안전한 유비쿼터스 서비스를 위한 MANET의 인증서 관리 시스템에 관한 연구

오 석 심*

Certificate Management System of MANET for Stable Ubiquitous Service

Suk-Sim Oh*

요 약

본 논문에서는 유비쿼터스 컴퓨팅의 핵심 기술인 MANET 환경에서 발생할 수 있는 보안 요구사항을 도출하고, MANET을 구성하는 멤버 노드들의 협력적인 부분 인증서 관리 서비스에 의해 인증서 관리 및 멤버 노드들의 동적인 변화에 즉각적으로 적응할 수 있는 시스템 모델을 제안한다. 제안한 모델을 통하여 MANET환경에서 발생되는 부하 집중 문제를 해결하고, 클러스터 내에서 통신하고 있는 기존의 노드들이 새로운 노드의 유입에 영향을 받지 않고 능동적으로 인증 서비스를 수행 할 수 있도록 높은 확장성과, 보안 위협을 방지할 수 있는 모델을 제안 하도록 한다. 아울러 제안한 시스템을 시뮬레이션을 통해 안정성 및 효율성과 견고성을 평가해 보도록 한다.

ABSTRACT

This study addressed security requirements for ad-hoc network environments, which lies at the heart of the ubiquitous computing revolution and proposed a partially-distributed certificate management system that can ensure security in mobile ad-hoc networks. The proposed model is characterized by its ability to handle dynamic mobility of nodes, minimize routing load and enhance expandability of network by allowing participating nodes to authenticate each other without being interrupted by joining the cluster. The security, efficiency and robustness of the proposed model were evaluated through simulation.

키워드

Ubiquitous Computing, MANET, Certificate Management MANET Security

Ⅰ. 서 론

Ubiquitous(유비쿼터스)는 심오한 기술로 보는 것이 아니라, 일상 생활속에 일부분으로 보고 있다[1]. 기존의 보안은 사이버 공간, 물리적인 공간이 디지털화 되어 컴퓨터에 저장된 정보들이 문제가 되었지만, 유비쿼터스 특성을 고려해 볼 때 유비쿼터스 네트워크에서는 개인의 모든 정보가 노출 될 수 있다는 보안(security)상의 문제점을 안고 있다. 또한 기존의 네트워크에서는 침입하는 장소가 개인의 컴퓨터로 한정되는 반면, 유비쿼터스 네트워크에서는 개인의 사적인 모든 공간이 노출되는 셈이 되는 것이다. 따라서, 유비쿼터스 네트워크에서의 사이버 테러는 물리공간과 사물, 그리고 육체에 대한 테러를 포함하게 되며, 개인이나 기업과 국가의 정보보호를 뛰어 넘어

* 남부대학교 디지털경영정보학과

접수일자 : 2006. 6. 9

광범위한 공간 보호가 요구된다.

유비쿼터스의 핵심 기술인 MANET의 가장 매력적인 점은, 중앙통제로부터 완전히 독립해 사용자가 네트워크 사용에 더 많은 자유와 유연성을 얻게 된다는 것이다. 하지만, MANET에서의 보안(security)은 각 멤버들이 수시로 변하는 동적인 구성을 가지고, 다른 멤버와는 독립적으로 운영되는 특징 때문에 기존 네트워크보다 보안 대책이 쉽지 않을 것으로 인식되고 있다[2][3][4].

MANET의 많은 Routing Protocol 및 Security Mechanism이 제안되었지만 기존의 설계 방식으로는 완벽한 MANET의 보안 대책이 되기는 어려울 것으로 판단되며 새로운 개념의 검토가 이루어져야 할 것이며 현재, 많은 기관과 단체에서 연구가 진행 중이다[5][6][7][8].

MANET 환경의 사용자 인증 메커니즘은 모두 공개키 기반 암호기술을 이용하고 있어 PKI(Public Key Infrastructure) 구축이 요구된다. PKI 메커니즘은 그 근본이 되는 공개키 암호의 안전성에 기인하며 이를 통해 전자 서명의 안정성을 보장 받을 수 있게 된다. 하지만 무선화 네트워크상의 신뢰 된 인증기관(trusted authentication authority)의 부재로 인해 공개키 기반 암호화 기법을 바로 적용 하는 데에는 문제가 있으며, 실제로 무선 네트워크 상에서의 인증은 사용자 인증(user authentication)이 아닌 디바이스 인증에 머물고 있는 실정이다. 또한 장치들의 안전한 사용을 위해 보안 요구에 따라 MAC 계층에서의 대칭 키 암호 시스템을 채택하고 있지만, 키를 안전하게 교환하는 방법에 대해서는 언급하고 있지 않은 취약점을 안고 있다.

MANET 환경 하에서 공개키 기반 암호화 기법을 적용하려고 할 때, Secret Share를 이용하여 k개의 부분 인증서(partial certificate) 결합에 의해 생성되는 인증서의 생성시간의 경우[2], Key의 bit값이 증가함에 따라 인증서 생성시간이 증대되는 단점이 있다. 이는 packet data length가 증가함에 따라 overhead 발생률도 증가하게 되고, MANET의 안정성 및 효율성을 저해하는 요인이 된다. 불안정한 네트워크는 통신 중 인증을 위한 메시지가 중간에 도청 또는 변조될 위험에 노출될 수가 있으며, 메시지가 유실될 경우 안정된 네트워크를 보장 받을 수 없게 된다. 따라서 본 논문에서는 k개의 부분 인증서를 결합하여 완전한 인증서를 생성하는데 소요되는 시간을 줄이고, packet data length가 늘어난다 하더라도 일정한 비율의 delivery time 유지하여 네트워크의 안정성(安定) 및 효율성을 보장 받을 수 있는 모델을 제안한다. 아울러 제안한 시스템 모델을 시뮬레이션을 통해 안정성 및 효율성과 견고성을 평가해 보도록 한다.

Ⅱ. 시스템의 구성

먼저 클러스터헤드와 클러스터 멤버간의 관계가 형성된 후 클러스터 헤드는 CA로서 도메인에 사용될 공개키와 인증서 서명을 위한 비밀키 쌍을 생성 및 구축한다. 클러스터 헤드는 생성한 공개키를 백본망을 통해 다른 클러스터 헤드에 유니캐스트함으로써 각 클러스터헤드는 도메인 공개키를 생성 보유하게 된다.

CA 역활을 담당하는 클러스터 헤드는 좌표평면상에 k개의 클러스터 멤버를 알고 있을 경우, 서로의 신뢰관계를 기반으로 교차인증(Cross-Certification)의 방법으로 Secret Share의 타당성을 검증한 후, 클러스터 백본망을 통해 다른 CA로 유니캐스트 한다. 이 때, 전송 받은 Secret Share의 값이 정당한 값인지 검증하고, 만일 검증이 실패하면 해당노드에게 실패한 Secret Share에 대한 재전송을 요청한다. 검증계수를 사용하여 성공적으로 전송 받은 난수에 대한 Secret Share의 검증이 완료되면, CA는 비밀키에 대한 부분키를 생성한다. 각 클러스터 멤버들은 부분 인증서를 생성하고, 인증서 발행을 위해 공개키에 대한 서명을 수행하고자 할 때 인접하는 멤버들에게 부분 인증서를 요청하게 된다. 부분 인증서는 CA역활을 하는 클러스터 헤드를 통해 해당 클러스터 멤버에게 전송되고 임계치 이상의 부분 인증서를 획득하게 되면 부분 인증서를 합하여 완전한 인증서를 생성할 수가 있다.

MANET의 특성상 멤버 노드들은 다른 관리 도메인에 가입하고 탈퇴하게 되는 다이나믹한 특성을 지니게 되는데, 새로운 노드가 클러스트 내에 진입 할 경우, 원활한 인증서 발급 서비스를 할 수 있도록 Secret Share를 분배해 줘야 한다. 제안하는 시스템에서는 Self-initialization 메커니즘을 적용하여 새로운 노드가 도메인 내의 한 클러스터 영역에 진입할 경우, 원활한 인증서 발급 서비스를 수행하기 위한 부분 서명키 SK를 분배해 주는 역할을 수행하게 된다.

Ⅲ. 시스템의 설계

클러스터 헤드들은 각자 생성한 임의의 난수를 이용하

여 도메인의 공개키를 협력적으로 생성하고 검증 가능한 부분 비밀키들을 분배한다. 그런데 클러스터 토폴로지가 변하게 되면 클러스터를 구성하는 멤버들의 역할도 변경되므로 클러스터 내의 키 일치 또한 대응되게 변화되어야 한다. 이는 키 일치를 위한 메시지 교환을 증가시키므로 그 연산 시간이 오래 걸릴 뿐 만 아니라 네트워크에 큰 부담이 된다. 따라서 본 논문에서는 키 일치를 위한 메시지를 부분 비밀키에 첨부하여 전달하도록 한다. 도메인 공개키와 인증서 서명을 위한 비밀키 쌍을 생성 및 구축하는 과정은 다음과 같다.

3.1. 도메인 공개키와 비밀키 생성 및 구축

CA역활을 하는 클러스터 헤드(이하 CA)는 임의의 난수 x_i를 발생시키고, 부분 공개키 값 PK_i를 생성하여 CA 백본망을 통해 다른 CA에게 유니캐스트 한다. 부분 공개키가 완료되면 각 CA 노드는 다음의 연산에 의해 도메인 공개키를 생성하여 보유하게 된다.

$$PK = \prod_{i=1}^{n} PK_i$$

4개의 클러스터가 형성되었다면, 각 CA는 4개의 부분 공개키 $PK_1, PK_2, \cdots, PK_4$를 획득하게 된다. CA 노드는 좌표평면상 k개의 노드를 알고 있을 경우 각각 생성한 난수 x_i를 y 절편으로 하는 K-1차 다항식 f_i를 생성한다. 서로의 신뢰관계를 기반으로 교차인증을 하는 방법으로 Secret Share의 타당성을 검증하기 위해 다항식 f_i의 계수들 f_{ij}에 대한 교차 정보 F_{ji}를 클러스터 백본망을 통해 다른 CA 노드들에게 유니캐스트 한다.

$$f_i = f_{i0} + f_{i1}x^1 + \cdots + f_{iK-1}x^{K-1}$$

단, $(f_{i0} = x_i)$

$$F_{ji} = g^{f_{ij}},\ (j = 0, \cdots, k-1)$$

CA 노드는 자신이 생성한 임의의 수에 대한 Secret Share 값 $SK_{ij} = g^{f1j}$를 클러스터 백본망을 통해 다른 CA 노드들에게 유니캐스트 방식으로 전송한다. Secret Share의 전송이 완료되면 CA 노드는 전송 받은 Secret Share SK_{ij}가 정당한 값인지 검증하고, 만일 검증이 실패할 경우 해당 노드에게 실패한 Secret Share SK_{ij}에 대한 재전송을 요청한다. 검증 계수를 사용하여 성공적으로 전송 받은 난수에 대한 Secret Share의 검증이 모두 완료되면 CA 노드는 다음 연산을 통해 Secret Share에 대한 부분키를 생성한다.

$$SK = \prod_{j=1}^{n} SK_{ji}$$

3.2. Secret Share의 분배

클러스터 헤드는 도메인 CA의 공개키와 비밀키 쌍을 생성하는데, 클러스터 내의 멤버들 중 대표로 참여하는 협력적인 분산 CA 역할을 한다. 클러스터 공개키는 앞 절의 과정을 통해 모든 클러스터 멤버들에게 알려진다.

각 클러스터 헤드는 도메인 CA의 공개키와 비밀키 쌍을 생성하는데, 클러스터 내의 멤버들 중 대표로 참여하여 협력적인 분산 CA 역할을 한다. 클러스터 공개키는 앞 절의 과정을 통해 모든 클러스터 멤버들에게 알려진다. 각 클러스터 헤드가 도메인 CA 공개키와 Secret Share를 구축하면 클러스터 내의 k 멤버들에게 Secret Share를 분배한다. 새로운 인증서는 멤버들 간의 협력에 의해 발급되어지며 비밀키는 인증키 역할을 한다. 이것은 분산된 CA 구조를 이루고 있으며, 신속하게 신뢰할 수 있는 다른 관리 도메인(Administrative Domain)에 가입하고 탈퇴하게 되는 다이나믹한 특성을 잘 적용한 것으로 이동성과 가용성이 좋아 키 일치 확률을 높일 수 있다. 그림.1는 각 클러스터내의 모든 멤버들에게 Secret Share 분배가 모두 완료된 후, 분산 CA 구조를 나타내고 있는 것이다.

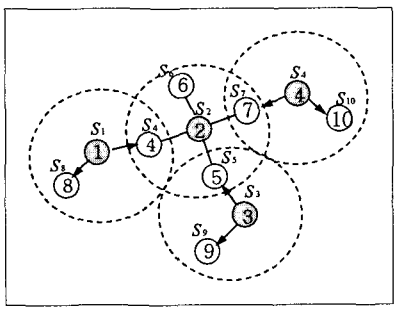

그림 1. 분산된 비밀키
Fig. 1. Distributed Secret Share

3.3. 인증서 생성

클러스터가 형성이 되면 인증기관 역할을 하기 위해 Secret Share를 가지고 있는 N개의 노드 중에서 k개의 노

드가 부분 전자 서명을 한 인증서 조각을 모으면 인증기관이 발행하는 인증서와 일치하는 인증서를 만들 수 있다.

본 논문에서는 Threshold Cryptography를 적용하여 CA는 생성된 부분 비밀 서명키를 클러스터 내의 k 노드들에게 분산시킴으로써 클러스터 내에 존재하는 노드가 중앙 집중적인 인증기관에 의존하지 않고 임계치 이상의 근접 노드와의 협력을 통해 인증서를 발급 받을 수 있도록 하였다. 클러스터를 구성하고 있는 각 멤버들은 Secret Share를 가지고 있으며, 인증서를 가지고 있지 않은 노드가 클러스터에 구성원으로 접속하고자 할 경우 인증서에 전자서명을 하기 위한 용도로 Secret Share를 사용 한다.

각 클러스터 멤버들은 부분 인증서를 생성한다. 인증서 발행을 위해 공개키에 대한 서명을 수행하고자 할 때, 인접하는 멤버들에게 부분 인증서를 요청한다. 부분 인증서는 클러스터 헤드를 통해 해당 노드로 전송되고, 임계치 이상의 부분 인증서를 획득하면, 부분인증서를 합하여 완전한 인증서를 생성할 수 있다. 가령 CA2가 인증서 발행을 위해 공개키에 대한 서명을 수행하고자 할 경우, CA2는 클러스터 내의 멤버 노드들에게 Secret Share를 요청하고, 임계치 이상의 Secret Share가 획득되면 서명키를 생성하여 인증서를 발행 할 수 있다. 그림.2는 인증서 생성 방식의 예를 도시한 것이다.

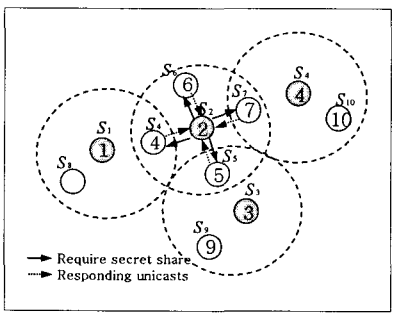

그림 2. 완전한 인증서 생성
Fig. 2. Creation of complete certificate

형성된 클러스터 내에 개별적으로 존재하는 노드들의 Secret Share를 공유하는 방법은, 비밀 다항식(Secret Polynomial) $f(x)$에 의해 이루어진다. 즉, 개인키/공개키 쌍 SK=<d,n> / PK<e,n>을 공개하는데 좌표명면상의 k개의 노드들을 알고 있을 경우 Lagrange Interpolation(라그랑지 보간법)을 통해 K-1차 다항식을 다음 수식(1)과 같이 정의함으로써 완전한 비밀(Secret)이 복구 될 수 있다.

$$f(x) = d + f_1 \cdot x + f_2 \cdot x^2 + \cdots, \\ + f_{K-1} \cdot x^{K-1}, N(> K) \quad (1)$$

단, 해당 집단 내의 노드들의 수가 k이하인 경우, 완전한 비밀에 대한 정보는 얻을 수 없다. 여기서 Shared Secret $f(0) = d$이고, 클러스터 C에 존재하는 노드 i가 가지는 Secret Share는 $P_i = f(C_i) \bmod n$이다.

인증서를 생성하기 위해서는 비밀키 SK=<d,n>를 사용하여 전자서명을 하여야 한다. 비밀키 SK=<d,n>는 k개의 Secret Share를 수집하여 d를 계산하면 쉽게 드러나게 된다.

$$d \equiv \sum_{i=1}^{K} (P_{C_i} \cdot l_{C_i}(0) \bmod n) \quad (2) \\ \equiv \sum_{i=1}^{K} SK_i \,(\bmod n)$$

여기서, $l_{C_i}(0)$은 Lagrange Coefficient(라그랑지 계수)이며, 클러스터 내에 존재하는 k개 이상의 노드들이 협력에 의해 부분 서명한 인증서 조각을 사용하여 SK=<d,n>를 생성할 수 있다. Lagrange Interpolation에 의해 sk_i는 수식(2)의 다항식 d로부터 복구될 수 있으며, 수식(1)에 의해 다음 수식(3)과 같이 표현될 수 있다.

$$d \equiv \sum_{i=1}^{K} (P_{C_i} \cdot l_{C_i} \bmod n) \quad (3) \\ \equiv t \times n + d$$

수식 (3)의 $(P_{C_i} \cdot l_{C_i} \bmod n)$는 n에 대한 모듈로 연산에 대해 d와 같은 값을 가지는 값이므로 k개의 인증서 조각을 곱하는 방법만으로는 완벽한 인증서를 생성하는데 한계가 있다. 따라서, K-bounded Coalition Offsetting 알고리즘을 적용하여 완전한 인증서를 생성할 수 있다[2].

$M^{t \times n + d} \equiv M^{t \times n} \times M^d \equiv M^d (\bmod n)$ (4)

여기서, M은 시스템의 공개키<e,n>이고, Threshold K는 각 집단 내에 존재하는 복수의 노드들을 의미하며, 반드시 큰 수일 필요는 없다. 새로운 인증서는 멤버들 간의

협력에 의해 발급되어지며 비밀키는 인증키 역할을 한다. 이것은 분산된 CA 구조를 이루고 있으며, 신속하게 신뢰할 수 있는 다른 관리 도메인(Administrative Domain)에 가입하고 탈퇴하게 되는 다이나믹한 특성을 잘 적용한 것으로 이동성과 가용성이 좋아 키 일치 확률을 높일 수 있다.

Ⅳ. 실험 및 평가

제안한 시스템의 성능 분석을 위하여 시뮬레이션 도구로는 유닉스 및 리눅스 환경에서 최적으로 구동되고, TCL(Tool Command Language) 스크립트 언어에 객체(Object) 지향의 개념을 추가한 OTCL(Object TCL)과 C++로 구성된 NS2(Network Simulator 2)를 이용하였다. 이를 통한 패킷 flow에 대한 추적결과를 근거로 데이터의 양과 시간을 계산, 다른 MANET의 라우팅 프로토콜들과 비교 분석하였다. 이를 통해 제안한 시스템의 네트워크 보안의 안전성 및 효율성과 견고성 등의 평가를 해 보도록 한다.

4.1. 시스템의 효율성 및 안정성 평가

표.1는 제안한 시스템에서 인증서를 계산 하는 시간을 도시한 것이다. PCC는 Secret Share를 사용하여 Partial Certificate를 계산하는 시간을 보여주고 있으며, Combine은 k Partial Certificates의 결합에 의하여 생성되는 인증서 생성 시간을 보여주고 있다[7].

표 1. TPC와 PCC와의 비교
Table 1. Compare TPC with PCC

key (bit)	PCC	Combine	TPC	Sum
512	0.0466	0.0928	0.407	0.301
768	0.1198	0.2416	0.462	0.361
1024	0.2610	0.5280	0.488	0.322
1280	0.4590	0.9742	0.551	0.426
1536	0.7944	1.5598	0.801	0.462
2048	1.7058	3.4410	1.461	0.488

TPC(가칭, Times of the Partial certificate Computation which using Lagrange interpolation operation)는 Lagrange interpolation을 이용하여 Partial Certification을 계산한 시간을 나타내고, Sum은 모든 k Partial Certificates를 합하여 비밀키를 획득, 인증서를 생성하는 계산시간이다.

그림.3는 Pentium III/500 laptop인 경우 Key의 bit수가 증가에 따른, PCC, Combine과 제안한 시스템의 TPC, Sum를 각각 비교한 결과를 그래프로 도시한 것이다.

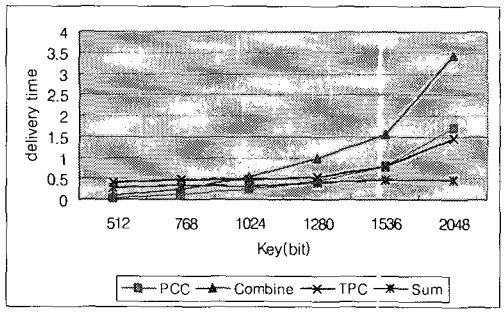

그림 3. TPC와 PCC의 비교
Fig. 3. Compare TPC with PCC

결과를 분석해보면, Key의 bit가 증가함에 따라, PCC, Combine이 점차적으로 증가되는 것을 볼 수 있다. 이러한 결과는 packet data length가 증가함에 따라 overhead 발생률도 증가하게 되므로, 네크워크의 효율성을 떨어뜨리게 된다. 이러한 문제점은 Ad-Hoc 네크워크의 안정성 및 효율성을 저해하는 요인이 된다. 또한 불안정한 네트워크로 인하여 전달되는 메시지가 중간에 도청 또는 변조될 위험이 있으며, 통신 중 메시지가 도중에 유실될 가능성이 높게 되어 네트워크의 안정성을 저해하는 요인이 된다.

반면, 제안한 시스템에서는 Key의 bit가 증가하더라도 TPC, Sum의 값은 bit가 늘어남에 따라 다소 증가함을 보여주고 있으나 기존의 PCC, Combine에 보다 현저하게 줄어들었음을 확인 할 수 있다. 특히, 모든 부분 인증서를 합하여 비밀키를 획득하여, 인증서를 생성하는 시간 Sum은 Packet data length가 늘어난다 하더라도 일정한 비율의 시간을 갖는 것을 확인할 수 있다.

따라서, 본 논문에서 제안한 시스템에서는 bit(packet data length)가 늘어난다 하더라고 일정한 delivery time을 유지함으로 네트워크의 안정성 및 효율성을 보장 받을 수 있다. 또한, k 값의 변화에 따라 인증서 조각을 생성하는데 요구되는 시간과 완전한 인증서를 생성하는데 요구되는 시간을 측정한 결과, Partial Certificate들은 연합 멤버들에 의해 일정하게 계산되기 때문에 매개 변수 k가 시스템 효율에 영향을 미치지 않은 것을 발견할 수 있으며, 특히 Partial Certificate를 요구하는 측면에서 Certificate를 생성하는 모든 작동이 이루어지므로 오버헤드 계산에 관해서 적절한 결과를 보여주고 있다.

4.2. 견고성의 평가

이 절에서는 CA 비밀키를 구축하기 위해 전송되는 패킷 및 부분 키 메시지의 키 일치율과 노드 수의 증가에 따른 라우팅의 오버헤드를 비교해 봄으로써 견고성을 평가해 보도록 한다. 먼저, 노드 수의 증가와 각 노드 간 송·수신 되는 length of message(packet) 비율을 비교해 봄으로써 키 일치성을 평가해 보았다.

그림.4는 노드 k가 30, 40인 경우의 키 일치율을 도시한 것이다. Avg. rate of Key agreement는 전송하는 패킷과 수신되는 패킷의 데이터의 비율을 나타내고, length of message는 패킷의 크기를 나타낸다.

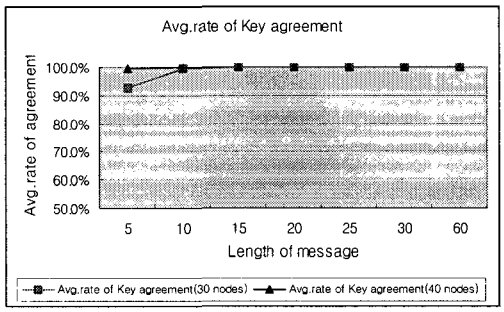

그림 4. 키 일치율의 비교
Fig. 4. Avg. rate of Key agreement

클러스터내의 노드가 30, 40 증가하더라도 100%에 근사한 키 일치율을 보여주고 있으며, packet length가 증가하더라도 키 일치율에는 영향을 미치지 않음을 알 수 있다. 이는 CA 비밀키를 구축하기 위해 전송되는 패킷 및 부분 키 메시지의 키를 완벽하게 생성 할 수 있음을 보여주고 있다. 또한 클러스터 내의 노드 수가 증가하더라도 노드 간 packet을 일정하게 송·수신 할 수 있게 됨으로 오버헤드를 줄일 수 있어 견고성을 보장 받을 수 있다.

V. 결 론

k Partial Certificates의 결합에 의해 생성되는 인증서의 생성시간은 Secret Share를 사용하여 Partial certificate를 계산하는 시간의 경우 Key의 bit값이 증가함에 따라 인증서 생성 시간이 증대된다. 이는 packet data length가 증가함에 따라 overhead도 발생률도 증가하게 되므로 네트워크의 효율성을 떨어트리게 된다. 이러한 문제점은 MANET의 안정성 및 효율성을 저해하는 요인이 된다. 불안정한 네트워크는 통신 중 인증을 위한 메시지가 중간에 도청 또는 변조될 위험에 노출될 수가 있으며, 메시지가 유실 될 경우 안정된 네트워크를 보장 받을 수 없게 된다. 반면, 본 논문에서 제안하는 시스템에서는 k Partial Certificate를 결합하여 완정한 인증서를 생성하는 시간이, bit가 늘어남에 따라 다소 증가함을 보여주고 있으나 기존의 시스템보다 현저하게 줄어들었음을 확인 할 수 있었다. 특히, 모든 Partial Certificate를 합하여 비밀키를 획득, 인증서를 생성하는 시간은 Packet data length가 늘어난다 하더라도 일정한 비율의 delivery time 유지함으로 네트워크의 안정성 및 효율성을 보장 받을 수 있음을 확인 할 수 있었다.

또한, k 값의 변화에 따라 인증서 조각을 생성하는데 요구되는 시간과 완전한 인증서를 생성하는데 요구되는 시간을 측정한 결과, Partial Certificate들은 연합 멤버들에 의해 일정하게 계산되기 때문에 매개 변수 k가 시스템 효율에 영향을 미치지 않은 것을 발견할 수 있으며, 특히 Partial Certificate를 요구하는 측면에서 Certificate를 생성하는 모든 작동이 이루어지므로 오버헤드 계산에 관해서 적절한 결과를 보여주었다. 아울러, 본 논문의 연구 결과는 CA로부터 부분 서명키를 사전에 분배 받거나 중앙 집중적인 인증기관에 의한 문제를 완전 해결하였으며, 이는 중앙 CA에 의존하지 않는 MANET에 적합한 모델로 사료된다. 또한 새로 노드가 도메인 내의 한 클러스터 영역에 진입했을 경우에도, 원활한 인증서 발급 서비스를 수행하여 통신하고 있는 기존의 노드들이 새로운 노드의 유입에 영향을 받지 않고 능동적으로 인증 서비스를 수행할 수 있도록 높은 확장성을 지원할 수 있었다.

참고문헌

[1] Charles E. Perkins, "Ad Hoc Networking," Addison Wesley, 2001.

[2] J.Kong. p.Zerfos. H.Luo. S.Lu and L.Zhang. "Providing robust and ubiquitous security support for mobile Ad-Hoc networks". IEEE ICNP.2001.

[3] Y. Zhang and W. Lee. "intrusion detection in wireless ad-hoc network." ACM Mobicom. Aug. 2000

[4] L.Zhou and Z. Hass. "Securing ad hoc networks." IEEE

Network. pp. 24-30. Nov/Dec. 1999

[5] D. Johnson, D, Maltz, Y-C. Hu, and J. Jetcheva. The dynamic source routing protocol for mobile ad hoc network. IEEE Internet Draft, March 2001. draft-ietf-manet=dsr-05.txt(work in progress)

[6] E.M. Belding-Royer and C,-K. Tho. A review of corrent routing protocols for ad-hoc mobile wireless networks. IEEE Personal Communications Magazine, pp. 46-55, April 1999.

[7] C.E Perkins and E. M. Royer. "Ad hoc on-demand distance vector routing". In IEEE Workshop on Mobile Computing Systems and Applications, pp.90-100, Feb. 1999.

[8] S. Murthy and J.J. Garcia-Lunca-Aceves. "An efficient routing protocol for wireless networks". ACM Mobile Networks and Applications Journal, pp.183-197, Oct. 1996

[9] A. Herzberg, M. jakobsson, S. Jarecki, H. Krawczyk, and M.Yung, "Proactive public-key and signature schemes.", In proceedings of the Fourth Annual Conference on computer Communications Security", ACM, 1997. pp. 100-120.

[10] A. Herzberg, S. Jarecki, H. Krawczyk, and M.Yung, "Proactive secret sharing or : How to cope with perpetual leakage", Advances in Cryptolgy-Crypto '95, Santa Barbara, Califonia, U.S.A, Aug. 1995, pp.457-469.

저자소개

오 석 심(Oh Suk-Sim)

전주대학교대학원미술학과(석사)
현. 남부대학교 디지털경영정보학과 박사과정
현. 전남과학대학 강사
광주시립민속박물관 자문위원
※관심분야 : 디지털 경영, 유비쿼터스 컴퓨팅. 전통민예품 관광상품화 및 체험프로그램 환경 개발.