 그중에서도 악의적인 MA가 적법한 호스트를 공격하는 공격방법에 관한 대처방안의 연구는 진보했으나 악의적인 호스트가 적법한 MA를 공격했을 경우 대처하는 방법에 대해서는 미흡하다. 홈으로 복귀한 MA의 공격여부를 확인하는 작업은 에이전트 서버에게 많은 부담을 준다. 이미 에이전트가 공격을 받았다면 에이전트 서버와 MA는 필요없는 수행을 하게 된 셈이다. 홈으로 복귀하기 전 공격여부를 알 수 있다면 에이전트 서버와 MA를 더욱 효과적으로 활용할 수 있다.

이에 본 논문에서는 MA가 공격을 받지 않았음을 증명해 주는 암호기반의 STAS (Security Tracking and Auditing Server)시스템을 제안하였다. STAS는 감사를 대리 수행하는 서버로서 경로에 대한 즉시 검증과 MA가 수행 후에 공격을 받았는지 대리 감사를 한다. 이로써 공격여부를 홈으로 복귀하기 전에 알 수 있고 에이전트 서버의 부하를 줄일 수 있다.

본 논문의 구성은 2장에서는 관련 연구에 대하여 살펴보고 3장에서는 제안된 STAS 시스템을 소개한다. 4장에서는 시스템의 검토 및 고찰을 알아보고 결론으로 5장을 마친다.

Ⅱ. 관련 연구

MA가 악의적인 의도를 가지고 있을 경우 다른 에이전트 서버로 이동하여 바이러스나 웜(Worm)과 유사한 행동을 하여 빠르게 네트웍을 오염시키고, 또한 DoS(Denial of Service)와 같은 공격을 통해 에이전트 서버를 무력화 시킬 수 있다. 이런 공격에 대해서는 접근통제 기술이나 에이전트 인증 기술 등으로 해결 할 수 있다.

MA는 어느 호스트이든 노출된다는 문제가 있다. 따라서 악의적인 호스트가 MA의 코드, 데이터, 제어 흐름에 관한 모든 정보에 접근할 수 있다. 최근 동향을 살펴보면, MA의 보안 문제를 해결하기 위해 가장 많이 사용되고 있는 방법이 전자 서명이다. 전자 서명은 MA를 검증하는 과정으로 시작한다. MA의 코드값에 생성자는 서명을 한다. 생성자의 서명값을 점검함으로써 무결성을 체크하고 생성자의 신원을 확인할 수 있다.

이런 전자서명 방식을 이용한 연구로 참고문헌 [3]의 방법을 보면 MA를 인증하기 위해 공개키를 통하여 공유키를 분배하고 이를 통하여 전자서명을 검사한다. 또한 다중 도메인으로 확장할 수 있는 유연성을 제공하기도 하지만 홈네트웍이라는 제안점이 있다. 다른 연구에서는 도메인을 이용하여 MA를 보안하는 방법이 참고문헌 [4,5]에서 제시되고 있다. 이것은 신뢰된 에이전트 시스템을 한 도메인으로 정하여 도메인 등록 관리, 정책 관리, 도메인간의 인증서비스를 제공하는 장점이 있으나 잦은 이동성을 필요로 하는 에이전트로 인하여 보안연산의 증대를 가져온다.

참고문헌 [6]에서는 디지털 서명을 통해 경로상에 있는 각 서버를 확인하고 감사도구로 여행 정보와 실행상태를 확인하는 구조를 제안하고 있다. 여행 정보를 통하여 여행 경로를 확인하고 실행상태 정보는 로그 정보를 분석하여 공격유무를 판단한다. 이 시스템에서 에이전트의 경유 정보에 있는 모든 에이전트 서버사이에서 작업 부하를 일으키게 된다. 경유 정보에 있는 한 에이전트 서버는 다음 경유지의 서버로 에이전트를 전송할 것을 알리고 이에 대한 응답을 받아야 한다. 그리고 결과가 없을 경우 이상으로 간주하고 에이전트 홈으로 결과를 보고해야 한다. 이 방법은 경유 정보에 대한 변경 행위만을 감사할 수 있어 변경 행위에 대하여 즉시 발견이 어렵고 또한 최종 홈으로 돌아왔을 경우에만 데이터의 변조를 감사 할 수 있다.

참고문헌 [7]에서 제안된 기술을 보면, 신뢰되지 않은 네트웍을 순회하는 동안 에이전트를 보호하는 방법으로 암호화 기법을 사용한다. 에이전트의 실행 상태를 검

증자에게 전달하여 실행 상태에 대한 추적을 통해 에이전트를 보호하는 구조를 제안하고 있다. 제안한 시스템은 에이전트 자신이 높은 단계의 기밀성을 체크한다. 이것은 암호화에 따른 많은 비용을 요구한다. 추적 정보는 실행 코드마다 생성되므로 매우 커질 수 있다. 또한 에이전트가 이동할 때마다 추적 정보를 검증자에게 전달하므로 이에 따른 네트웍 자원을 많이 사용하게 되며 에이전트 자신이 기밀성 체크를 함으로 에이전트의 역할이 가중된다.

Ⅲ. STAS 시스템 제안

3.1 STAS 시스템 제안

상기와 같은 MA에 대한 관련연구를 바탕으로 본장에서는 새로운 STAS 시스템을 제안하였다. 동질의 네트웍인 구조적인 P2P환경에서 MA는 모든 피어를 순회하면서 임무를 수행하고 각각의 결과값을 STAS에게 전송한다. STAS 시스템은 MA로부터 받은 결과값에 대하여 안전한 경로를 증명하고 감사를 이용하여 공격유무를 확인하는 시스템이다. 이를 지원하기 위하여 STAS에 체계적인 DBMS를 운영한다.

3.2 STAS 설계

3.2.1 이동 에이전트 구조

구조적인 P2P망에 가입된 피어는 가입시 STAS의 주소를 획득하게 된다. MA는 다른 피어로 이동하여 임무를 마치고 획득한 피어의 주소와 데이터를 전자 서명하여 STAS에게 통보한다. MA 생성시 STAS의 주소를 탑재하게 되며 그 영역은 다음 그림과 같다.

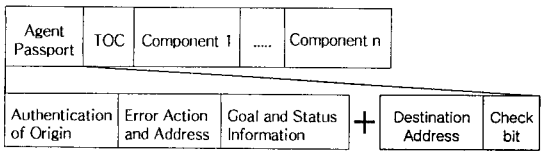


그림 1. 확장된 MA 구조
Fig. 1. Architecture of Extended Mobile agent

상기와 같이 MA를 정의하는 각 항목들은 다음과 같다.

• Agent Passport

이는 MA가 서로 다른 호스트간의 이동을 위해 요구되는 기본 정보들로 구성된다. MA 생성자의 신원에 대한 생성자 인증 정보, 에러 발생시 행동할 내용을 제공하는 에러 행위 및 주소 정보, MA의 목적 및 상태 그리고 다른 에이전트와의 관계를 표현한 목적 및 상태정보가 있다.

• TOC(Table of Contents)

이는 MA 구조에 대한 정보를 제공하는 부분으로 크기, 유형 및 중요도 필드로 구성된다.

• Component1 - Component n

MA가 수행해야 할 작업들이 기술된 곳이다.

기본 구조외에 본 논문에서 추가적으로 제안한 부분은 Destination Address와 Check bit이다. Destination Address는 STAS 주소가 탑재될 부분이다. Check bit는 STAS에게 결과값을 보고한 횟수를 카운트한 값이 들어가며 MA에 의해 그 값이 계속 증가한다.

3.2.2 STAS 구조

STAS의 구조를 나타내면 다음 그림 2와 같다. STAS는 크게 세 개의 구성요소로 나뉜다. MA를 감사하기 위해 필요한 부분과 이를 지원하는 시스템 제어부, 관리부로 나뉜다. STAS는 하위 계층의 STAS를 둔다. 이것은 공격이나 업무폭주에 대비한 것이다. 자세한 사항은 기능에 기술한다.

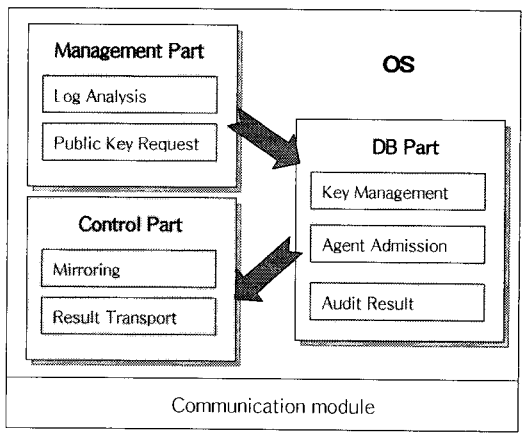


그림 2. STAS 구조
Fig. 2. STAS Architecture

3.2.3. STAS 기능

STAS 시스템은 다음과 같은 기능을 제공한다.

- 공개키 요청 : MA의 서명값을 확인하기 위해 공개키를 인증기관에 요청한다.
- Security Tracking : 경로 검증.
- Auditing : 전자서명된 데이터를 감사.
- 키 저장 DB : 받아온 공개키는 재사용에 대비하여 저장한다.
- 결과 저장 DB : MA가 감사기능을 요청한다. 정상 확인된 정보나 보안상에 문제가 되었던 결과를 날짜, 시간 및 어떤 문제점인지에 대하여 상세하게 저장한다.
- 결과 메시지 전송 : 보안상에 문제가 된 사항을 생성자에게 결과를 전송하여 생성자가 파악할 수 있게 한다.
- Admission : 이동 피어에 대한 등록 허가
- 미러링 : STAS 시스템이 다른 악의적인 호스트로부터 공격받을 수 있다. 혹 MA의 폭주량으로 인해 부하가 생길 경우 다른 서버로 미러링하여 기능을 제공한다.

3.3 STAS 시스템 동작과정

이동 피어는 CA(Certificate Authority)로부터 전자서명에 필요한 공개키를 요청하고 디렉토리에 공개한다. 구조적인 P2P에 가입하고 STAS의 주소를 얻어온다. STAS에 이동 피어의 ID값과 단말기에 대한 정보 즉, device No.와 함께 가입하며 다음의 형태를 따른다.

$$EKu_{STAS}\ [ID\_MP,\ Device\ No.\ ||\ N1]$$

MA는 임무수행을 위해 구조적인 P2P망안의 여러 피어로 순회하면서 원하는 정보를 획득한다. 처음 도착한 피어에서 작업을 마친 후 그 피어의 주소와 결과값을 전자 서명하여 STAS로 전송하고 MA는 다른 피어로 옮겨가며 구조적인 P2P망의 모든 피어를 순회한다. STAS는 MA가 전송한 서명값을 확인하기 위해 공개 디렉토리에서 공개키를 얻어 검증작업에 들어간다. 하나의 MA가 전송한 데이터값을 STAS는 DB에 체계적으로 저장한다. 구조적인 P2P에 가입된 모든 피어의 수 만큼 MA는 순회할 것이다. STAS는 이 결과값을 종합하여 최종 결과값을 산출해 낸다.

이동 피어는 이동성에 의하여 구조적인 P2P망의 다른 부분으로 이동해 갈수 있다. 어느 곳으로 이동했더라도 제약받지 않으며 이동 피어는 STAS에게 결과값을 요구한다. 그리고 STAS는 이동 피어의 공개키로 암호화하여 전송한다.

다음 그림은 STAS의 동작과정을 그림으로 나타내었으며 세부과정을 나열하였다.

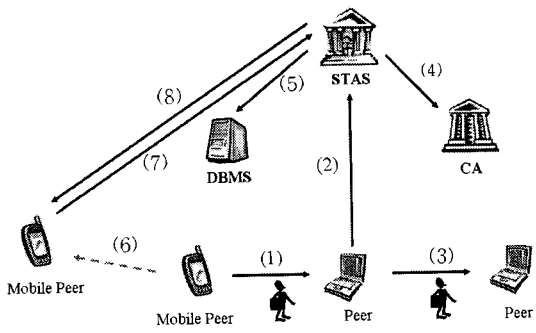


그림 3. MA의 검증과정

Fig. 3. verification process of Mobile agent

[약어 표기]

E : Encryption

D : Decryption

C : MP가 인증을 위해 STAS에게 보낸 암호화된 값

CQ : 질의를 서명한 값

CH : 중간 피어의 주소와 데이터를 해쉬한 값

CD : STAS가 MP의 공개키로 결과값을 암호화한 값

CR : 최종 결과값을 암호화한 값

add_p : 중간 피어 주소

step 0) MP는 STAS에 등록한다.

mp-> stas : C = { EKu_{STAS} [ID_MP, Device No.] }

*C*값을 *ma*에게 탑재하여 전송한다.

step 1) MP는 MA에게 C값과 질의를 서명한 값 CQ를 탑재하여 다음 피어로 전송한다.

CQ = EKr_{MP}[Query]

mp-> ma = { C || Query || CQ }

MA는 중간 피어에게 질의한다.

ma-> peer : { Query }

작업을 수행하여 이동한 피어의 주소와 데이터를 획득한다.

peer-> ma : { [data, add_p] }

step 2) MA는 획득한 주소와 데이터를 해쉬함수를 통해 해쉬값을 얻고,

ma : CH = H[data, add_p]

MA는 인증값과 질의, 이동한 피어의 주소, 데이터를 서명하여 STAS에게전송한다.

ma-> stas : { C || Query || CQ || [data , add_p] || CH }

step 3) MA는 다음 피어로 이동하여 같은 작업을 반복한다.

ma-> peer : { Query }

step 4) STAS는 받은 파일을 복호화하기 위해 MP의 공개키를 요청하고 감사한다.

stas : { C || Query || CQ || [data , add_p] || CH }

ma의 생성자를 인증하고,

ID_MP, Device No. = $DKr_{STAS}[C]$

쿼리 수정을 확인한다.

Query = DKu_{MP}[CQ],

Query = 기존 *Query*와 비교.

중간 피어를 확인하기 위해 주소와 데이터를 해쉬 함수로 돌린다. STAS가 확인한 값을 S_CH라 가정하면,

S_CH = H[add_p, data]

최종확인에 들어간다.

S_CH = CH(ma가 전송한 해쉬값)

step 5) 안전성이 검증되면 데이터를 이동 피어의 공개키로 암호화하여 DB에 저장한다.

stas : CD = EKu_{MP}[data, add_p]

step 6) MP는 다른 망으로 이동했다.

step 7) MP는 인증정보와 질의를 주고 최종 결과값을 요청한다.

mp-> stas : { C || Query || CQ || Result }

step 8) STAS는 MP의 인증정보과 질의를 확인하고 최종 결과값을 암호하여 전송한다.

*stas-> mp : MP*의 인증

{ID_MP, Device No.} = $DKr_{STAS}[C]$

쿼리를 확인

Query = DKu_{MP}[CQ]

MP에게 최종 결과값을 암호화하여,

CR = EKu_{MP} [data]

MP에게 전송한다.

{ C || Query || CQ || CR}

3.4 안전한 경로 증명 및 감사

다음과 같이 가정해 본다. 여행에 관심이 있는 피어들이 구조적인 P2P망을 형성하여 있다고 하자. 항공권 예매에 대하여 이동 피어는 어느 싸이트에서 운영하는 항공권이 가장 저렴한지를 파악하기 위해 MA에게 임무를 준다. MA는 구조적인 P2P망의 모든 피어들을 방문하여 해당정보 즉, 항공사, 날짜, 시간, 좌석예약수, 비용 등에 대한 정보를 중간 피어들로부터 얻어온다. 각각의 피어에서 제공받은 다음의 값 *{ Query || EKr_{MP}[Q] || [data, add_p] || H[data, add_p] }* 을 STAS에게 전송하면 STAS는 검증작업에 들어간다. A, B, C와 같은 피어가 있다고 하자. A에서 정보를 획득하고 전자 서명값을 생성하여 전송하고 B와 C도 같은 작업이 이루어졌을 경우 STAS의 검증작업은 다음과 같다.

첫째, 피어가 적법한 피어인지를 검증한다. 피어 B에서 작업을 마친후 다음의 값 *{ EKu_{STAS} [ID_MP, Device No.] || Query || EKr_{MP}[Query] || [data , add_p] || H[data, add_p]}* 를 전송하고 피어 C로 이동하였을 경우 STAS에서는 질의 변경에 대한 여부를 확인할 것이다. 왜냐하면 질의를 가장 비싼 항공사로 변경하였을 경우 본인의 싸이트가 선정될 것이기 때문이다. 다음과 같이 확인한다.

CQ = EKr_{MP}[Query]로 가정한다면,

Query = DKu_{MP}[CQ] 하여 질의 여부를 확인한다.

이미 질의가 (*DKu_{MP}[Q]*) ≠ *Query*가 일치하지 않는다면 악의적인 피어로 검증된다. 이는 경로상에 문제가 있음을 즉시로 추출할 수 있는 것이다. STAS는 더 이상 MA가 전송한 데이터를 인정하지 않는다. 둘째, 질의에 맞는 데이터인지 확인한다. 앞의 경우 B에서 질의가 수정되었음이 확인된다면 C의 데이터값은 형식이나 그 범위가 현저하게 다르기 때문에 의미없는 값이 되는 것이다. 이로써 MA의 오동작에 대한 검증을 할 수 있다.

Ⅳ. 제안 시스템의 검토 및 고찰

제안한 STAS 시스템의 가장 중요한 기능은 MA의 순회 경로에 대한 안정성 즉시 증명과 데이터의 무결성 체크이다. 이 두 가지 핵심 기능이 지원된 가운데 MA로부터 획득한 값으로 STAS는 최종 결과값을 산출하여 이동 피어로 전송한다. 이러한 시스템은 이동 피어의 이동중 통신장애와 MA의 업무부하에 최적의 효과를 지원한다.

STAS 시스템은 4가지 측면에서 검토 및 고찰 될 수 있다. 먼저 인증방법이다. MA는 임무를 가지고 있다. 이것은 곧 질의이다. 악의적인 중간 피어로부터 질의 수정이라는 공격을 확인하는 방법으로 전자 서명을 사용한다. 중간 피어로부터 얻은 결과를 MA는 해쉬 함수를 통해 서명값을 생성하고 이동 피어의 인증값, 결과값과 이에 대한 서명값, MA 질의와 질의의 서명값, 중간 피어의 주소를 함께 STAS로 전송하여 무결성, 인증, 부인봉쇄를 확인하도록 했다. 이동 피어에 대한 인증으로는 식별자 ID와 이동 단말의 Device No.값을 STAS의 공개키로 암호화하여 전송한다. Device No.는 유일한 번호이다. 동일값이 출현한다면 이 정보에 대한 유출이 확인되는 것이다. STAS는 최종 결과값을 이동 피어가 요청할 경우 질의, 질의에 대한 전자 서명값, 최종 결과값, 최종 결과값에 대한 전자 서명값을 이동 피어에게 전송한다. 이것은 참고문헌[3][4] [5][6]에서 제시한 인증 방법보다 그 강도를 높였다. MA를 확인하기 위해 인증서외에도 식별자 ID와 Device No.를 사용하여 MA의 확인정보에 대한 보안기술을 강화하였다. 그러나 피어가 이동한 후 STAS에게 결과값을 요구하도록 설계하였다. 만약 STAS가 결과값을 전송할 경우 이동 피어가 이동중이었다면 데이터 손실이 발생할 수도 있다. 이로써 이동성으로 인한 통신 장애를 막을 수 있다.

다음으로 STAS의 기반구조에 대한 검토를 해보면, STAS 시스템은 공개키 기반으로 MA의 전자 서명값을 대리 감사하는 기능을 제공한다. 효율적인 감사를 위해 지원해야 할 사항을 구체적으로 열거해 보면 다음과 같다.

첫째, DBMS를 구축하여 이동 피어의 등록 및 감사 결과를 체계적으로 저장하였다. 저장된 정보중 가장 중심이 되는 것은 이동 피어의 인증정보, MA가 오동작한 시점, 최종 결과값이다. 둘째, 감사중에 발생된 악의적인 중간 피어들에 대한 정보를 제공함으로 MA의 피해를 줄 일 수 있다. 셋째, STAS와 보안 통신하는 모든 가입된 피어들의 공개키를 저장하여 감사속도의 향상을 가져왔다. 제 3의 인증기관으로 키를 요청하는 단계가 번복되지 않기 때문이다.

세번째로는 MA 구조에 대한 검토이다. 참고문헌[6]에서 제시했듯이 기존의 시스템은 MA가 얻은 데이터와 서명값을 순회가 끝날때까지 가지고 이동한다. 이것은 MA를 무겁게 만들고 성능을 저하시키는 요인이다. 본 논문에서 제시한 방법은 중간의 데이터값을 서명하여 바로 STAS에게 전송하도록 MA 구조를 확장 설계하였다. 그러므로 데이터의 변경사항을 STAS 시스템에서 즉시 발견할 수 있고 MA에게 주어진 업무의 축소로 작업 수행에 부담을 줄였다.

마지막으로 STAS는 공격 받을 위험이 있고 감사업무의 폭주에 대비하여 미러링 기능을 제공한다. STAS는 구조적인 P2P망안의 가입된 모든 피어에 대한 감사를 하기 때문에 공격에도 강하게 동작하여야 한다. 제안한 STAS 시스템은 하위 계층의 STAS를 두어 상위 계층의 STAS가 권한을 하위계층의 STAS에게 위임할 경우 하위 계층의 STAS는 이 사실을 가입된 모든 피어에게 통보하여 감사에 차질을 주지 않도록 설계하였다.

Ⅴ. 결 론

본 논문은 MA의 보안 문제를 효과적으로 검증하는 방안으로 STAS 시스템을 제시하였다. 구조적인 P2P망안에 STAS 시스템을 도입하고 구조적인 P2P망에 가입하는 피어들은 가입과 동시에 STAS 주소를 획득한다. MA가 하나의 피어를 경유한 경우 STAS를 거쳐 보안감사를 실시함으로 에이전트 생성지에서 보안감사를 할 경우보다 더 빠르고 안전하며 더 편리하게 보안감사를 할 수 있다.

구조적인 P2P는 오버레이 네트웍으로 구성되므로 망의 확장이 유연하게 형성된다. STAS의 독립성은 망의 형태에 제한받지 않는다. 그러나 감사 기능은 하나의 구조적인 P2P망 안에서만 제공한다는 제약이 주어지므로 피어가 2개 이상의 구조적인 P2P망에 가입이 되어 있을 경우 STAS의 주소에 대한 별도의 관리를 필요로 하게 된다.

향후 연구는 STAS에서 감사할 수 있는 기능의 확대

이다. 본 논문에서는 공개키 기반의 감사를 제안하였으나 다양한 방법의 보안 프로토콜을 적용할 수 있는 기능이 추가된다면 훨씬 효과적인 감사 시스템이 될 것이다. 또한 모든 MA를 한곳에서 감사하기 때문에 악의적인 호스트로부터 공격의 대상이 된다. 외부에서의 공격에 대하여는 미러링보다 근본적으로 차단하는 방법에 대한 연구가 필요하다.

참고문헌

[1] Jukka Valkonen, "Mobile agents in Peer-to -Peer Networks", 2005.

[2] T.H.-T. Hu, B. Thai, and A. Seneviratne, "Supporting mobile devices in Gnutella file sharing network with mobile agents", 2003.

[3] 김재곤, 김구수, 엄영익, "홈 네트워크 환경에서 다중 도메인을 지원하는 공유키 및 공개키 기반의 이동 에이전트 인증기법", 정보보호학회논문지, 제14권 제5호, 2004.

[4] G. Noordende, F. Brazier and A. Tanenbaum, "A Security Framework for a Mobile Agent System," the second Internatinal Workshop on Security of Mobile Multiagent System(SeMAS 2002), July 2002.

[5] N. Motrovic and U.A. Arribalzaga, "Mobile Agent Security using Proxy-agents and Trusted domains," the second Internatinal Workshop on Security of Mobile Multiagent System(SeMAS 2002), July 2002.

[6] 백주성, 이동익. "디지탈 서명과 감사도구를 이용한 이동 에이전트의 보호", 한국정보과학회 학술발표논문집, Vol. 24, No 2, 1997.

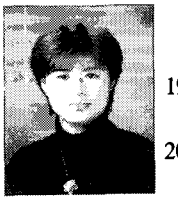
[7] Vigna, Ciovanni. "Protecting Mobile Agents through Tracing", Accepted paper for the Mobile Object Systems ECOOP Workshop, 1997.

[8] 전용희, 장정숙, "침입탐지 응용을 위한 이동 에이전트 기술에 대한 연구", 대구 가톨릭대학교 자연과학연구논문지, 제1권 1호, pp13-27, 2003.

[9] 이영화, 이남용, "이동 에이전트의 보호", 통신정보학회지, 제8권 제1호, pp77-98, 1998.

[10] 정성종, "이동 에이전트 기술", 정보과학회지, 제9권 1호, pp65-70, 1996.

저 자 소 개



김 선 영(Seon-Young Kim)

1999년 배재대학교 전자계산학과 (공학사)
2001년 배재대학교 컴퓨터공학과 (공학석사)
2004년 ~ 현재 배재대학교 컴퓨터공학과 (박사수료)
※관심분야 : 네트워크 보안, 컴퓨터 네트워크, MIPv6

조 인 준(In-June Jo)

1982년 전남대학교 계산통계 학과 (공학사)
1985년 전남대학교 전자계산 학과 (공학석사)
1999년 아주대학교 컴퓨터공학과 (공학박사)
1983년~1994년 한국전자통신연구원 선임연구원
1994년~현재 배재대학교 컴퓨터공학과 교수
※관심분야 : 정보보호, 컴퓨터 네트워크, 전산조직응용