한국정보전자통신기술학회논문지(jkiiect)'16-04, Vol.9 No.2
논문 16-09-02-162 http://dx.doi.org/10.17661/jkiiect.2016.9.2.162

위성환경에서의 Timestamp 기반 키 교환 프로토콜

송인아, 이영석*

Timestamp based Key Exchange Protocol for Satellite Access Network

In-A Song, Young-seok Lee*

요 약 위성환경에서 키 교환 프로토콜은 안전한 암호화 통신을 하기 위해선 매우 중요하다. 그러나 ETSI 표준안의 키 교환 프로토콜은 Diffie-Hellman 방식을 사용하기 때문에 중간자 공격에 취약하다. 중간자 공격을 방지하기 위해서는 인증이 필수적인 과정이며 확실하고 안정적인 인증을 위해 인증서를 이용한 프로토콜이 제안되었지만 무선통신을 기반으로 하는 위성환경에서는 적합하지 않은 방식이다. 본 논문에서는 이러한 점을 개선하기 위하여 Timestamp 기반 키 교환 프로토콜을 제안하였다. 제안프로토콜은 Timestamp를 이용한 시간 값 계산으로 중간자 공격을 방지 할 수 있게 된다. 또한 성능분석 및 성능평가를 통하여 제안하는 프로토콜이 ETSI 표준안 프로토콜과 인증서 기반 프로토콜에 비해 메모리 사용량, 통신량 그리고 연산량에서 효율적임을 증명하였다.

Abstract The key exchange protocols are very important to provide the secure communication in broadband satellite access network. However key exchange protocol of ETSI(European Telecommunications Standards Institute) is vulnerable to man-in-the-middle-attack by using Diffie-Hellman algorithm. And the key exchange protocol using certification is not useful in satellite environment. We propose the key exchange protocol using Timestamp which have the resistant to man-in-the-middle-attack. Proposed protocol is able to prevent the man-in-the-middle-attack by calculated time value. Also showing experiment results, we prove that proposed protocol improve memory usage, communication amount and calculation amount than other protocols.

Key Words : Key Exchange, Diffie-Hellman, Timestamp, Man-in-the-Middle Attack, Protocol

1. 서론

현재 위성통신은 정보의 도청, 비정상적인 패킷의 전송, 메시지 재사용, 데이터 위 · 변조 등 공격에 쉽게 노출되어 있다. 공격을 방어하기 위해 위성통신은 내부 및 외부 네트워크 보안 정책을 갖고 있으며 이러한 보안정책들은 인증, 데이터의 기밀성, 가용성, 무결성, 부인방지 등의 보안 서비스를 제공할 수 있어야 한다.

위의 사항들을 충족하기 위해 사용하는 방법은 키를 이용한 데이터의 암 · 복호화 및 인증이다. 안전한 키 교환을 위해 ETSI (European Telecommunication Standards Institute) EN 301 790 표준에서 제시하는 키 교환 프로토콜과 이를 개선한 인증서 기반 프로토콜이 존재한다. 그림 1은 광대역 망을 위한 위성 참조 모델을 나타내고

This paper was performed with the support ICT development projects promising technologies of information and communication technology promotion center.(No.R6910-15-11-2)
*Corresponding Author : Dept. of Information & Telecommunication Engineering, Kunsan National University(leeys@kunsan.ac.kr)
Received April 12, 2016 Revised April 21, 2016 Accepted April 26, 2016

있다. NCC(Network Control Centre)는 망 관리 센터를 의미하며, RCST(Return Channel Satellite Terminal)은 사용자의 위성 단말을 의미한다. 키 교환 프로토콜은 NCC와 RCST 사이에서 수행된다. 그러나 기존에 존재하고 있는 키 교환 프로토콜들은 공격에 노출 될 위험을 지니고 있어 안전하다고 보기 어렵다.

국제 표준기관인 ETSI에서 표준화한 EN 301 790의 기본 키 교환 프로토콜은 Main Key Exchange(MKE)이다. 쿠키 기반의 사용자 인증을 통해 키 교환을 수행하게 된다[1].

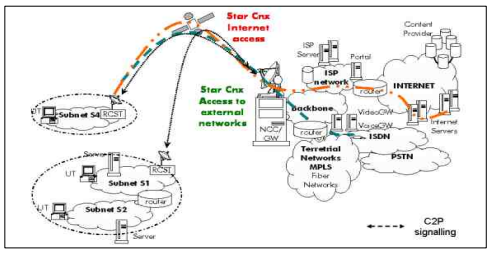

그림 1. 광대역 위성 참조 모델
Fig. 1. Satellite Broadband Reference Model

그러나 NCC와 RCST가 초반에 쿠키를 교환하는 시점에서 데이터 노출이 되기 때문에 중간자 공격이 가능하게 된다. 또한 데이터 전송 시 암호화를 하지 않기 때문에 재전송 공격도 가능하게 된다. 이러한 단점을 개선하여 인증서 기반의 키 교환 프로토콜을 제안하였다.[2] 인증서를 이용하기 때문에 확실한 사용자 인증이 가능하게 되지만 무선통신 기반의 위성통신에서는 부적합한 방법이다. 다른 인증 방법은 패스워드를 이용한 키 교환 프로토콜이다.[3] 그러나 위성환경의 NCC는 다수의 RCST를 관리해야하며 이는 NCC가 관리 하는 패스워드 자원 문제를 갖게 한다. 그리고 키 체인을 적용한 키 교환 프로토콜들도 있다. 해시 트리를 이용하여 메시지 인증 프로토콜[4]이나 시간의 흐름에 따라 키 체인을 이용하여 키를 생성하는 μTESLA을 적용한 키 교환 프로토콜[5]이 있지만 패스워드를 이용한 방식과 마찬가지로 NCC가 해쉬 트리 또는 키 체인을 관리하기에는 NCC의 자원 문제가 있으며 위성환경은 센서 네트워크와는 달리 여러 hop을 거쳐 데이터를 전송하지 않기 때문에 적용하기에는 부적절한 방식이다.

본 논문에서는 기존 프로토콜들의 단점을 개선하기 위해 Timestamp를 도입하여 중간자 공격을 방지하는 키 교환 프로토콜을 제안하였다. MKE 프로토콜과 마찬가지로 Diffie-Hellman 키 교환 방식을 이용하기 때문에 키 교환을 위해 NCC와 RCST에서의 공개 키 계산과정은 필수적이다. NCC는 데이터 송 · 수신 시 Timestamp를 생성하여 NCC와 RCST의 공개 키 계산 시간을 이용하여 중간자 공격을 확인 할 수 있게 된다. 또한 데이터 크기가 적은 Timestamp를 이용하기 때문에 기존 프로토콜들에 비해 자원 문제에서도 자유로워지게 된다.

본 논문의 2장에서는 기존 프로토콜들의 수행 방법과 단점을 살펴보고, 3장에서는 본 논문에서 제안 프로토콜을 기술한다. 4장에서는 기존 프로토콜들과 본 연구에서 제안한 프로토콜의 성능을 메모리 사용량, 연산량, 통신량의 3가지 측면에서 비교하고, 제안 프로토콜의 효율성에 대해 살펴본다. 5장에서는 성능 분석의 효율성을 분석하기 위해 시뮬레이션을 수행하고 성능 평가의 결과를 기술하고 6장에서 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

기존의 위성망 기반 키 교환 프로토콜은 여러 종류가 존재한다. ETSI 표준안인 EN 301 790에서는 MKE 프로토콜 뿐만 아니라 빠른 키 교환을 위해 데이터 교환 없이 인증만을 수행하는 Quick Key Exchange 프로토콜, NCC의 주도로 여러 RCST에게 공통의 키를 교환하는 Explicit Key Exchange 프로토콜이 있다. 또한 MKE 프로토콜을 개선한 인증서 기반의 키 교환 프로토콜, MTI 키 교환 프로토콜을 이용한 개선 프로토콜, 타원곡선 방식을 이용한 개선 프로토콜이 존재한

다.[2][6]

본 논문에서는 표준안인 MKE 프로토콜을 중점으로 중간자 공격을 막기 위한 제안 프로토콜을 제안하고 있으며, 이를 위해 본 장에서는 MKE 프로토콜과 이를 인증서 기반으로 개선한 인증서 기반 프로토콜을 살펴 보도록 한다.

2.1 Main Key Exchange (MKE)

MKE는 EN 301 790 표준에서의 기본 키 교환 프로토콜이다. NCC와 RCST 사이의 비밀 값 공유를 위해 Diffie-Hellman 알고리즘을 사용하며, RCST가 NCC에게 사용자 인증을 하기 위해 쿠키 값이 사용된다. 새로운 세션을 설정할 때 마다 수행되는 프로토콜이며, 쿠키 값 갱신을 위해 새로 만들어진 공유 값을 선택적으로 사용하게 된다. 마지막으로 페이로드 스트림 데이터 처리를 위한 암호문을 생성하기 위한 공유 비밀 키를 유도한다.

그림 2는 프로토콜의 수행 절차를 나타내고 있으며, '||'는 데이터의 연결, '(UC)x'는 x값을 unsigned char형으로의 형 변환과정, ' '는 empty string 그리고 nonce는 난수를 의미한다. 총 수행 단계는 간략하게 3단계로 살펴볼 수 있다.

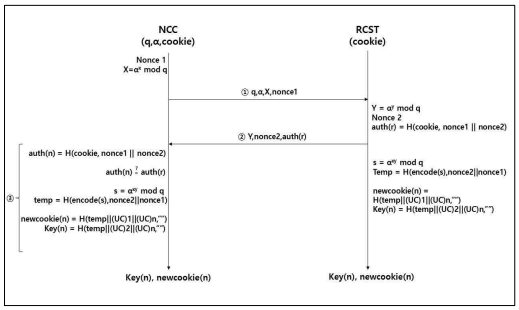

그림 2. Main Key Exchange Protocol
Fig. 2. Main Key Exchange Protocol

NCC에서 Diffie-Hellman 소수 a, q와 난수 nonce1 를 결정한 뒤 비밀 값 x를 이용하여 생성한 공개 값 X를 RCST에게 전달한다.

a, q, X, NCC의 난수 nonce1를 수신한 RCST는 자신의 비밀 값 y를 이용하여 공개 값 Y를 생성한 뒤 난수 nonce2를 생성한다. 수신하였던 NCC의 난수와 자신의 난수, 쿠키 값을 이용하여 인증 값 auth(r)을 생성한 뒤 NCC에게 자신의 난수 nonce2, 공개 값 Y, 인증 값 auth(r)을 NCC에게 전달한다. 그리고 NCC의 공개 값 X, 자신의 공개 값 Y를 이용하여 비밀 값 s를 생성 한 뒤 비밀 값 s와 NCC와 자신의 난수 값 nonce1, nonce2을 이용하여 temp를 생성한다. 생성한 temp값을 이용하여 쿠키의 갱신과 키 생성을 완료한다.

RCST에게 RCST의 난수 nonce2, 공개 값 Y, 인증 값 auth(r)을 수신한 NCC는 RCST와 같은 방식으로 인증 값 auth(n)을 생성 한 뒤 일치하는지 비교를 한다. 일치하면 올바른 터미널로 판단하여 RCST와 같은 방식으로 temp 생성 후 쿠키의 갱신과 키 생성을 완료한다.

2.2 인증서 기반 프로토콜

1 절에서 살펴본 MKE 프로토콜의 단점을 극복하기 위해서 인증서 기반 프로토콜을 제안하였다.[2] 인증서를 이용하여 장기간 공개 값을 사용하기 때문에 비밀의 신선도를 위해 난수를 도입하였다. 그리고 세션마다 변경되는 비밀의 키 확신성을 위하여 MKE 프로토콜과는 다르게 2차 인증 값을 이용하게 된다.

또한, 공격자가 공개 채널에서 임의의 난수를 알고 있다 가정하여도 비밀 값을 구하는 것은 DLP(Discrete Logarithm Problem)을 풀어내는 확률만큼 어렵다. 다음 그림 3은 MKE 기반 인증서 프로토콜의 수행 절차를 나타내고 있다.

h(n)은 NCC의 2차 인증 값, h(r)은 RCST의 2차 인증 값을 나타내며 Cert(N)은 NCC의 인증서, Cert(R)은 RCST의 인증서, rn은 NCC가 생성한 난수, rr은 RCST가 생성한 난수를 의미한다. 각각의 공개 값은 인증서 안에 포함되어 있다.

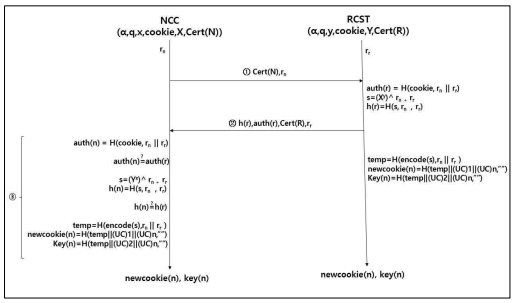

그림 3. 인증서 기반 프로토콜
Fig. 3. Certification based Protocol

NCC에서 난수 rn을 생성 한 뒤 인증서 Cert(N)와 함께 RCST에게 전달한다.

RCST는 난수 rr을 생성 한 뒤 수신한 NCC의 인증서 Cert(N)를 이용하여 공개 값 X을 획득한다. NCC의 난수 rn과 자신의 난수 rr 그리고 쿠키 값을 이용하여 인증 값 auth(r)을 생성한다. 그리고 난수와 공개 값을 이용하여 비밀 값 s를 생성 한 뒤 이를 이용하여 2차 인증 값 h(r)을 생성한다.

NCC에게 자신의 2차 인증 값 h(r)과 인증 값 auth(r), 자신의 인증서 Cert(R), 난수 rr을 전송 한 뒤 temp 값 생성 후 쿠키의 갱신과 키 생성 과정을 수행한다.

NCC는 인증 값 auth(n)을 생성 후 RCST의 인증 값 auth(r)를 이용하여 인증 과정을 진행한다. 인증 과정이 올바르면 비밀 값 s를 생성 한 뒤 2차 인증 값 h(n)을 생성한다. RCST에게 수신 한 h(r)과 자신의 2차 인증 값 h(n)을 비교하여 최종 인증 과정을 진행 한다. 올바른 터미널이라 판단되면 temp 값 생성 후 쿠키의 갱신과 키 생성 과정을 수행한다.

3. 제안 프로토콜

3.1 프로토콜 절차

ETSI 표준인 EN 301 790의 MKE 프로토콜은 중간자 공격과 재전송 공격에 취약하며 인증서 기반 프로토콜은 인증서를 이용하기 때문에 위성통신에서는 효율적이지 않은 단점이 있다. 이러한 단점들을 보완한 NCC와 RCST가 공유하고 있는 데이터가 없어도 공격을 확인 할 수 있으며 인증서를 사용했을 경우 보다 통신량 및 장치의 연산량이 적은 프로토콜을 제안한다.

비밀 키를 생성하기 위해 Diffie-Hellman 알고리즘을 사용하며 중간자 공격을 막기 위해 시간 값(Timestamp)을 적용하였다. 제한 프로토콜은 고정된 공개 값을 사용하지 않기 때문에 키의 신선도를 가질 수 있으며 인증 과정을 제공하기 때문에 키의 확신성도 보장된다. 제안 프로토콜은 다음과 같은 가정사항을 갖는다. 키 교환 프로토콜은 Diffie-Hellman 알고리즘을 사용한다.

Diffie-Hellman 알고리즘의 공개 값 연산시간은 NCC에서 SATELLITE 까지의 전송시간 보다 크다.모든 장치는 동일한 성능을 갖는다. 표 1은 제안 프로토콜에서 사용되는 시스템 파라미터를 보여준다. 그림 4는 제안 프로토콜의 흐름도를 나타내며 총 8단계의 과정으로 수행된다. NCC에서 Diffie-Hellman 소수 a,q 를 결정한 뒤 시간 값 T0를 생성한다. 비밀 값 x과 소수 a,q를 이용하여 공개값 X를 계산한 뒤 NCC는 위성에게 소수 a, q, 공개 값 X를 전송하고 시간 값 T1를 생성한다. 공개 값 연산시간 TC를 위해 T1-T0=TC 를 계산한다. 위성은 a, q, X를 수신후, RCST에게 전달한다. 위성에서 a, q, X를 수신한 시간 값 T2를 계산한 뒤 NCC에게 전송한다. NCC는 시간 값 T2를 수신하고 T2 수신 시간 확인을 위해 시간 값 T3를 생성한다. NCC와 위성사이의 Round Trip 시간(Tr) 을 확인하기 위해 T3-T1 = Tr 를 계산한 뒤 Tr/2 = TN-S를 구한다.

표 1. 프로토콜 파라미터
Table 1. Protocol Parameters

Notation	Descriptions
a, q	Prime numbers for Diffie-Hellman operation
x	Private value of NCC
y	Private value of RCST

X	Public value of NCC
Y	Public value of RCST
S	Secret key created by NCC and RCST
T_c	Computation time to create public value X of NCC
T_r	Round-trip time between NCC and Satellite
T_{N-S}	Transmission time from NCC to Satellite
T_T	Round-trip time between NCC and RCST
T_H	Round-trip threshold time between NCC and RCST
T_0	Time value before public value computation of NCC
T_1	Time value after public value computation of NCC
T_2	Time value when satellite receivers values of NCC
T_3	Time value when NCC receives time-stamp of satellite
T_4	Time value when NCC receives public value Y of RCST

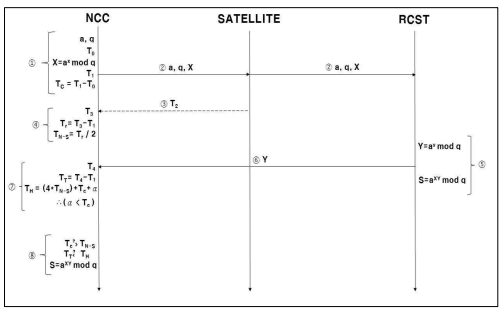

그림 4. 제안 프로토콜
Fig. 4. Proposed Protocol

a, q, X 를 수신한 RCST는 자신의 비밀값 y와 소수 a , q를 이용하여 공개 값 Y를 생성한다. 공개 값 Y를 위성에게 전송한 뒤 수신하였던 NCC의 공개 값 X와 자신의 공개 값 Y를 이용하여 비밀 키 S를 계산한다.

RCST에게 공개 값 Y를 수신한 위성은 NCC에게 전달한다.

공개값 Y를 수신한 NCC는 수신시간 확인을 위해 시간 값 T4를 생성한 뒤, NCC와 RCST 사이의 Round Trip 시간 확인을 위해 T4-T1 = TT를 계산한다. NCC와 RCST 사이의 Round Trip Threshold 시간을 구하기 위해 (4*TN-S)+TC+α = TH 을 계산한다. 이때 α 변수시간으로서 공개값 계산시간 (TC)보다 크게 설정하면 안된다.

중간자 공격을 확인하기 위해 NCC는 시간 값을 이용하여 판단하게 된다. 첫 번째로 NCC와 위성 사이에서의 중간자 공격을 확인하기 위해 공개값 계산시간 TC 가 NCC와 위성 사이의 전송시간 TN-S 보다 큰지 확인한다. (TC > TN-S) 만약 전송시간 TN-S가 공개 값 계산시간 TC보다 클 경우 중간자 공격이 수행되었다 판단하여 프로토콜을 수행을 중단한다.

첫 번째 단계가 참일 경우 두 번째 단계로 위성과 RCST 사이에서의 중간자 공격을 확인하기 위해 NCC와 RCST 사이의 Round Trip 시간 TT가 Round Trip Threshold 시간 TH 보다 작은지 확인한다. (TT < TH) 만약 TT가 TH보다 클 경우 중간자 공격이 수행되었다 판단하여 프로토콜을 수행을 중단한다.

첫 번째와 두 번째 판단과정이 모두 참이면 NCC는 자신의 공개 값 X와 RCST의 공개 값 Y를 이용하여 비밀 키 S를 계산한다.

3.2 안전성 분석

3.2.1 중간자 공격

Diffie-Hellman 알고리즘의 큰 단점은 중간자 공격이 가능하다는 점이다. 이를 해결하기 위해 대부분 인증서를 적용하지만 위성환경에서는 적합하지 않다. 광대역 위성 엑세스 망에서의 중간자 공격은 NCC와 위성 사이의 중간자 공격과 RCST와 위성 사이의 중간자 공격 2가지로 나눌 수 있다. 제안 프로토콜은 중간자 공격을 방지 하기 위해 Timestamp을 도입하고 이를 이용하여 2단계의 판단을 수행한다.

첫 번째 판단 (TC > TN-S)으로 NCC와 위성 사이의 중간자 공격을 확인할 수 있으며 두 번째 판단 (TT < TH)으로 RCST와 위성 사이의 중간

자 공격을 확인 할 수 있다. 이를 흐름도를 통하여 제안프로토콜의 중간자 공격 면역성을 살펴본다. 다음 표2는 중간자 공격 흐름도를 위한 추가 파라미터를 나타낸다.

표 2. 중간자 공격 흐름도를 위한 추가 파라미터
Table 2. Additional Parameters for Man-in-the-middle attack flow diagram

Notation	Descriptions
b. p	Prime numbers generated by Man-in-the-middle
M	Public value created from b, p
N	Public value created from a, q
S_{NM}	Secret Key created by NCC and Man-in-the-middle
S_{MR}	Secret Key created by RCST and Man-in-the-middle
T_M	Time value when Man-in-the-middle modify the time value T_2

그림 5는 NCC와 위성 사이의 중간자 공격의 흐름도를 나타낸다. 총 13단계의 과정으로 수행된다.

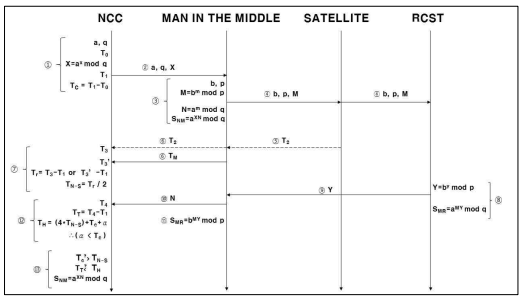

그림 5. NCC와 위성 사이의 중간자 공격
Fig. 5. Man-in-the-middle attack between NCC and Satellite

NCC에서 Diffie-Hellman 소수 a,q 를 결정한 뒤 시간 값 T0를 생성한다. 비밀 값 x과 소수 a,q를 이용하여 공개값 X를 계산한 뒤 위성에게 소수 a, q, 공개 값 X를 전송하고 시간 값 T1를 생성한다. 공개 값 연산시간 TC를 위해 T1-T0=TC를 계산한다.

중간자는 NCC가 보내는 소수 a, q, 공개 값 X를 가로챈다.

중간자는 RCST와의 Diffie-Hellman 과정을 위한 소수 b,p를 결정한 뒤 자신의 비밀 값 m을 이용하여 RCST를 위한 공개값 M을 생성한다. 중간자는 NCC로 위장하여 소수 b, p, 공개 값 M을 RCST에게 전송한다. 전송한 후 NCC의 소수 a, q를 이용하여 NCC와의 비밀 키 생성을 위한 공개 값 N 계산한 뒤 공개 값 N과 NCC의 공개 값을 X 이용하여 비밀키 SNM을 생성한다.

위성은 b, p, M를 수신하고 RCST에게 전달한다.

위성에서 b, p, M 을 수신한 시간 값 T2를 계산한 뒤 NCC로 위장한 중간자에게 전송한다.

위성에서 보낸 시간 값 T2를 수신한 중간자는 그대로 NCC에게 T2를 전송하거나, 자신에게 상황을 유리하게 하기 위한 시간 값 T2을 수정한 시간 값 TM을 전송한다.

위성에서 보낸 시간 값 T2나 중간자가 수정한 시간 값 TM 수신한 NCC는 수신 시간 확인을 위해 시간 값 T3 또는 T3' 을 생성한다. NCC와 위성사이의 Round Trip 시간 (Tr)을 확인하기 위해 T3 - T1 = TP or T3' - T1 = TP 를 계산한 뒤 Tr / 2 = TN-S 를 구한다.

b, p, M 를 수신한 RCST는 자신의 비밀 값 y와 소수 b , p를 이용하여 공개값 Y를 생성한다. 공개 값 Y를 위성에게 전송한 뒤 수신하였던 공개 값 M와 자신의 공개 값 Y를 이용하여 비밀키 SMR를 계산한다. SMR는 중간자와 RCST 사이의 비밀 키이지만 RCST는 NCC와의 비밀 키로 오해하고 있는 상태이다.

RCST에게 공개 값 Y를 수신한 위성은 NCC로 위장한 중간자에게 전달한다.

공개 값 Y를 수신한 중간자는 이전 과정에서 생성하였던 NCC와 비밀 키 생성을 위한 공개값 N을 NCC에게 전송한다.

중간자는 NCC 에게 공개 값 N을 전송한 뒤 RCST에게 받은 공개값 Y와 자신이 RCST와 비밀 키 생성을 위한 공개 값 M을 이용하여 RCST

와의 비밀 키 SMR을 생성한다.

공개값 N를 수신한 NCC는 수신시간 확인을 위해 시간 값 T4를 생성한 뒤, NCC와 RCST 사이의 Round Trip 시간 확인을 위해 T4-T1 = TT를 계산한다. NCC와 RCST 사이의 Round Trip Threshold 시간을 구하기 위해 (4*TN-S)+TC+α = TH 을 계산한다. 이때 α 변수시간으로서 공개값 계산시간 (TC)보다 크게 설정하면 안된다.

중간자 공격을 확인하기 위해 첫 번째 판단 단계인 공개 값 계산시간 TC 가 NCC와 위성 사이의 전송시간 TN-S 보다 큰지 확인한다. (TC > TN-S) 중간자는 (3)과정에서 RCST와의 비밀 키를 생성하기 위한 공개 값 M 계산 시간이 추가되어 TN-S의 시간이 지연되기 때문에 공개 값 계산 시간 TC는 NCC와 위성 사이의 전송시간 TN-S보다 작게 된다. 이로 인하여 NCC는 중간자 공격이 있다 판단하고 프로토콜 수행을 중단한다.

3.2.2 재전송 공격

NCC와 RCST가 교환하는 소수 및 공개 값을 획득하여 재전송을 시도 할 수 있다. 이를 방지하기 위한 방법은 데이터 송신자를 확인하는 방법이다. 제안프로토콜에서 데이터 송신자 확인을 위해 데이터 전송 시 NCC와 RCST가 식별 할 수 있는 자신의 ID를 첨부하도록 수정한다면 재전송 공격을 방지 할 수 있다.

4. 성능 평가

제안하는 프로토콜과 이전 프로토콜들의 효율성을 분석하기 위해 시뮬레이션을 수행하였다. 시뮬레이션은 NS-2을 이용하여 수행하였다. 환경 옵션은 NS-2 위성 시뮬레이션 설정 시 많이 사용하는 옵션으로 사용하였으며 위성 채널은 기본 제공 Channel/Sat을 따라 설정하였으며 Uplink와 Downlink의 대역폭은 2Mb, Queue의 형식은 Droptail 방식을 사용하였다.

NCC와 RCST의 역할을 위해 2개의 터미널을 설정하였으며, 하나의 터미널은 서울의 위치로 설정하여 위도 33.5°, 경도 126.9°, 다른 터미널은 LA의 위치인 위도 33.7°, 경도 84.4°로 설정하였다.

전송 계층(Transport Layer)는 TCP로 설정하였으며, 응용 계층 프로토콜은 CBR(Constant Bit Rate) 트래픽 타입으로 설정하였다. MAC과 Link 계층 사이에 수신경로에 대한 에러 모델을 추가하였으며, 위성환경에서 많이 사용되는 설정 값으로 추가하였다. 각각의 키 교환 프로토콜은 총 4단계로 네트워크 전송이 이루어지게 된다. 2개의 터미널에서 데이터 전송 시 중간에 정지 위성을 경유하기 때문에 src 터미널에서 위성, 위성에서 dst 터미널, dst 터미널에서 위성, 위성에서 src 터미널로 수행 된다.

표 5는 시뮬레이션에서 사용한 파라미터의 크기를 나타낸다. 파라미터의 크기는 ETSI 방식과 일반적으로 많이 사용하는 방식으로 결정하였다.

표 3. 파라미터의 크기
Table 3. Size of parameters

		Key size (bit)	Output (bit)	Method
Key exchange	Diffie-Hellman	512	512	ETSI
Hash	HMAC-SHA1		160	ETSI
Random number	Pseudo		64	ETSI
Timestamp			32	General

메모리 사용량과 통신량, 연산량 비교 그래프는 표 3, 표 4, 표 5를 참고하여 수행된 결과를 보여준다. 성능 비교 그래프의 내용은 다음과 같다.

NCC의 메모리 사용량에 대한 그래프는 시간에 흐름에 상관없이 비슷한 메모리 사용량을 보여준다. 그림과 같이 MKE 프로토콜의 메모리 사용량은 Diffie-Hellman 알고리즘에서 키 생성을 위해 사용하는 소수 값과 이를 이용하여 생성하는 NCC와 RCST의 공개 값, 새로운 쿠키 값과 비밀 값을 저장하기 때문에 약 330Byte로 보여진다. 인

증서 기반 프로토콜은 인증서에 따른 추가 메모리 사용량으로 인하여 약 460Byte로 보여지며 제안 프로토콜의 경우 MKE 프로토콜과는 다르게 소수 값과 공개 값, 데이터의 크기가 작은 Timestamp를 이용하기 때문에 약 230Byte로 제일 작게 보여진다.

통신량은 1000초까지 프로토콜을 수행 했을 경우 MKE 프로토콜은 Diffie-Hellman 과정을 위한 소수 값과 NCC와 RCST의 공개 값으로 인하여 약 172 KB로 나타나며 인증서 기반 프로토콜은 공개 값 전송 시 인증서를 이용하여 전송하기 때문에 약 196 KB로 제일 크게 나타난다. 그러나 제안 프로토콜의 경우 패킷의 크기가 작은 Timestamp의 통신이 대부분이므로 제일 작은 약 113 KB로 나타난다.

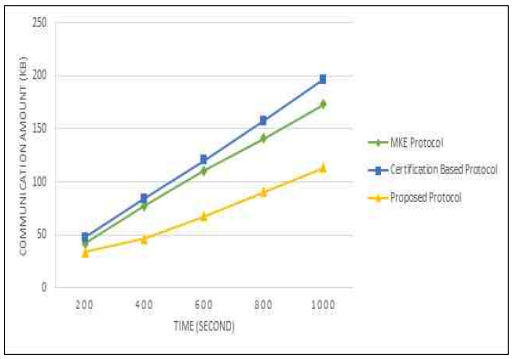

그림 9. 각 프로토콜의 통신량
Fig. 9. Communication amount in Protocols

연산량은 MKE 프로토콜과 인증서 기반 프로토콜이 처음에는 비슷한 양을 보이다 시간이 지날수록 인증서 기반 프로토콜의 연산량이 많아 지는 것을 확인할 수 있다. 제안프로토콜의 경우 다른 프로토콜과는 달리 해쉬 연산이나 인증서에 대한 추가 연산이 없으므로 제일 적은 연산량을 보인다.

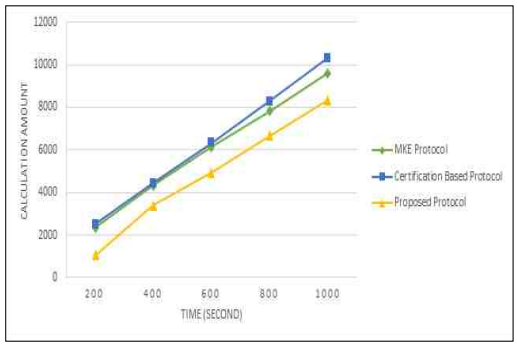

그림 10. 각 프로토콜의 연산량
Fig. 10. Calculation amount in Protocols

6. 결론

위성망에서 사용하는 보안정책들은 인증, 데이터의 기밀성, 가용성, 무결성, 부인방지 등의 보안 서비스를 제공할 수 있어야 하며 이를 충족하기 위해 키를 이용한 데이터의 암 · 복호화 및 인증 방식을 사용한다. 안전한 키 교환을 위해 ETSI 표준 EN 301 790에서는 MKE 프로토콜을 제시하였지만 NCC와 RCST가 초반에 쿠키를 교환하는 시점에서 데이터 노출이 되기 때문에 중간자 공격에 취약하다.

중간자 공격을 방지 하기 위해 인증서를 이용한 인증서 기반 프로토콜을 제안하였지만 무선통신 기반의 위성환경에서는 사용하기에 적합하지 않다. 또한 패스워드를 이용한 키 교환 프로토콜, 해시 트리를 이용하여 메시지 인증하는 프로토콜 그리고 μTESLA 방식을 적용한 키 교환 프로토콜이 있지만 이러한 프로토콜을 적용하게 되면 다수의 RCST를 관리하는 NCC는 많은 데이터를 관리해야 하기 때문에 자원 문제가 발생하게 된다. 또한 위성환경은 여러 hop을 거쳐 데이터를 전송하지 않기 때문에 키 체인을 이용하는 키 교환하는 방식은 적용하기 어렵다.

본 논문에서는 Timestamp를 이용하여 중간자 공격을 방지 할 수 있는 프로토콜을 제안하였으며 성능 분석 및 성능 평가를 통해 기존 프로토콜과 비교하여 제안프로토콜의 자원 관리 및 통신자원

의 효율성을 보였다. 안전성 분석을 통해 제안프로토콜의 중간자 공격 방지 과정을 살펴 보았으며 ID 파라미터 추가로 재전송 공격도 방지 할 수 있게 된다. 제안 프로토콜을 운용 환경에 맞게 적용한다면 기존의 NCC와 RCST를 이용하는 위성통신에서 효율적인 프로토콜이 될 것으로 기대된다.

REFERENCES

[1] ETSI, EN 301 790 V1.5.1(2009), Retrieved July., 25, 2015, from http://www.etsi.org

[2] H. R. Oh and H. Y. Youm, "Key exchange protocols for domestic broadband satellite access network," Korea Institute of Information Security and Cryptology(KIISC), vol. 14, no. 3, pp. 13-25, June. 2004.

[3] M. K. Park, E. S. Cho and T. K. Kwon, "Multi server password authenticated key exchange using attribute-based encryption," Journal of Korean Institute of Communication Sciences(KICS), vol. 40, no. 8, pp. 1597-1605, Aug. 2015.

[4] B. H. Lee, S. B. Lee, J. Y. Moon and J. H. Lee, "Lightweight DTLS message authentication based on a hash tree," Journal of Korean Institute of Communication Sciences(KICS), vol. 40. no. 10, pp. 1969-1975, Oct. 2015.

[5] J. C. Choi, J. I. Kang, D. H. Nyang and K. H. Lee, "Operation μTESLA based on variable key-slot in multi-hop unattended wsn," Journal of Korean Institute of Communication Sciences(KICS), vol. 39, no. 3, pp. 223-233, March. 2014.

[6] Zhong Yantao and Ma Jianfeng, "A highly secure identity-based authenticated key-exchange protocol for satellite communication," Journal of communications and networks, vol. 12, no. 6, pp. 592-599, Dec. 2010.

저자약력

송 인 아(In-A Song) **[학생회원]**

- 2015년 2월 : 군산대학교 정보통신공학과 졸업
- 2015년 3월 ~ 현재 : 군산대학교 전자정보공학부 석사과정

<관심분야> 네트워크 보안, 인터넷 프로토콜

이 영 석(Young-seok Lee) **[종신회원]**

- 1992년 2월 : 충남대학교 컴퓨터공학과(학사)
- 1994년 2월 : 충남대학교 컴퓨터공학과(석사)
- 2002년 2월 : 충남대학교 컴퓨터공학과(박사)
- 2002년 3월 ~ 2004년 8월 : 한국전자통신연구원 선임연구원
- 2004년 9월 ~ 현재 : 군산대학교 컴퓨터정보통신공학부 교수

<관심분야> 정보보호, 사물인터넷, 이동컴퓨팅