위성 통신망에서 키 분배 프로토콜의 성능분석

정회원 진 상 민*, 조 동 호**, 이 상 한***, 강 건 우***

Performance Analysis of Key Distribution Protocol in Satellite Network

Sang-min Jin*, Dong-ho Cho**, Sang-han Lee***, Gen-woo Kang***
Regular Members

요 약

본 논문에서는 위성 통신망에서의 효율적인 키 분배 프로토콜에 대한 성능을 분석하기 위하여 지상망에서 적용되는 키 분배 프로토콜중에서 Kerberos와 같은 push 모델과 X9.17로 대표되는 pull 모델에 대한 키 분배 처리 절차를 분석하고 분석된 키 분배 프로토콜이 위성망을 통하여 수행되는 절차를 기술하였다. 아울러, VSAT 위성 통신에서 키 분배 프로토콜이 Inband/Outband 채널을 통하여 처리되는 경우에 키 분배 프로토콜의 성능을 처리 지연과 처리율 측면에서 분석하였다.

ABSTRACT

In this paper, in other to analyze the performance of efficient key distribution protocol, we evaluate the key distribution processing method of push model like the Kerberos and pull model such as X.9.17 used in wired network. Also, we evaluate the procedure of key distribution protocol in satellite network. Besides, we analyze the performance of key distribution protocol procedure in view of delay and throughput in the case that key distribution protocol is processed based on Inband/Outband channel.

I. 서 론

현재 미국, 일본 등 선진 각국에서 행해지고 있는 초고속위성통신기술개발 프로젝트의 대부분은 초고속정보통신의 기반인 광대역 ATM 전송상의 문제점, 위성신호의 전파지연시간에 따른 문제점 및 지상통신망 등과의 상호접속에 따른 제반 문제점을 해결하고 효율적으로 서비스를 제공하기 위한 연구들이다. 즉, B-ISDN, 고속 LAN과 위성망과의 상호연동, ATM 셀 전송실험 및 초고속 데이터 전송 등에 대한 실험이 진행되고 있으며 그에 따른 지구국들도 위성제원과 전송속도 등에 따라 다양하게 개발 시험되고 있다. 일반적으로 위성 통신망은 위성국과 지구국으로 구성되며 각 지구국은 규모가 큰 지구국 혹은 규모가 적은 VSAT일 수도 있다. 지구국에는 기존의 지상망이나 사용자 단말기가 연결될 수 있다. 위성 통신망에서는 대상이 되는 정보를 보호하기 위해 암호화 기법이 사용되어진다. 따라서 위성 통신망에서 암호화 기법은 링크상에 오가는 정보를 보호하기 위해 사용되고, 전체 시스템에서의 보안을 위해서는 시스템에 적합한 보안 시스템이 필요하다. 그리고 이에 따른 각종 보안 절차나 보안 메카니즘이 정의되어야 한다.

여러 가지 암호화 메카니즘 중에서 분산환경에서 인증 메카니즘에 대한 연구가 활발하게 이루어지고 있는데 대표적인 방법은 MIT의 Athena 계획의 일

* (주)두루넷연구소연구1팀(smjin@corp.thrunet.com),
** 한국과학기술원 전기 및 전자공학부(dhcho@ee.kaist.ac.kr) *** 국방과학연구소
논문번호 : 98545-1221, 접수일자 : 1998년 12월 21일
※ 본 연구는 국방과학연구소의 지원을 받아 수행한 위탁연구 과제의 연구결과입니다.

환으로 개발된 Kerberos이다. 이는 인증 서비스이며 안전한 서버의 서비스를 통하여 사용자들을 인증할 수 있는 시스템이며 공통키 암호 방식을 사용하고 있다[1],[2].

Kerberos에서는 영역간의 서비스에 대한 제안이 없으므로 X.509를 적용하여 영역간에 연결된 체인을 이용하여 다른 영역과 인증을 수행하도록 하는데 X.509는 공개키 방식에 기반을 두고 있다[3],[4]. 사용자의 비밀키와 상대방을 인증하기 위하여 인증을 수행하는 상호간에 사용하는 핸드세이킹 함수를 알고 있다는 가정하에서 프로토콜이 설계되었지만, 메시지의 재전송 문제가 발생하는 단점을 가지고 있으며, 이를 보완하기 위하여 타임 스탬프 개념이 적용되었다. 또한, 각 사용자의 암호화 키가 누설되지 않았다는 가정하에서 핸드세이킹 함수를 사용하지 않아도 되는 키분배 프로토콜이 제시되었다[6]. 공통키 암호화 시스템에서의 통신키 분배 및 사용자 인증 프로토콜로서 인증 서버를 통한 사용자 상호인증과 메시지 재전송 탐지를 위한 사용자 비밀키로서 챌린지(challenge) 암호화등을 수행하여 전송하는 방식이 제안되었는데, 이는 키를 인증서버에게서 제공받거나 자신이 직접 관리해야 하는 단점이 존재한다[7]. 이러한 문제를 해결하기 위하여 사용자 ID 정보에 의한 통신키 분배 및 사용자 인증 프로토콜이 제안되었다[8].

본 논문에서는 위성통신 암호시스템에서의 키 관리 방안으로 기존의 인증 및 키 분배 프로콜을 조사 분석하였다. Kerberos 프로토콜로 대표되는 push 모델의 인증 및 키 분배 프로토콜들을 연구하고, X9.17로 대표되는 pull 모델의 인증 및 키 분배 프로토콜을 분석하였다. Push 모델과 pull 모델 프로토콜의 성능을 평가하기 위하여 키 분배 프로토콜이 신호 채널을 사용할 때의 성능과 데이터 채널을 사용할 때의 성능을 지연과 처리율 측면에서 분석하였다.

서론에 이어 2 장에서는 위성통신망에 대하여 기술하였고, 제 3 장에서는 키 분배 프로토콜의 동작 절차에 대하여 기술하였다. 또한, 제 4 장에서는 본 논문의 성능분석을 위하여 키 분배 프로토콜의 모델링에 대하여 기술하였고 시뮬레이션을 통하여 분석한 결과를 기술하였다. 마지막으로, 제 5 장에서 본 논문의 결론을 맺는다.

Ⅱ. VSAT 위성통신망

일반적인 VSAT 망의 구성도가 다음 그림 1에 나타나 있다. 각각의 네트워크가 하나의 위성에 연결되어 VSAT(Very Small Aperture Terminal) 망을 이루고 있다. 각 네트워크에는 그 네트워크에서 키 관리 및 분배를 담당하는 KDC(Key Distribution Center) 가 존재한다.

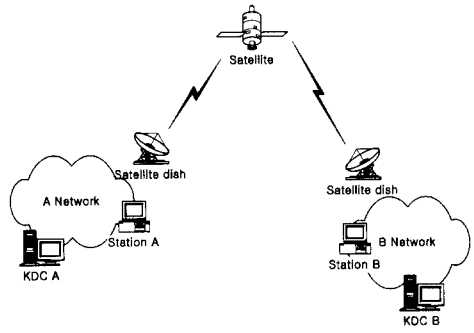


그림 1. 위성 통신망 구성도

VSAT망은 응용분야 (데이타, 음성, 영상, 복합서비스), 지연특성, 망의 크기, 단말의 비용 및 설치장소, 위성체의 특성, 안테나의 크기 등을 고려하여 설계하여야 하며 제공되는 서비스에 따라 다음과 같이 크게 세가지 형태로 나눌수 있다.

방송망은 VSAT 기술을 상업적으로 응용한 최초의 망으로 초기에는 C-band를 사용하였으나 현재 Ku-band쪽으로 발전하고 있다. 방송망에서 지구국은 패킷화된 데이터, 방송 수준의 음성, 영상 혹은 이 세가지가 결합된 신호를 모든 혹은 일부 그룹의 VSAT으로 단방향 전송한다.

점대점 망은 지구국 없이 두 지점간에 음성, 데이터 및 영상의 양방향 전송을 제공한다. 이 망은 원격 batch, 파일전송, 디지틀 팩시밀리등 여러 가지 서비스를 제공하는 SCPC(Single Carrier per Channel) 전송을 사용한다. 양방향 interactive 망은 현재 가장 널리 이용되는 망으로 star 형태로 구성되며 다양한 범위의 음성, 영상 및 데이터 서비스를 지구국과 다수의 VSAT 사이에 제공한다. 지구국에서 원격 VSAT으로의 outband 전송은 고속 TDM 채널을 이용하여 모든 VSAT에 전송하고 VSAT은 outband를 항상 수신하고 있다가 자신에게 해당되는 데이터가 전송되면 받아들인다.

Ⅲ. 키 분배 프로토콜

1. 개요

X9.17과 같은 시스템에서 사용되는 pull 모델은 스테이션 A가 B측에 통신을 요구하게 되면, B측이 필요한 키를 KDC(Key Distribution Center)로부터 얻은 후에 안전한 통신을 수행하는 프로토콜 모델로서 WAN과 같은 환경에 적합하다. 앞으로 표현되는 내용은 다음의 표기를 따르게 된다.

- A, B, C, X, Y : Principals
- K_x : X의 KEK(Key Encryption Key) (KDC와 X만이 알 수 있음)
- K_{xy} : X와 Y가 공유하는 비밀 키
- $E_x(M)$: 키 K_x를 사용하여 메시지 M을 암호화
- $E_{xy}(M)$: 키 K_{xy}를 사용하여 메시지 M을 암호화
- $MAC_x(M)$: 키 K_x를 사용한 메시지 M의 MAC
- $MAC_{xy}(M)$: 키 K_{xy}를 사용한 메시지 M의 MAC

일반적인 인증 프로토콜에서 A, B는 양방향 인증을 수행하는 두 개의 네트워크 요소의 ID를 나타낸다. 변수 N_a와 N_b는 'nonces'로 불리는 임시적인 난수이다. MAC_{ba}와 MAC_{ab} 등의 표기는 메시지 인증 코드(Message Authentication Codes)로서 K_{ba}, K_{ab}와 같은 키로 계산된 암호화 단방향 해쉬 함수를 나타내며, 자신의 파라메터 문자의 출처를 보장하기 위해 사용된다. MAC 파라메터 간의 쉼표는 MAC 함수가 적용된 메시지에 연속해서 MAC 함수가 적용되는 것을 의미한다. DES와 같은 대칭적 암호화 시스템에서 MAC 기능을 구현할 때, K_{ba}, K_{ab}는 A와 B에 의해 공유되는 비밀키가 된다.

Pull 모델이나 push 모델에서 도메인간의 통신을 위해서는 다음과 같은 가정이 필요하다. 먼저 서로 다른 도메인에 있는 KDC와의 안전한 통신을 수행하기 위해 도메인 간에 사용되는 키가 존재하게 된다. 도메인간의 키 K_{12}는 KDC_1과 KDC_2가 공유하게 된다. 또한, 이름을 가지고 상대방이 속한 도메인을 알 수 있다는 가정도 필요하다.

2. A-B-K Pull 모델

일반적으로, KDC는 키를 다루는 중요한 요소이기 때문에 KDC가 도메인 외부 요소와 통신하는 경우 KDC가 갖는 보안이 침해받을 수 있으므로, A-B-K pull 모델에서 KDC는 통신에 참여하지 않게 된다. 따라서, KDC는 직접 통신이 허용되지 않으며 단지, 자신의 도메인에 속한 스테이션의 티켓과 상대방 도메인의 KDC로의 티켓을 생성하게 된다. A-B-K Pull 모델의 동작절차가 그림 2에 나타나 있다. 이 A-B-K 모델의 동작 절차는 다음과 같다.

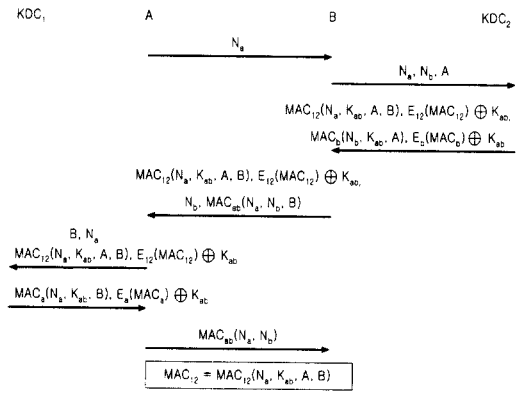


그림 2. A-B-K 도메인간의 프로토콜

Flow 1. A가 N_a(nonce A)에 기반한 인증을 수행하기 위해 B와 접촉한다.

Flow 2. B는 KDC에게 A의 ID와 N_a와 자신의 N_b를 전송한다. 여기서 B의 이름은 전송되는 메시지에서 추론될 수 있다고 가정한다.

Flow 3. KDC는 메시지 흐름 2를 수신하고, B의 KDC_2는 A가 외부 도메인에 속한 것을 발견하고 다음과 같은 절차를 수행한다.

a) KDC_2는 A의 이름에서 A의 도메인을 추출한다.

b) 도메인명에서 KDC_2는 A의 KDC(KDC_1)를 식별하여 KDC_1과 공유하는 K_{12}를 얻게 된다.

c) 일반적으로 KDC_2는 새로운 키 K_{ab}를 생성하고 B를 위한 티켓을 생성한다. 그러나 A의 KEK K_a를 모르기 때문에 A를 위한 티켓은 생성하지 못하고 KDC_1으로의 티켓을 생성하여 B에 전송하게 된다.

Flow 4. B는 자신의 티켓을 읽고, A에게 자신의 인증 토큰과 티켓을 전송한다.(B는 A가 다른 도메인에 있다는 사실을 모름) KDC는 새로 생성된 K_{ab}를 포함하는 두 개의 티켓을 B로 전송한다.

Flow 5. 티켓을 수신한 B는 K_{ab}를 추출하여 MAC의 재계산을 통해 완전성과 유효성을 검사

한다. 이 때 B는 나중에 A와 통신하기 위해 수신 티켓을 저장할 수도 있다. B로부터 티켓을 받은 A는 다음의 절차를 수행하고 티켓을 KDC_1에게 전송한다.

a) K_a를 이용하여 키를 추출하기 위해 노력한다.

b) B의 이름에서 B가 다른 도메인에 속한다는 것을 알게 한다.

Flow 6. KDC_1은 티켓을 검사하고 K_{at}를 추출하여, 티켓이 KDC_2에서 생성되었고, A에게 B와 통신이 가능함을 알린다.

Flow 7. 메시지를 수신한 A는 K_{ab}를 추출하여 MAC의 재계산을 통해 B로부터의 인증 코드가 정확한지 검사하고, 자신의 $MAC_{ab}(N_a, N_b)$를 B에 전송하여 키 분배 절차를 마친다.

Flow 6에서 KDC_1은 티켓 사용 기간의 종료 여부는 알 수 없지만, 티켓의 생성시에 종료 시간이 속성으로 첨가되기 때문에 이러한 문제는 고려하지 않아도 된다. 이 A-B-K Pull 모델은 매번 메시지 교환을 위해서 새롭게 티켓을 KDC로부터 얻을 필요가 없이 A, B에 티켓을 저장하여 쉽게 K_{ab}를 추출하여 사용할 수 있는 장점을 가진 반면에, A로의 티켓이 B에 의해 저장되어 있고, B가 A와의 통신을 위해 저장된 티켓을 전송하여 수신된 티켓이 KDC_1로 전송되는 경우 문제가 발생할 수 있다.

3. A-B-K-K Pull 모델

KDC 간의 통신을 지원하기 위한 A-B-K-K pull 모델의 메시지 흐름이 그림 3에 나타나 있다. 제시된 A-B-K-K 모델의 동작 절차는 다음과 같다.

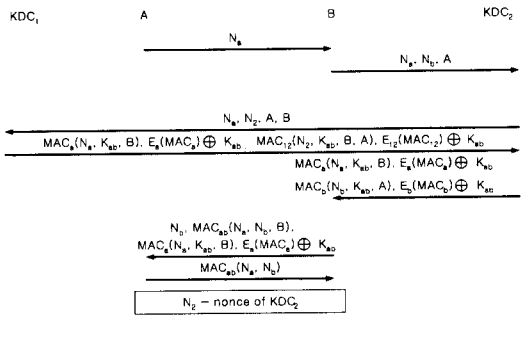


그림 3. A-B-K-K 도메인간의 프로토콜

Flow 1. A가 N_a(nonce A)에 기반한 인증을 수행하기 위해 B와 접촉한다.

Flow 2. B는 KDC에게 A의 ID와 N_a와 자신의 N_b를 전송한다. 여기서 B의 이름은 전송되는 메시지에서 추론될 수 있다고 가정한다.

Flow 3. KDC는 메시지 흐름 2를 수신하고, B의 KDC_2는 A가 외부 도메인에 속한 것을 발견하고 수신된 N_a와 자신의 N_2, 그리고 A와 B의 ID를 KDC_1로 직접 전송한다.

Flow 4. Flow 3을 수신한 KDC_1은 수신한 N_a, N_2, A, B의 ID를 이용하여 K_{ab}를 추출하여 MAC 계산을 수행한다. 또한, 새로운 키를 포함하는 티켓을 생성하여 KDC_2로 직접 전송한다.

Flow 5. KDC_2는 수신한 티켓을 단순히 B에게 전달하는 역할만을 수행한다.

Flow 6. B는 티켓을 검사하고 K_{at}를 추출하여 수신한 티켓을 A로 전송하여 A와 B의 통신이 가능함을 알린다.

Flow 7. 메시지를 수신한 A는 K_{ab}를 추출하여 MAC의 재계산을 통해 B로부터의 인증 코드가 정확한지 검사하고, 자신의 $MAC_{ab}(N_a, N_b)$를 B에 전송하여 키 분배 절차를 마친다.

이 때, KDC간의 메시지 흐름이 존재하지만 A와 B의 경우 A-B-K와 A-B-K-K 프로토콜이 달라지지는 않는다. 그림에서와 같이 KDC_1은 새로운 키를 포함하는 티켓을 생성하게 되고, KDC_2는 B에게 수신된 티켓을 전달하는 역할만을 수행한다. 이 때 KDC_2는 KDC_1과 통신하기 전에 B를 위한 티켓을 생성하거나, KDC_1의 응답을 받은 후에 티켓을 생성할 수 있다. 또한, KDC_2는 A, B, K_{ab}, N_a 등의 값을 저장해야 하며, KDC_1도 상태 변수를 저장해야 한다. 이처럼 A-B-K-K 모델은 작은 크기의 메시지를 사용할 수 있는 장점을 가진 반면에, KDC_2의 구현이 복잡한 단점을 가진다.

4. K-A-B Push 모델

Kerberos와 같은 시스템에서 사용되는 push 모델은 스테이션 A측과 B측이 안전한 통신을 수행하기 위해서 A측이 KDC에서 필요한 키를 구하는 프로토콜 모델로서 LAN과 같은 환경에 적합하다. 도메인간의 K-A-B push 모델에서도 이전 모델에서 사용된 가정이 그대로 사용된다. 또한, KDC가 외부와 통신하는 것은 보안상 문제가 발생할 수 있으므로 KDC가 외부 요소와 직접 통신하는 것은 허용되지 않는다. 도메인간의 A-B-K pull 모델에서 티켓의 생성 후에 인증이 이루어진 것과 달리 도메인간의

K-A-B push 모델의 경우에는 인증이 일어난 후에 티켓이 생성된다. 그림 4 에 도메인간의 K-A-B push 모델에서의 키 분배 과정이 잘 나타나 있다. K-A-B 프로토콜의 동작 절차는 다음과 같다.

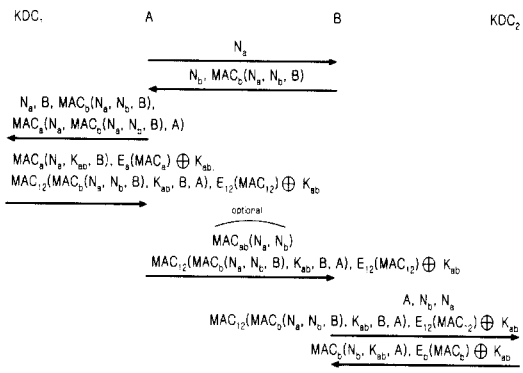


그림 4. K-A-B 도메인간의 프로토콜

Flow 1. A가 N_a(nonce A)에 기반한 인증을 수행하기 위해 B와 접촉한다.

Flow 2. B는 수신한 N_a와 자신의 N_b, 자신의 ID로 MAC 연산을 수행하고 자신의 N_b와 같이 A로 전송한다.

Flow 3. A는 수신한 N_b와 MAC 연산, 그리고, 메시지에 자신의 ID를 포함하여 KDC_1로 전송한다. 또한, B가 전송한 메시지를 이용하여 A가 MAC으로 재연산한 정보를 KDC_1로 전송한다.

Flow 4. KDC_1은 수신한 정보를 이용하여 티켓 K_{ab}를 생성한다. 여기서 KDC_1은 수신한 정보를 이용하여 인증을 수행한 후에 티켓을 생성한다. 또한, 생성된 티켓을 이용하여 MAC 연산을 수행하여 이를 A로 전송한다.

Flow 5. KDC_1로부터 메시지를 수신한 A는 수신 정보에서 티켓을 추출해 낸다. 그리고 KDC_1과 KDC_2의 MAC 연산을 B로 전송한다.

Flow 6. A로부터 메시지를 수신한 B는 A의 ID와 N_a와 자신의 N_b를 KDC_2로 전송한다. 또한, KDC_1과 KDC_2의 MAC 연산을 그대로 KDC_2로 전송한다.

Flow 7. KDC_2는 티켓을 검사하고 K_{at}를 추출하여, 티켓이 KDC_1에서 생성되었고, A에게 B와 통신이 가능함을 알린다. 이때, A의 ID와 추출한 K_{ab}를 이용하여 MAC 재연산을 수행하여 이를 B로 전송한다.

안전한 통신을 위해 사용되는 K-A-B 키 분배 절차는 A-B-K pull 모델과 거의 동일하다. 다만, 4번째 메시지의 전송에 의해서 KDC가 A와 B의 인증 토큰인 $MAC_b(N_a, N_b, B)$와 $MAC_a(N_a, MAC_t(N_a, N_b, B), A)$를 확인하게 되는 것이 다르다. 하지만, 어느 한 쪽의 토큰이 잘못되었다 하더라도, KDC는 어떤 토큰이 잘못되었는지 알려줄 수 없다. 즉, KDC는 단순히 토큰이 옳은지, 그른지 만을 알려주는 것이다. 이러한 잘못을 고치기 위해 3번째 메시지 전송 시에 $MAC_b(N_a, N_b, B)$가 포함되도록 확장할 수 있다.

또한, A-B-K 모델과의 차이점은 마지막 단계에서 다시 한번 인증을 수행하는 것이다. 이는, A와 B가 항상 새로운 키 K_{ab}를 갖고 있다는 보장을 할 수 없기 때문이다. 이러한 차이점은 새로운 키 K_{ab}를 이용하여 통신을 하는 경우에 문제가 발생할 수 있다. 따라서, 마지막 부분에 B와 KDC_2간에 직접적인 핸드셰이킹을 수행하여 보완한다. 이 K-A-B 모델은 인증이 일어난 후에 티켓을 생성하므로, 티켓을 생성한 후에 인증을 수행하는 A-B-K 모델과 대조를 이룬다.

5. K-K-A-B Push 모델

KDC 간의 통신을 지원하기 위한 K-K-A-B push 모델의 메시지 흐름이 그림 5에 나타나 있다. 제시된 K-K-A-B 모델의 동작 절차는 다음과 같다.

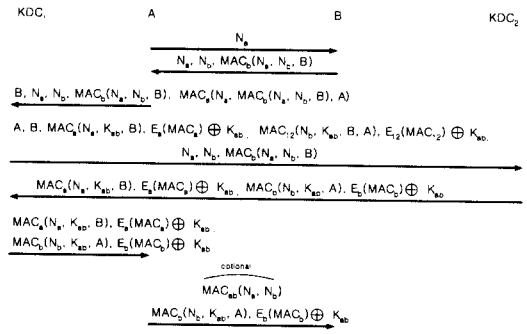


그림 5. K-K-A-B 도메인간의 프로토콜

Flow 1. A가 N_a(nonce A)에 기반한 인증을 수행하기 위해 B와 접촉한다.

Flow 2. B는 수신한 N_a와 자신의 N_b, 자신의 ID로 MAC 연산을 수행하고 자신의 N_b와 같이 A로 전송한다.

Flow 3. A는 수신한 N_b와 MAC 연산, 그리고,

메시지에 자신의 ID를 포함하여 KDC_1로 전송한다. 또한, B가 전송한 메시지를 이용하여 A가 MAC으로 재연산한 정보를 KDC_1로 전송한다.

Flow 4. 메시지를 수신한 KDC_1은 수신한 정보를 이용하여 인증을 수행하고 MAC 연산을 수행한다. 또한, K_{ab}를 생성하고 이를 이용하여 수신한 A의 ID, B의 ID, N_a, N_b의 MAC 연산을 수행하고 그 정보를 KDC_2로 직접 전송한다.

Flow 5. KDC_1로부터 메시지를 수신한 KDC_2는 수신 정보에서 티켓을 추출해 낸다. 그리고 추출한 티켓을 이용하여 인증을 수행하고 A와 B가 통신이 가능함을 KDC_1로 전송한다.

Flow 6. KDC_2로부터 통신 가능함을 통지받은 KDC_1는 그 정보를 A로 전송한다. 이때 발생된 K_{ab}를 이용하여 인증과정을 수행하고 A로 전송한다.

Flow 7. KDC_1로부터 통신 가능 메시지를 수신한 A는 수신정보로부터 티켓을 검사하고 K_{ab}를 추출한다. 또한, A는 B와의 통신이 가능함을 알리기 위한 정보를 B로 전송한다. 이때, 정보 전송시 인증절차를 수행한다.

Ⅳ. 성능분석 및 결과고찰

1. 시뮬레이션 파라메터

인증 및 키 분배 프로토콜에서 사용하는 MAC을 위한 해쉬함수와 암호화를 위한 암호화 알고리즘에 대한 시뮬레이션 파라메터는 다음과 같다[9].

- MAC(Message Authentication Code)을 위한 해쉬함수
  - MD5(Message Digest 5)
  - 6.12 Mbits/second의 처리능력
  - digest 길이 : 128bits
- 암호화 : DES
  - Key 길이 : 56bits
  - 4 Mbits/second의 처리능력
- 그외 스테이션 ID나 비표의 길이 : 48bits

암호적으로 안전한 해쉬 함수는 디지털 서명, 메시지 인증, 키 유도와 같은 분야에서 중요한 암호의 도구로 이용되고 있다. 현재까지 제안된 소프트웨어로 고속 수행이 가능한 해쉬 함수들의 대부분이 MD4의 설계 원리에 기반을 두고 있으며 이 MD4의 단점을 개선한 MD5를 MAC에서 사용하는 해쉬 함수로 이용하였다 이 해쉬 함수는 주로 128 비트의 비트스트링을 출력하는 함수이며 초당 처리되는 bit의 수는 6.12정도이다. 암호화로 사용된 DES 알고리즘에서 키 길이는 56bit이나 암호화를 위한 8bit를 추가하여 64비트의 블록 크기를 가진다.

2. 신호 구조

본 성능분석에서는 두 가지의 신호 방식에 대한 성능평가를 수행하였는데, 신호 채널 구조가 그림 6에 나타나 있다.

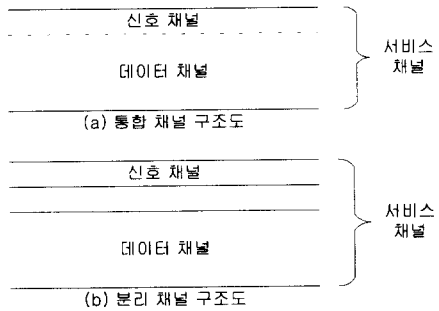


그림 6. 채널 구조도

첫 번째 채널 구조인 통합 채널 구조는 신호 채널과 데이터 채널이 전체 용량에서 가변적으로 사용되는 구조이다. 두 번째 채널 구조인 분리 채널 구조는 신호 채널이 전체 용량에서 고정적으로 할당되어 전송되는 채널 구조이다. 또한 채널특성에서 강우시에 나타나는 BER의 특성과 대기감쇠는 본 논문에서 고려하지 않았다.

일반적으로 VSAT 망을 구축할 때, Inband와 Outband의 채널 대역 비율을 참조로 하여 Inband와 Outband의 캐리어 수는 7:1의 비율로 구성하였으며, 지연이 큰 위성망을 고려하였기 때문에 왕복지연을 250ms로 가정하였다[10].

3. 트래픽 모델링

데이타 서비스의 모델링은 현재 신용카드 조회 업무와 관련된 트래픽 형태를 고려하여 메시지 길이가 평균 45byte이고 도착 프로세스가 poission 분포인 것으로 가정하였다.

신호 트래픽의 모델링은 각 채널당 전체 대역에서 고정적으로 1/8정도를 할당하였다. 두 스테이션과 하나의 KDC 간에 수행되는 MAC 연산의 총 파라메터 길이는 208비트이고, 하나의 스테이션과

KDC간에 수행되는 MAC 연산의 총 파라메터는 160비트이다. 암호화를 위한 DES 연산 처리 속도는 4Mbits/second로 가정하였으므로, 이를 위해 소요되는 시간을 0.03ms로 가정하였다. 에러환경은 1%FER을 고려하였으며, 신호트래픽은 즉각적인 검출과 재전송이 수행되는 것으로 가정하였다. 신호 메시지의 발생주기는 메시지 크기를 기반으로 신호 정보의 트래픽 밀도에 따라 poisson 분포를 따르면서 발생하도록 하였다.

4. 결과 분석

Pull 모델의 지연 특성이 그림 7에 나타나 있다. Pull 모델의 지연특성을 살펴보면, Inband방식의 지연은 트래픽 밀도가 증가하면 증가하는데 반하여, Outband방식의 지연은 고정적인 특성을 나타냄을 알 수 있다. 이는 Outband방식은 신호채널을 고정적으로 1/8 정도를 할당받아 사용하는 전용 채널이므로 데이터 트래픽 밀도의 영향을 받지 않기 때문이다. 또, 두 모델의 지연 차이는 A-B-K-K 모델이 A-B-K 모델보다 위성 링크를 두 번 더 지나게 되며 위성 링크로 직접 키를 분배 할 때 MAC 연산과 암호화 작업을 두 번 수행하기 때문에 나타남을 알 수 있다.

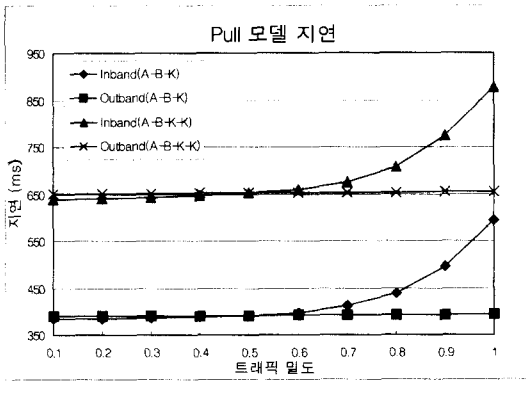


그림 7. Pull 모델의 지연특성

Pull 모델의 처리율 특성이 다음 그림 8에 나타나 있다. Pull 모델의 처리율 특성을 살펴보면, Outband 방식보다는 Inband방식의 성능이 더 우수함을 알 수 있다. 이는 Outband방식에서 신호 전용 대역폭 점유율은 1/8수준이지만, 데이터 트래픽이 공유하지 못하기 때문에, 처리율의 저하가 발생한다. 반면에, Inband방식에서는 신호 채널의 대역이 트래픽 전송을 위하여 공유가 되므로 Outband 방식보다 처리율이 향상됨을 알 수 있다.

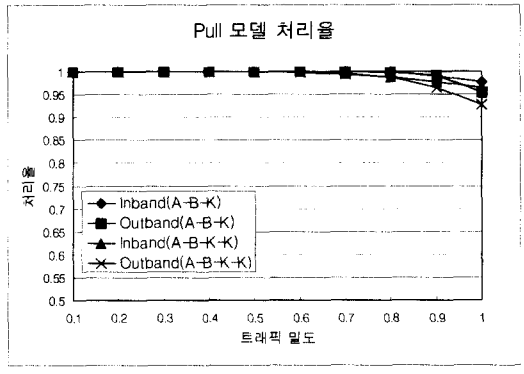


그림 9. Pull 모델의 처리율

Push 모델의 지연특성이 그림 9에 나타나 있다. 전체적인 지연 특성은 Pull 모델과 비슷한 특성을 나타내며 전체적인 지연은 Pull 모델보다 높은 수치를 나타낸다. 이는 프로토콜의 안전성을 위하여 마지막 단계에서 한번 더 인증 과정을 수행하기 때문이다. 또한, K-A-B 모델과 K-K-A-B 모델간의 지연 차이는 서로 다른 도메인에 위치한 키 분배 센터가 직접 통신하여 인증 및 키 분배 작업을 수행하기 때문이다. 즉, K-K-A-B 모델이 위성 링크를 더 경유하기 때문에 두 모델간의 지연 차이가 나타나게 된다.

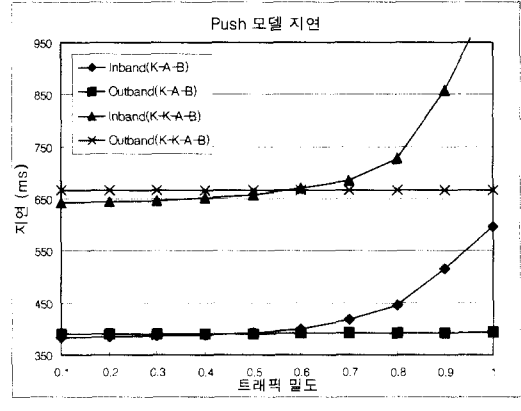


그림 10. Push 모델의 지연특성

Push 모델의 처리율 특성이 그림 10에 나타나 있다. Push 모델의 처리율은 전체적으로 Pull 모델보다 나쁨을 알 수 있다. 이는 마지막 키 분배 단계에서 마지막으로 한번 더 인증 과정을 수행하기 때문에 나타나는 지연 특성으로 인하여 처리율이 나빠짐을 알 수 있다. 또한, Outband 방식에서보다 Inband방식에서의 처리율이 우수한 것은 Outband방

식은 신호처리용으로 전체 대역의 1/8 정도로 고정되어 있기 때문에 채널 이용율의 한계가 있는 반면, Inband방식에서는 키 분배 프로토콜의 전송 이외에 남는 부분을 데이터 전송에 이용할 수 있기 때문에 채널의 처리효율이 우수하다.

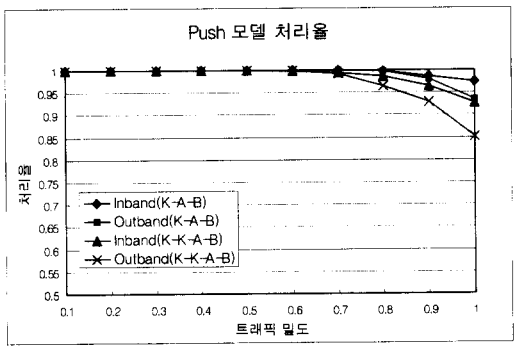


그림 12. Push 모델의 처리율

V. 결 론

본 논문에서는 키 분배 프로토콜을 위성망에 적용하였을 경우의 동작절차와 처리절차를 기술하였다. 또한, 키 분배 프로토콜의 처리절차를 위성망에서 Inband와 Outband방식으로 처리하였을 경우의 성능을 분석하였다.

성능분석 결과, Push 모델과 pull 모델에서 Inband방식에서의 지연특성은 트래픽 밀도가 증가함에 따라 급격히 증가하며, Outband방식에서의 지연특성은 거의 일정함을 알 수 있었다. 이는 Inband방식에서는 키 분배 프로토콜의 지연특성이 전송되는 데이터 양이 많음으로 인하여 지연이 증가하기 때문이다. 또한, Outband방식에서는 전송되는 데이터 양과 상관없으므로 일정한 지연을 유지함을 알 수 있었다. 처리율 특성을 살펴보면, 트래픽 밀도가 증가함에 따라 Inband방식에서는 키 분배 프로토콜의 처리율이 우수한 반면, Outband방식에서는 처리율이 나쁨을 알 수 있었다. 즉, Inband방식에서 처리되는 키 분배 프로토콜은 채널을 공유하여 사용하므로 처리율이 우수한 반면, Outband방식에서는 신호전용 채널을 공유하지 못하므로 데이터 처리율이 나쁨을 알 수 있었다.

참 고 문 헌

[1] B.C. Neuman and T.Ts'o, "Kerberos:An Authentication Service for Computer Networks", IEEE Comm. Mag., Sep. 1994

[2] Warwick Ford, Computer Communications Security, Prentice-Hall, 1995.

[3] Bruce Schineier, "Applied Cryptography", John Wiley & Sons, 1996.

[4] William Stallings, "Network and Internetwork Security", Prentice Hall, 1995.

[5] R.M. Needham and M.D. Schroeder, "Using Encryption for Authenticaion Large Networks of Computers", Comm. of ACM. Vol. 21, No. 12, pp. 993-999, Dec. 1978

[6] E.D. Dorothy and G.M. Sacco, "Timestamps in Key Distribution Protocols", Comm. of ACM, Vol. 24, No. 8, pp.533-536, Aug. 1981.

[7] D. Otway and O. Rees, "Efficient and Timely Mutual Authentication", Operation System Review, Vol. 21, No. 1, pp. 8-10, Jan. 1987

[8] E. Okamoto and K.Tanaka, "Identity-Based Information Security Management System for Personal Computer Networks", IEEE JSAC., Vol. 7, No. 2, pp. 290-294, Feb. 1989.

[9] 신상욱, 류대현, 이상진, 이경현, "MDx-계열 해쉬 함수에 기반한 새로운 해쉬 함수", 통신정보보호학회 논문지, 제 7 권, 제 4 호, pp 59-71, 12, 1997.

[10] 장대익 외 2인, "저속데이타 전송용 VSAT 통신시스템의 링크설계", 한국통신학회 논문지, Vol.19, No.7, pp.1213-1223, 7, 1994.

진 상 민(Sang-min Jin) 정회원

1993년 3월~1997년 2월 : 경희대학교 공과대학 전자계산공학과(학사)

1997년 3월~1999년 2월 : 경희대학교 공과대학 전자계산공학과(석사)

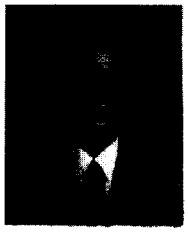
1999년 3월~현재 : (주)두루넷 연구1팀

<주관심 분야> 통신 프로토콜, 유무선 통신서비스, 유무선 멀티미디어 통신, 유무선 통신망관리

조 동 호(Dong-ho, Cho) 정회원

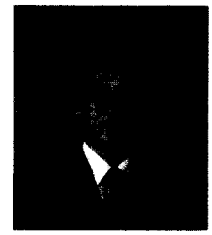
1979년 : 서울대학교 전자공학과(학사)
1981년 : 한국과학기술원 전기 및 전자공학과(석사)
1985년 : 한국과학기술원 전기 및 전자공학과(박사)
1985년~1987년 : 한국과학기술원 통신공학연구실 선임연구원
1987년~1998년 : 경희대학교 전자계산공학과 교수
1989년~1995년 : 경희대학교 전자계산소 소장
1998년~현재 : 한국과학기술원 전기 및 전자공학과 부교수
<주관심 분야> 유무선 통신 프로토콜,유무선통신서비스,유무선 멀티미디어 통신망

이 상 한(Sang-han, Lee) 정회원



1990년 3월~1994년 2월 : 경북대학교 전자공학과(학사)
1994년 3월~1997년 2월 : 경북대학교 전자공학과(석사)
1997년 3월~현재 : 국방과학연구소 연구원

강 건 우(Gen-woo, Kang) 정회원



1969년 3월~1973년 2월 : 연세대학교 공과대학 전자공학과(학사)
1973년 1월~1981년 11월 : 한국과학기술연구소 연구원
1983년 3월~1987년 2월 : 한국과학기술원 전기 및 전자공학과(석사)
1991년 3월~1996년 2월 : 한국과학기술원 정보및통신공학과(박사)
1981년 12월~현재 : 국방과학연구소 책임연구원