디지털산업정보학회 논문지 제4권 제3호 - 2008년 9월

Ubiquitous Sensor Network에서 안전성 증가를 위한 신뢰모델과 신뢰값에 관한 프로토콜 설계*

장 근 원** · 서 장 원***

Design of Secure Protocol based on trust model and trust values for Ubiquitous Sensor Networks

Jang, Kun Won · Suh, Jang Won

〈Abstract〉

Mobile devices do not need the fixed network infrastructure in ad-hoc network, these devices communicate each other through the distributed control. Accordingly, mobile devices can discover several services using dynamic searching method and provide safely public ownership of these services. Ad-hoc network needs the distributed control and topology of dynamic network because the limited power for processing and network communication. This paper is devoted to provide the secure protocol that provides efficient services discovery using SDP(Service Discovery Protocol) and considers the security requirements. Proposed protocol provides the distributed control based on PKI without central server, the discovery of trusted service, secure telecommunication, the identification among mobile devices, and service access control by user authority.

Key Words : Ad-hoc Network,, Mobile Device, SDP, PKI

Ⅰ. 서론

최근 무선통신 관련 기술의 발달로 인하여 제한된 통신과 프로세스 처리능력을 가진 작은 크기의 무선장비들이 대량으로 사용되고 있다. 이러한 형태의 장비들은 휴대전화, PDA와 같이 눈에 보이는 장비들뿐만이나라 눈에 보이지 않으면서 우리들 주변에서 여러 가지 형태로 서비스를 제공해주는 "pervasive system"으로 알려져 있다. 무선통신 관련 장비 각각은 사용자에게 특별한 형태의 서비스를 제공하고 있으며 기술적인 진화가 점점 더 이루어져서 통신능력이 비약적으로 발전하게 되면 가까운 장래에 훨씬 더 복잡한 형태의 서비스들을 위해 무선장비들이 서로 연합하게 될 것이다. 이를 위해 ad-hoc 네트워크는 동적으로 연합된 서비스들을 발견해야하며 공유할 수 있어야 한다.

* 이 논문은 2007년도 동서울대학 부설 산업기술연구소의 지원에 의해 연구되었음

** (주)크레듀 HR연구소 선임연구원

*** 동서울대학 컴퓨터소프트웨어과 조교수(교신저자)

ad-hoc 네트워크는 제한된 능력을 가진 장비들로 이루어져 있기 때문에 장비들이 가지고 있는 자원들의 활용성을 극대화 하여야 한다. 이를 위해 세 가지 부분에 대하여 고려하여야 한다. 첫 번째로 프로세스를 처리하기 위해 요구되는 전력의 소모보다 이를 전송하기 위한 전력의 소모가 더 크기 때문에 전체 전송량을 줄이는 것이 매우 중요하다. 두 번째로 네트워크에 합류하거나 이탈하게 되었을때 서비스의 가용성과 비가용성에 대한 여부를 가능한 빨리 발견할 수 있는 구조를 설계하여야 한다. 마지막으로 악의적인 의도를 가진 서버와 사용자들의 부주의한 사용으로 인해 여러 가지 문제들이 발생할 수 있기 때문에 ad-hoc 네트워크에서의 보안성도 반드시 고려하여야 한다.

본 논문에서는 ad-hoc 네트워크가 요구하는 세 가지 측면의 요구사항을 모두 만족할 수 있는 안전한 service discovery protocol을 제안하고자 한다. 안전한 service discovery protocol은 ad-hoc 환경을 위해 분산된 제어구조를 제공하며 신뢰할 수 있는 서비스, 비밀정보의 보호, 안전한 통신체계, 무선장비들 사이의 식별, 서비스 접근 제어를 제시하였다.

본 논문의 구조는 다음과 같다. 2장에서 관련연구로 현재까지 제안된 service discovery protocol의 특징을 살펴보고, 3장에서는 안전한 알고리즘과 신뢰모델을 살펴보고, 4장에서는 이를 구현하기 위해 고려해야 될 이슈들을 정리하였다. 마지막으로 5장에서는 앞으로의 연구과제에 대해서 살펴보도록 하겠다.

Ⅱ. 관련연구

인터넷으로 제공되는 서비스의 수가 비약적으로 증가하게 되면 서비스를 자동으로 발견하여 사용자로 하여금 여러 서비스 중에서 사용 가능한 것을 선택할 수 있는 기능이 매우 중요하게 되며 특히 ad-hoc 네트워크 환경에서는 필수적인 요소가 된다. 이장에서는 무선장비의 속성을 포함하여 네트워크상에 존재하는 서비스를 자동으로 발견하고 해당 서비스가 자신의 존재를 동적으로 알려주는 서비스 발견(service discovery) 프로토콜에 대하여 살펴보고 무선 장비들 사이에 안전하게 정보를 주고받을 수 있게 해주는 신뢰모델 및 신뢰 값을 표시하는 프로토콜에 대하여 살펴보도록 하겠다.

2.1 Service Discovery Protocol

컴퓨터 네트워크를 이용한 공동사회가 형성되면서 많은 서비스가 존재하게 되었으며 이로 인해 SDP(Service Discovery Protocol)의 필요성이 대두되게 되었다. 따라서 많은 기업들, 컨소시엄, IETF(Internet Engineering Task Force)와 같은 연구기관들이 SDP에 대한 연구를 시작하였다. SDP에 대한 다양한 연구와 개발이 이루어지면서 대표적인 것들로는 SLP[1], Jini[2], Salutation[3]과 같은 고정된 네트워크에서 다양한 수준의 서비스를 제공하는 프로토콜들과 Upnp's SSDP[4], DEAPspae[5]와 같은 동적환경을 위한 프로토콜들이 제안되었다.

그러나 이와 같은 솔루션들은 ad-hoc 네트워크의 환경을 고려하지 못한 솔루션들로써 고정된 무선 네트워크 환경에 적합한 설계방식을 사용하였다. 먼저 SLP, Jini, Salutation와 같은 솔루션들은 중앙 서버를 필요로 한다[6]. 중앙서버는 디렉토리 서비스를 제공하며 시스템보안을 염두에 둔 신뢰 있는 기관들이 디렉토리 서비스를 관리한다. ad-hoc 네트워크는 중앙 서버와 같이 영구적인 목적을 가진 하나의 장비에 의존할 수가 없으며 또한 특정한 시간에 서버와 같은 역할을 하는 장비가 존재할 수 없다. 둘째로, SSDP와 같이 중앙 서버가 없이 작동하는 솔루션들은 무선네트워크의 가장 큰 제약점 중에 하나인 전력사용량의 제약사항을 고려하지 않고있다. SSDP는 유선 네트워크에서는 거의 비용이 들지 않는 multicast나 broadcast 전송방식을 활용하였으나 무선네트워크에서는 전력의 고갈을 가져오게 된다. 세 번째로 보안적인 고려가 잘 정의되지 않았다. SSDS[7]는 규모가 큰 네트워크

에서는 보안서비스를 제공할 수 있지만 ad-hoc 네트워크와 같은 모바일 서비스 환경에서는 제대로 작동하지 않는다. Splendor[8]는 인증서 취소목록(Certificate Revocation List)을 제공하지 않으며 PKI의 신뢰모델을 따르지 않고 있는 것이 단점이다. 이 두 가지 방식은 신뢰할 수 있는 서버에 의존적이며 IP 수준의 솔루션만을 제공하고 있다.

2.2 신뢰관계

ad-hoc 네트워크에서는 하나의 중앙 서버를 통해 다른 장비들을 신뢰하는 서비스 구조를 사용할 수 없기 때문에 여러 장비들을 통해 신뢰관계의 값을 계산하여야 한다. 따라서 신뢰관계의 값을 계산하기 위한 수학적 표현식[9]을 살펴볼 필요가 있다. 신뢰의 정도를 표현하기 위해 양수와 음수를 사용하며 각각의 개체에 대한 신뢰도를 나타내기 위해 개체의 단위 처리업무에 특정 값을 부여하며 이 값은 만약 단위업무가 수행되지 않았을 경우 1씩 줄어들고 그렇지 않으면 1씩 증가한다.

2.2.1 신뢰관계 분류

제한된 단위행위만을 수행한다면 그 개체를 완전히 신뢰할 필요가 없으며 해당 내용에 대해서만 신뢰관계를 형성한다. 인증 프로토콜의 역할을 하기위해서는 다음과 같은 6가지의 기능을 가지고 있어야 한다.

① key generation : 키 생성
② identification : 공개 키 사용
③ keeping secrets : 은밀성
④ non interference : 통신간섭 배제
(eavesdropping / impersonating)
⑤ clock synchronization : 동기화 유지
⑥ performing algorithmic steps : 프로토콜 명세서를 정확하게 따름

위의 분류에 대한 각각의 요소에 대한 신뢰관계는 다른 요소의 신뢰성과는 독립적이어야 한다.

2.2.2 직접 신뢰관계와 권고 신뢰관계

신뢰에는 직접신뢰(Direct trust)와 권고신뢰(Recommendation trust)라는 2가지 종류의 신뢰등급이 존재한다[10]. 네트워크상의 어떤 개체를 직접적으로 신뢰한다는 것은 모든 신뢰관계의 분류를 완벽하게 인정하는 것이다. 권고신뢰는 다른 개체의 신뢰관계 분류 중 특정 내용만을 신뢰하는 것이며 이는 제 3자 개체가 권고하는 내용을 인정하는 것이다. 권고신뢰는 제한된 방식으로 받아들여져야 하며 직접신뢰 하는 개체가 권고해주는 개체로부터 멀어질수록 신뢰도가 점점 더 낮아지게 되어 보다 많은 제한을 받게 된다. 직접신뢰와 권고신뢰를 나타내기 위한 표현식은 다음과 같다.

(1) 직접신뢰

$$P\ trust\underset{x}{^{seq}}\ Q\ value\ V$$

직접신뢰는 P가 Q에 대하여 신뢰관계 분류의 모든 기능을 양수 값으로 가지고 있음을 나타낸다. seq는 이전의 값을 연결해주는 개체들의 수열로써 P와 Q를 제외한 양수의 값을 전달하는 권고신뢰 관계로 볼 수 있다. V는 Q가 신뢰할 수 있는 행동을 보이는 확률을 나타내는 신뢰관계의 값이다. 이 값은 P가 Q에 대해 가지고 있는 양수 값의 수를 나타낸다. p가 양수 값일 때, 이를 나타내는 값 v_x는 다음과 같이 계산된다.

$$v_x(p) = 1 - \alpha^p \qquad (1)$$

(2) 권고신뢰

$$P\ trust.rec\underset{x}{^{seq}}\ Q\ when\ \mathcal{S}_p\ when.target\ \mathcal{S}_t\ value\ V$$

권고신뢰관계는 P가 신뢰관계 분류 x 대하여 제 3자

로부터 Q에 대한 정보를 얻어서 이를 받아들이고자 할 때 존재하는 관계이다. 이 신뢰성은 경로에 대한 제약 집합인 S_p 에 포함되는 개체가 중재하는 목표 제약집합인 S_t 에 포함되는 개체가 가지고 있는 값에 제약을 받는다. seq는 권고신뢰를 중재하는 개체의 수열이고 V는 신뢰관계의 값이다. 이 값은 P가 Q로부터 얻고자 하는 값의 일부분을 나타내고 Q가 권고하는 개체들에 대해 P가 포함하고 있는 값에 근거한다. 권고된 개체들을 나타내는 양수와 음수의 값인 p와 n 주어진 값을 통해 각각 권고된 신뢰 값인 V_r 은 다음의 식으로 계산된다.

$$Vr(p,n) = \begin{cases} 1-\alpha^{p-n} & \text{if } p > n \\ 0 & else \end{cases} \quad (2)$$

만약 음수 값이 양수 값보다 숫자적으로 더 많다면 신뢰 값은 0이 되고 해당개체는 신뢰권고의 제약 집합으로부터 제거한다.

$$Vr(p,n) = 0 \text{ for } p = 0$$
$$Vr(p,n) = approaches\,1\, with\, growing\ p \text{ and } n$$
$$Vr(p,n) = approaches\,0\, with\, growing\ n \text{ and } p$$

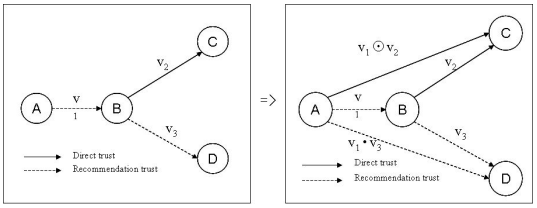
2.2.3 신뢰관계의 추출

신뢰도 표현식에 의한 신뢰정도의 표현은 신뢰권고가 이루어졌을 때 어떻게 새로운 신뢰도가 만들어지는지를 설명하는 규칙을 이끌어낸다. 이러한 규칙을 통하여 각 개체는 초기의 관계 집합으로부터 새로운 신뢰관계를 추출할 수가 있다. 신뢰모델 자체는 두 개의 개체들 사이에 신뢰관계를 추출할 수 있는 방법이 존재하지 않기 때문에 추출 알고리즘이 추가되었다. 여기서는 추론 규칙과 두 개의 유도 알고리즘을 설명한다.

(1) 추론 규칙

앞의 권고신뢰 최종 식을 통하여 추론규칙을 설명하면 다음과 같이 나타낼 수 있다.

1. $A trusts.rec_{x}^{seq_1} B\ when.path\ S_{p_1}\ when.target\ S_{t_1} value\ V_1$
2. $B trusts_{x}^{seq_2} C\ value\ V_2$
3. $B\ trusts.rec_{x}^{seq_3} D\ when.path\ S_{p_3}\ when.target\ S_{t_3} value\ V_3$



<그림 1> 신뢰관계의 유도

(2) 직접신뢰 유도

위의 표현식 1,2로부터 A에서 C로의 새로운 직접신뢰관계가 다음과 같이 유도된다.

(a) C is in S_{t_1} (C는 가능한 신뢰대상이다)
(b) All entities in seq_2 are also in S_{p_1} (직접신뢰관계를 중재하는 개체는 권고신뢰에 의해서 배제되지 않는다)
(c) seq_1 contains no entity from seq_2 (비순환성을 나타낸다)

새로운 권고신뢰의 경로는 seq_1, B, seq_2 로 구성되며 $seq_1 \circ B \circ seq_2$ 로 나타낸다. 새로운 신뢰관계의 값은 다음의 표현식으로 나타낸다.

$$V_1 \odot V_2 = 1-(1-V_2)^{V_1} \quad (3)$$

이 표현식은 식 (1)에 직접신뢰의 계산방법에 의해 유도되며 권고신뢰의 값을 표현한다. 만약 V_2 가 P의 양수 값을 기준으로 하면 다음의 등식으로 표현될 수 있다.

$$V_1 \odot V_2 = 1-(1-\alpha^{p})^{V_1} = 1-\alpha^{V_1 \cdot p} \quad (4)$$

따라서 새로운 값은 "$p \cdot V_1$" 의 값으로 표현할 수 있다.

(3) 권고신뢰 유도

식 (1), (4)로부터 새로운 권고신뢰의 관계가 A에서 D로 유도되며 다음과 같은 특성을 제공한다.

(a) seq_3 에 존재하는 D와 모든 개체들은 또한 모두 S_{p_1} 에 존재한다. (D와 D에 포함되는 신뢰관계의 중재자는 포함되지 않는다)

(b) seq_3 에 존재하는 D와 모든 개체들은 seq_1 에 존재하지 않는다. (비순환성을 나타낸다)

새로운 신뢰관계는 권고신뢰의 경로 $seq_1 \circ B \circ seq_3$ 로 나타내고, 제한된 경로 집합인 $S_{p_1} \cap S_{p_2}$, 제한된 대상 집합인 $S_{t_1} \cap S_{t_2}$, 그리고 $V_1 \bullet V_3$ 의 값을 포함한다. 신뢰 값에 대한 곱셈은 권고신뢰의 경로가 늘어날수록 신뢰 값이 작아지도록 해준다.

2.2.4 새로운 추론의 규칙

추론의 규칙을 다음과 같은 표현식으로 나타낼 수 있다.

(1) 새로운 직접신뢰의 규칙

$$
\begin{aligned}
&P\ trusts.rec\overset{seq_1}{x}\ Q\ when.path\ S_p\ when.target\ S_t\ value\ V_1\\
&\wedge\ Q\ trusts\overset{seq_2}{x}\ R\ value\ V_2\\
&\wedge\ R \in_s S_t\\
&\wedge\ \forall X\colon (X \in_l seq_2 \Rightarrow (X \in_s S_p \wedge X \notin_l P \circ seq_1))\\
&\Rightarrow P\ trusts\overset{seq_1 \circ Q \circ seq_2}{x}\ R\ value\ (V_1 \odot V_2)
\end{aligned}
$$

(2) 새로운 권고신뢰의 규칙

$$
\begin{aligned}
&P\ trusts.rec\overset{seq_1}{x}\ Q\ when.path\ S_{p_1}\ when.target\ S_{t_1}\ value\ V_1\\
&\wedge\ Q\ trusts.rec\overset{seq_2}{x}\ R\ when.path\ S_{p_2}\ when.target\ S_{t_2}\ value\ V_2\\
&\wedge\ \forall X\colon (X \in_l seq_2 \circ R \Rightarrow (X \in_s S_{p_1} \wedge X \notin_l P \circ seq_1))\\
&\Rightarrow P\ trusts.rec\overset{seq_1 \circ Q \circ seq_2}{x}\ R\ when.path\ (S_{p_1} \cap S_{p_2})\ when.target\ (S_{t_1} \cap S_{t_2})\ value\ (V_1 \bullet V_2)
\end{aligned}
$$

$\circ$ 는 수열간의 연결 또는 원소를 수열에 덧붙이는 관계를 표현하고 있다. 술부(predicate) $\in_l$과 $\in_s$ 는 수열이나 집합에 대한 원소의 포함관계를 나타낸다.

Ⅲ. 제안 Protocol

본 논문에서는 중앙 서버가 없는 ad-hoc 네트워크 환경을 고려하기 때문에 기존의 Service Discovery Protocol과 같은 디렉토리 서비스를 제공할 수 없다. 따라서 서비스의 존재를 알리기 위한 두 가지 방법을 제안한다. 먼저 "Pusth" 방법은 서비스를 제공하는 개체가 사전에 요구되지 않은 광고를 메시지로 보내고 다른 장비들은 이 광고 메시지를 듣고 자신들이 원하는 서비스를 선택하는 방법이다. 다음으로 "Pull" 방법은 서비스가 필요한 개체가 서비스를 요청하게 되면 서비스를 제공하는 개체가 요청에 응답하는 방법이다. ad-hoc 네트워크에서는 전송 메시지의 전체 수를 최소화 하는 것이 중요하며 이를 통해 전력 사용량을 줄여야 한다. 또한 네트워크로 개체가 합류하거나 떠날 때 제공되는 서비스의 가용성 여부를 가능한 빨리 발견할 수 있는 구조를 설계하여야 한다. 이러한 고려사항을 Push와 Pull의 방법을 선택할 때 적절히 고려할 수 있어야 한다.

본 논문에서는 Push와 Pull의 두 가지 방법을 통합하여 ad-hoc 네트워크 성능을 향상시킬 수 있는 새로운 Service Discovery Protocol을 제안한다. 또한 더불어 PKI를 응용한 신뢰모델을 설계한 후에 기존의 프로토콜을 이용하여 보안성을 높일 수 있다. 이러한 분산 신뢰모델은 신뢰도가 높은 중앙 서버나 계층구조적인 매커니즘을 요구하지 않는다.

3.1 기호 및 정의

본 논문에서 제안하는 ad-hoc 네트워크에서의 분산된 서비스 검색을 설명하기 위하여 다음과 같이 정의한다. ad-hoc 네트워크는 장치 D(devices)들로 구성되고 각 장치는 서비스 S를 제공하고 시간 T 동안 네트워크에서 가용한 상태로 존재한다. T는 사전에 장치 내부에 설정되어 있다.

각 D는 User Agent(UA)와 Service Agent(SA)를 가진

다. UA는 네트워크상에서 서비스를 제공하는 정보를 검색하기 위한 작업을 처리하며 SA는 장치에게 제공하는 서비스를 광고하는 작업을 처리한다. SA는 항상 서비스가 가용한 시간인 T를 포함한다. 각 장치들은 네트워크에서 광고되는 서비스의 목록을 포함하는 캐시메모리를 가지고 있다. 캐시의 각 원소 e는 UA와 연합하여 3개의 필드를 가지는데 서비스에 대한 설명, 서비스 생애주기, 서비스 만료시간으로 구성된다. 서비스 만료시간은 서비스가 가용한 시간으로 추정되는 시간이다. 이시간은 장치가 사용가능할 것으로 예정된 시간 T와 서비스 제공자가 서비스가 사용가능하다고 알려준 시간 중에 최소값으로 계산된다. 시간이 경과하면 캐시에 저장되어 있던 서비스에 대한 등록을 삭제한다.

보안적인 측면에서 각 장치들은 신뢰할 수 있는 장치와 이 장치와 결합한 신뢰등급에 대한 목록을 관리한다. 이 신뢰등급에 따라서 장치들은 장치가 제공하는 서비스를 자신의 캐시메모리에 저장할지의 여부를 결정한다. 만약 장치가 서비스에 접근하고자 할때는 가장 큰 신뢰등급을 가지고 있는 장비를 선택한다.

3.2 알고리즘

본 논문에서 제안하는 프로토콜은 두 개의 의무적인 메시지를 가진다. SPDP Service Request는 서비스 공지를 보내기 위해서 사용되며 SPDP Service Reply는 SPDP Service Request에 대한 응답을 하는데 사용된다. 또한 하나의 선택적인 메시지인 SPDP Service Deregister를 가질 수 있으며 이 메시지는 서비스가 더 이상 사용할 수 없다는 것을 알려준다.

3.2.1 User Agent

장비의 어플리케이션이나 최종 사용자가 특별한 형태의 서비스 또는 환경에 의해서 제공되는 서비스를 필요로 할 때 UA는 서비스를 요청한다. 만약 특별한 형태의 서비스를 요청하면 UA는 자신의 캐시에 존재하는 내부의 서비스 목록 중 에서 해당 서비스를 먼저 검색한다. 만약 서비스를 찾는다면 UA는 어플리케이션에 해당 서비스의 내용을 전달한다. 만약 서비스가 없다면 UA는 해당 서비스에 대한 Service Request를 브로드캐스트하고 CONFIG_WAIT_RPLY를 몇초동안 기다린다. 만약 응답메시지가 도착하지 않는다면 UA는 어플리케이션에게 해당 서비스가 사용할 수 없음을 알린다. 만약 어떤 응답메시지가 도착한다면 UA는 적절하게 자신의 캐시내용을 업데이트한다. 악의적인 장치에게까지 서비스 공지가 알려지는 것을 최소화하기위해 UA는 신뢰하지 못하는 장치가 제공하는 서비스는 저장하지 않는다. 이런 후에 신뢰하는 서버로부터 제공받은 서비스 설명을 어플리케이션에게 제공한다.

3.2.2 Service Agent

SA는 장치가 제공하는 서비스를 광고하며 Service Request 메시지를 처리하고 필요하다면 Service Reply를 생성하여야 한다. SA가 서비스 형태가 ALL인 Request를 받게 되면 다음과 같이 동작한다.

① 먼저 요구되는 서비스 S가 내부 서비스 인지 또는 캐시에 존재하는지를 확인한다. 만약 그렇다면 임의의 시간 t를 장치의 가용시간 T보다 반비례하여 생성한다. 즉 장비가 더 많은 시간을 서비스하면 할수록 장치가 먼저 응답할 확률이 더 높아야 한다.
② 이 시간동안 SA는 같은 요구에 대한 Service Reply가 있는지를 네트워크상에서 확인하여 캐시를 적당하게 업데이트 한다.
③ 시간이 만료 되었을 때 만약 SA가 서비스 S를 제공하는 추가적인 장치에 대한 정보가 있지만 공지되지 않았다면 Service Reply를 보낸다.

제안하는 프로토콜이 형태가 ALL인 서비스를 요청받으면 다음과 같이 동작 한다.

① 임의의 시간 t를 생성하며 장치의 가용시간 T와 캐시에 저장된 원소들의 수와 반비례한다. 즉, 장치가 서

비스를 제공할 시간이 길고 보다 큰 캐시를 가졌다면 먼저 응답할 확률이 더 높아야한다.

② 시간간격 t 동안에 SA는 같은 요청에 대한Service Reply를 네트워크상에서 확인하고 캐시를 적당히 업데이트 한다.

③ 시간이 만료가 되었을때 SA가 아직까지 공지를 하지 않은 새로운 서비스를 알고 있다면 캐시에 저장된 서비스목록과 내부 서비스목록을 더하여 Service Reply 메시지를 보낸다.

3.3 Security 알고리즘

제안하는 프로토콜은 anarchy PKI를 통해 인증, 권한부여, 데이터 무결성, 비밀성, 부인봉쇄 서비스를 제공한다.

상호인증은 장치사이의 challenge 매커니즘에 근거한다. 외부 장치가 ad-hoc 네트워크에 합류하기를 원할 때 분산된 신뢰환경을 구축하기 위해서는 해당 장치를 신뢰하고 있는 이미 신뢰할 수 있는 장치를 찾아야 한다. 만약 어느 누구도 해당 장치를 알고 있지 못하다면 해당 장치를 신뢰할지의 여부를 반드시 결정하여야 한다. 인증 서비스는 부인봉쇄의 목적으로 전자서명을 사용한다.

서비스에 대한 권한부여는 보안정책에 따라서 부여하며 이러한 정책은 사용자가 제공되는 서비스에 접근할 수 있는 신뢰등급을 결정하여야 한다.

데이터무결성과 비밀성은 IPSec[11]을 통해 보증할 수 있다. SPDP Service Request와 SPDP Service Reply 메시지는 무결성과 비밀성을 보증하기 위해 보호된다.

마지막으로 본 논문은 모든 장치가 스스로 인증기관(CA)으로 행동하면서 자신의 서비스를 위한 인증서를 발급하는 분산된 신뢰모델을 제공하는 anarchy 신뢰모델을 제안하였다. 신뢰관계는 인증기관 사이에서 형성되고 이를 통해 자신의 캐시에 존재하는 서비스 정보가 신뢰할 수 있음을 보증한다. 즉 각 장치는 다른 신뢰할 수 있는 user agent나 service agent로부터 생성된 서비스 정보만을 자신의 캐시에 저장한다.

각 장치들은 신뢰할 수 있는 장치들의 목록을 관리하고 이 장치들의 신뢰등급 또한 관리한다. 제안하는 모델은 매우 간단하며 신뢰등급이 단지 두 구성요소 사이에서만 형성되는SSDS와 Splendor와는 다르다.

제안하는 방법은 anarchy 신뢰모델을 선택하였다. 이는 다음과 같은 장점이 있다.

① 새로운 장치는 쉽게 네트워크에 합류할 수 있다.

② 만약 장치가 공격을 받아서 보안에 위협이 된다하더라도 전체 네트워크의 보안은 영향을 받지 않는다.

③ 장치들 사이에는 여러 개의 신뢰경로가 정의될 수 있다.

④ 위의 모든 요소는 중앙서버의 의존성을 회피할 수 있다.

3.3.1 Trust Model

기존에 제안되었던 PKI 신뢰모델들은 모바일과 자치네트워크 (Autonomous)에는 적당하지 않다[11]. 이런 연구들은 시간의 흐름에 따른 동적인 신뢰모델을 정의하지 않았으며 항상 root server에 의존해야만 하고 ad-hoc 네트워크에는 적당하지 않다.

분산된 신뢰모델을 공식화 하기위해서는 3가지 형태의 신뢰 속성을 정의한다. A, B, C 각각은 ad-hoc 네트워크에서의 하나의 장치를 의미하고 A와 B 두 장치사이에 신뢰관계 함수를 R(A, B)로 표현한다. R은 0과 1사이의 값을 나타내는 연속적인 함수이다. 0은 불신(distrust)을 나타내고 신뢰관계에 대한 정보가 없어서 알지 못할 경우는 1/2, 신뢰할 경우에는 1의 값을 사용한다. 함수 R의 속성은 다음과 같다.

① 재귀속성

모든 장치는 자기 자신을 신뢰한다. R(A, A) = 1

② 비대칭

A가 B를 신뢰한다고 해서 반드시 B가 A를 신뢰할 필요는 없다.

$$R(A,B)=\beta \wedge R(B,A)=\gamma \neq > \beta=\gamma$$

③ conditionally transitive

A가 B를 신뢰하고 B가 C를 신뢰한다면 A 상황에 따라 C를 신뢰한다.

$$R(A,B)=\beta \wedge R(B,C)=\gamma \Rightarrow R(A,C) \leq \beta \cdot \gamma$$

(1) 신뢰 표현식

새로운 장치는 상호작용을 위해 초기 신뢰 값을 형성하기위한 어떤 과거의 판단근거가 존재하지 않는다. 따라서 초기의 신뢰등급을 부여하기 위한 두 가지 방법을 제시하였다. 직접신뢰(direct trust)는 초기의 기본 값, 예를 들면 무지를 나타내는 값인 0.5로 부여한다. 이 값은 장치의 정보, 즉 장치형태나 소유주 등의 정보를 얻음으로 인해 신뢰 값이 향상될 수 있다. 또는 관리자의 개입을 통해 부여할 수 도 있다. 간접신뢰(indirect trust)는 장치들 간에 이미 신뢰관계가 형성되어 있을 때 적용할 수 있다. 신뢰 값은 신뢰할 수 있는 제 3자가 부여한 권고에 의해 생성된다. 여기서 신뢰할 수 있는 제 3자는 threshold α 0.5 보다 큰 신뢰 값을 가진다.

제 3자가 발행한 인증서도 일종의 권고신뢰의 한 구조이다. 이 경우 권고신뢰 값인 R(Di)는 α가 되고 그렇지 않으면 권고신뢰는 제 3자가 보낸 신뢰등급이 될 것이다. 장치 B의 신뢰 값은 다음과 같이 계산된다.

$$B(T_B)=\frac{1}{\sum_{i=1}^{n} T_i}\sum_{i=1}^{n} R(D_i)\times T_i \qquad (6)$$

R(Di)는 장치 i의 신뢰권고 값이다. T_i는 신뢰 값이고 n은 추천한 장치들의 수이다.

(2) 신뢰 값의 변화

앞에서 살펴본바와 같이 신뢰등급이 시간이 흐름에 따라 지속적으로 변하기 때문에 신뢰 값에 대한 학습은 계속해서 경험적으로 그리고 동적으로 얻게 되어 사실상 신뢰 값에 대한 많은 집합들이 존재한다. 신뢰 값은 특별한 상황에서는 양수와 음수 값에 따라 변하게 된다. 따라서 새로운 신뢰 값 T_{i+1}은 이전의 신뢰 값 T_i, 장치의 행위의 결과 값인 V_a에 대해서 과거와 현재의 값을 고려해야 계산해야 한다.

과거의 값의 가중치로서 $0 < \beta < 1$ 사이의 값을 가지는 파라미터인 β를 정의하면 다음과 같다.

$$T_{i+1}=\beta \times T_i+(1-\beta)\times V_a \qquad (7)$$

위 함수를 통해서 신뢰 값은 장치의 행위에 따라서 증가하거나 감소한다.

Ⅳ. 평가 및 적용 이슈

본 논문이 제안하는 방식은 ad-hoc 네트워크에 맞게 설계된 프로토콜로써 다음과 같은 특징을 제공한다.

- 개방형 분산 아키텍처를 근간으로 하기 때문에 중앙집중적인 통제방식을 필요로 하지 않는다(메인 서버).
- 단순한 아키텍처를 지향하여 단지 두 가지 형태의 구성요소만을 필요로 한다. 두 가지 구성요소는 유저 에이전트와 서비스 에이전트이다.
- 가용 서비스를 발견하기 위하여 자치 에이전트와 모바일 에이전트를 제공한다.
- 서비스 발견을 위해 필요한 정보의 전송 횟수를 최소화하여 모든 디바이스에서 사용하는 전력양을 최소화한다.
- 디바이스가 요구하는 보안수준을 보증하기위하여 기존의 제안된 보안 모델을 통합하였다.

본 논문에서 제공하는 보안 서비스는 디바이스 간 상호인증, 데이터 무결성, 비밀성, 부인봉쇄, 그리고 신뢰등급에 따른 접근제어 등의 보안 이슈를 고려하였다. 성능은 다음과 같은 결과를 얻었다.

- ad-hoc 네트워크에서 가용한 서비스의 발견을 위해 전송된 메시지의 수가 감소
- ad-hoc 네트워크에서 가용하지 않은 서비스를 발견할 확률의 감소
- 디바이스에 요구되는 보안 서비스를 구현하기 위해 IPSec과 신뢰모델의 상호운영성을 적용

V. 결론

제안하는 방식은 ad-hoc 네트워크 환경에서 분산된 관리 방식을 적용하여 전력사용의 양을 최소화하고 안전한 통신 구조를 이루기 위한 서비스모델을 연구하였다. 구현상의 어려움으로 Network Simulator 2 를 사용하였으며 여러 가지 디바이스 환경을 고려하기 위하여 다양한 시나리오를 고려하고 있다. 향후 이를 통해 여러 시나리오 상에서 제안하는 프로토콜을 적용하도록 할 예정이다.

본 논문이 제안하는 방식은 ad-hoc 네트워크에 따라 설계된 service discovery 프로토콜이다. 프로토콜은 분산된 개방형 구조를 기초하여 중앙 서버를 필요로 하지 않으며 단지 두 가지 형태의 구성요소인 user agent와 service agent만을 사용하기 때문에 매우 간단하다. 또한 자치형 매커니즘과 가용한 서비스의 발견을 위해 단순한 방법을 사용하는 mobile agent를 제공한다. 마지막으로 서비스를 발견하기위해 필요한 메시지의 수를 최소로 만들어주기 때문에 모든 장치들의 전력소모 양을 최소로 한다.

참고문헌

[1] Erik Guttman, "Service Location Protocol: Automatic Discovery of IP Network Services," *IEEE Internet Computing*, 3(4): 71-80, July 1999.

[2] Sun. "Jini Architectural Overview," *Technical White Paper*, http://www.sum.com/jini/, 1999.

[3] Salutation Consortium. "Salutation Architecture Overview," *White Paper*, http://www.salutation.org/whitepaper /, 1998.2005.

[4] Universal Plug and Play Forum, "Universal Plug and Play Device Architecture," *Version 0.91*, March 2000.

[5] Michael Nidd, "Service Discovery in DEAPspace," *IEEE Personal Communications*,, August 2001.

[6] E. Guttman, C. Perkins, and J. Kempf, "Service Templates and Service:Schemes," *IETF*, RFC 2609, June 1999.

[7] Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, and Randy H. Katz, "An architecture for a secure service discovery service," *In proceeding of Mobicom* '99, 1999.

[8] F. Zhu, M. Mutka, and L. Ni, "Splendor: A secure, private, and location-aware service discovery protocol supporting mobile services," in Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (Percom '03). IEEE Computer Society, pp. 235-242, March 2003.

[9] R. Yahalom, B. Klein, and Th. Beth, "Trust Relationships in Secure Systems-A Distributed Authentication Perspective," Proceeding of IEEE Symposium on Research in Security and Privacy, pp.150-164, 1993.

[10] T. Beth, M. Borcherding, and B. Klein, "Valuation

of trust in open networks," *in Proceedings of the European Symposium on Research in Computer Security (ESORICS '94, Brighton, UK)*, Heidelberg, Germany, 875 in Lecture Notes in Computer Science, pp. 3-18, Springer-Verlag, November 1994.

[11] S. Kent and R. Atkinson, "Security Architecture for the internet protocol (IPSec)," November 1998.

■ 저자소개 ■

장 근 원
Jang, Kun Won

2007년 9월~현재 (주)크레듀 HR 연구소 선임연구원
2007년 2월 승실대학교 컴퓨터학과(공학박사)
2003년 2월 승실대학교 정보통신학과(공학석사)
1998년 2월 고려대학교 영어영문학과(문학사)

관심분야 : 정보보호, Sensor Network, DRM, Steganography
E-mail : jaques72@naver.com

서 장 원
Suh, Jang Won

2001년 9월~현재 동서울대학 컴퓨터소프트웨어과 조교수
2000년 8월 승실대학교 컴퓨터학과(공학박사)
1996년 2월 승실대학교 전산공학과(공학석사)
1992년 2월 서울산업대학교 전산학과(공학사)

관심분야 : 정보보호, 암호 이론, 암호 시스템
E-mail : jwsuh@dsc.ac.kr

논문접수일 : 2008년 6월 30일
수 정 일 : 2008년 8월 6일
게재확정일 : 2008년 8월 12일