종단 간 암호화 통신을 위한 키 전달 프로토콜에 관한 연구[1)]

김정윤*, 황인용**, 이종언**, 김석중**, 이유신**, 최형기*
*성균관대학교 정보통신공학부
**삼성탈레스 종합연구소/SIAT
steal83@ece.skku.ac.kr, inyong08.hwang@samsung.com, jong-eon.lee@samsung.com,
seokjoong.kim@samsung.com, yous.lee@samsung.com, hkchoi@ece.skku.ac.kr

A Study on Key Delivery Protocols for End-to-End Secure Communication[1)]

Jung-Yoon Kim*, In-Yong Hwang**, Jong-Eon Lee**, Seok-Joong Kim**,
You-Shin Lee**, Hyoung-Kee Choi*
*School of Information and Communication Engineering, Sungkyunkwan University
**R&D Center, Samsung Thales Co., Ltd.

요 약

All-IP는 통신에 관련된 모든 개체가 IP를 사용하는 네트워크를 의미한다. All-IP 네트워크에서는 보안을 위해 통신 내용에 대한 암호화가 반드시 이루어져야 하며, 신뢰할 수 있는 3자 (trusted third party)는 서비스 중재 및 부가 서비스 제공을 위해 통신에 사용된 암호화 키를 획득할 수 있어야 한다. 이는 통신 개체가 다른 개체에게 암호화 키를 전달하는 메커니즘이 필요하다는 것을 의미한다. 우리는 본 논문을 통해, 통신 개체가 다른 개체에게 키를 전달하는 기법을 3가지로 분류하고 각 기법에 대해 상세히 설명한다. 또한, 우리는 새로운 키 전달 프로토콜을 제안하고 3가지 운용 모드를 제시한다. 우리가 제안하는 프로토콜은 사용자의 필요에 따라 보안 기능을 선택적으로 운용할 수 있다. 성능평가 및 분석 결과에 따르면, 보안 기능과 성능은 뚜렷한 트레이드오프(trade-off) 관계가 있음을 알 수 있고, 기본적인 기능만을 운용할 경우 추가적인 오버헤드가 거의 없음을 알 수 있다.

키워드 : Network security, Network protocol, Key delivery, End-to-end security

I. 서론

최근 네트워크의 성능을 향상시키기 위한 연구가 활발히 진행됨에 따라, 네트워크에 존재하는 모든 개체들이 IP를 이용하여 통신에 참여하는 all-IP 네트워크가 주목받게 되었다. All-IP 네트워크는 통신의 효율성을 향상시킬 뿐 아니라, 통신의 안정성, 나아가서는 호환성 및 확장성을 모두 제공하는 차세대 네트워크의 이상적인 구조이다. 현재 연구가 활발히 진행중인 4세대 네트워크는 대표적인 all-IP 네트워크로서, 4세대 네트워크에 존재하는 모든 개체들은 IP를 사용하여 인터넷에 연결될 것이다.

한편, all-IP 네트워크는 모든 개체가 개방형 네트워크인 인터넷에 접속되기 때문에, 각종 위협들로부터 통신 내용이 보호되어야 한다. All-IP 네트워크에서는 효율적으로 보안을 제공하기 위해, 종단 간 보안을 적용하는 것이 필요하다 [1].

한편, 네트워크를 안전하게 보호하기 위해서는, 1) 통신을 수행하는 개체 간의 상호 인증, 2) 통신 내용의 암호화, 3) 통신 내용의 무결성 보장, 4) 통신 내용에 따른 차등화 된 접근 제어가 기본적으로 제공되어야 한다. 추가적으로, 통신을 수행하는 두 개체가, 신뢰하는 3자로부터 부가 서비스를 제공받거나, 추후 법적 분쟁 발생 시 중재자로서의 역할을 요청하기 위해, 5) 암호화 키를 전달하는 기능이 제공되어야 한다. 예를 들어, 미국의 경우 Communications Assistance for Law Enforcement Act (CALEA) 라는 법안에 의해 모든 Voice over IP (VoIP) 시스템은 신뢰할 수 있는 3자가 암호화 키를 획득할 수 있도록 설계되어야 한다. 이는 지적재산 보호, 산업 스파이 차단 등 다양한 목적에 의한 것이

※ "이 논문은 2008년도 삼성탈레스(주)의 재원을 지원 받아 수행된 연구임."

며, 영국, 네덜란드 등 유럽 국가에서도 유사 법안을 시행하고 있다.

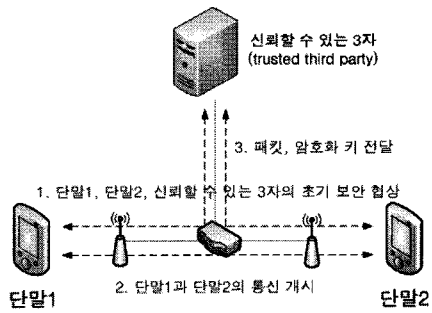
그러나 위에서 제시한 네트워크 보안의 요구사항을 모두 만족하는 표준 프로토콜은 아직까지 존재하지 않는다. 일반적인 응용에서는 개인정보 문제로 인해 암호화 키를 신뢰할 수 있는 3자에게도 전달하지 않기 때문이다. 그러나 법적 분쟁이 발생할 수 있는 응용이나, 미국의 VoIP 시스템과 같은 응용에서는 다양한 문제로 인해 신뢰할 수 있는 3자의 암호화 키 획득이 필수적이고, 이를 제공하는 프로토콜에 관한 연구가 반드시 진행되어야 한다. 현재까지는 신뢰할 수 있는 3자에게 암호화 키를 전달하는 기법에 관한 연구가 거의 진행되지 않았기 때문에, 본 논문에서는 관련 연구를 다루지 않는다.

우리는 본 논문을 통해, 신뢰할 수 있는 3자에게 암호화 키를 전달하는 기법을 3가지로 분류하고, 이를 위한 새로운 프로토콜을 제안한다. 제안하는 프로토콜은 상위 프로토콜과 독립적으로 동작하도록 구성함으로써, 어떠한 응용을 사용하더라도 암호화 키 전달이 가능하도록 설계하였다. 그리고 키 전달 메시지에 대한 선택적인 보안 적용이 가능하도록 함으로써, 제안하는 프로토콜에 의한 오버헤드를 최소화 할 수 있도록 유연성을 제공한다.

본 논문의 이후 구성은 다음과 같다: 2장에서는 암호화 키 전달 기법을 분류하고 설명한다. 3장에서는 새로운 키 전달 프로토콜을 제안하며, 4장에서는 제안하는 프로토콜의 성능을 평가한다. 끝으로 5장에서는 향후 연구과제를 제시하며 결론을 내린다.

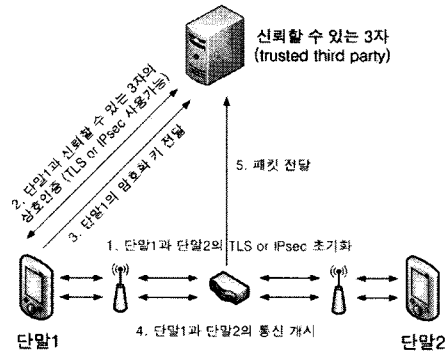
II. 암호화 키 전달 기법

우리는 통신 주체가 신뢰할 수 있는 3자에게 암호화 키를 전달하는 기법을 3가지로 분류하였다. 우리가 분류한 3가지 키 전달 기법은 각각 키 사전 유도 방식, 키 사전 전달 방식, 키 실시간 전달 방식이다.



〈그림 1〉 키 사전 유도 방식

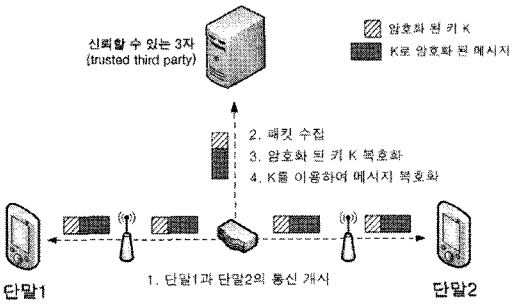
〈그림 1〉은 키 사전 유도 방식을 나타낸다. 이 방식은 보안을 제공하기 위해 초기 협상 과정 (handshake)을 거치는 프로토콜에 적용할 수 있으며, 초기 협상 과정을 거치는 대표적인 보안 프로토콜에는 IPsec (IP security) [2], TLS (Transport Layer Security) [3]가 있다. 이 프로토콜들의 경우, 통신 주체들은 실제 통신을 수행하기에 앞서, 보안을 제공하기 위해 암호화 및 메시지 무결성에 관련된 정보를 협상하는 과정을 먼저 수행한다. 이 때, 신뢰할 수 있는 3자도 이 협상 과정에 함께 참여하여, 암호화 및 메시지 무결성에 관련된 정보를 함께 협상하게 되면, 자연스럽게 암호화 키를 신뢰할 수 있는 3자도 획득할 수 있게 된다. 키 사전 유도 방식의 장점은 신뢰할 수 있는 3자가 초기 협상 과정에 참여하여, 해당 통신에 대한 키 전달의 안정성을 더욱 향상시킬 수 있다는 점이다. 즉, 초기 협상 과정에서 문제가 발생할 경우 통신 자체가 진행되지 않기 때문에, 정상적으로 개시되는 통신에서 사용되는 암호화 키는 100% 획득이 가능하다. 반면, 키 사전 유도 방식의 단점은 IPsec, TLS 등의 보안 프로토콜을 그대로 사용할 수 없고 반드시 수정해야 한다는 점이다. 또한, 프로토콜의 초기 협상 과정을 수행하기 위해 통신 주체들 뿐 아니라 신뢰할 수 있는 3자도 수많은 정보들을 송수신하기 때문에, 불필요한 오버헤드를 유발할 수 있다.



〈그림 2〉 키 사전 전달 방식

〈그림 2〉는 키 사전 전달 방식을 나타낸다. 키 사전 전달 방식은 키 사전 유도 방식과 달리, 신뢰할 수 있는 3자가 통신 주체들의 초기 협상 과정에 개입하지 않는다. 또한, 통신 주체들이 초기 협상 과정을 거치는 프로토콜을 사용하지 않더라도 암호화 키 획득이 가능하다. 키 사전 전달 방식은, 통신 주체들이 어떠한 방식으로든 키를 생성하고 나면, 이후에 신뢰할 수 있는 3자에게 해당 키를 전달하는 방법이다. 통신 주체들은 키 생성 후, 통신 개시와 동시에 암호화 된 키를 신뢰할 수 있는 3자에게 1회 전달하게 되며, 이를 수신한 신뢰할 수 있는 3자는 통신 주체에 대한 정보와 함께 해당 키를 저장한다. 키 사전 전달 방식의 장점은 키 사전 유도 방식에 비해 오버헤드가 작다는 점이며, 통신 주체들이 어떠한 방식으로 키를 생성하는지 상관없이 암호화 키를 획득할 수 있다는 점이다. 그러나, 신뢰할 수 있는 3자에게 키를 전달하는 시점에서 네트워크에 문제가 발생하면, 키 획득이 지연될 수 있다는 단점이 있고, 통신 주체가 송신한 키가 손실될 가능성이 존재한다. 뿐만 아니라, 다수의 통신이 동시에 진행될 경

우, 각각의 통신 및 통신 주체에 대한 정보, 그리고 키에 대한 정보를 모두 저장해야 하며, 이는 신뢰할 수 있는 3자의 저장 공간에 대한 오버헤드를 유발한다. 또한, 다수의 키 정보가 저장될 경우, 추후에 키 정보를 검색하는데 소요되는 시간이 증가할 수 있다. 데이터베이스를 이용해 키 정보를 효율적으로 관리하더라도, 통신의 개수가 늘어날 경우 데이터베이스의 규모가 커지면서 키 검색에 소요되는 시간도 크게 증가할 수 있다.

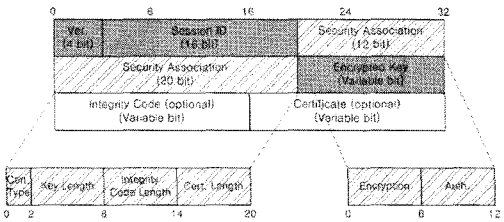


〈그림 3〉 키 실시간 전달 방식

〈그림 3〉은 키 실시간 전달 방식을 나타낸다. 키 실시간 전달 방식은 송신되는 모든 메시지에 키를 탑재하여 송신하는 방식으로, 해당 키는 신뢰할 수 있는 3자의 공개 키 (혹은 신뢰할 수 있는 3자와 통신 주체가 공유하고 있는 비밀 키)를 이용하여 암호화 되어 있다. 메시지를 수신한 신뢰할 수 있는 3자는 메시지와 함께 탑재되어 있는 암호화 된 키를 복호화하여 키를 획득할 수 있다. 키 실시간 전달 방식은 키 사전 유도 방식과 키 사전 전달 방식의 장점을 모두 가지고 있다. 즉, 정상적으로 개시된 통신에 대해서는 100% 키 획득이 가능하며, 통신 주체들이 어떠한 방식으로 키를 생성하든 상관없이 키 획득이 가능하다. 그러나 키 실시간 전달 방식은 송신 오버헤드가 크다는 단점을 가지고 있기 때문에, 키 획득의 대상을 제한할 필요가 있다.

III. 제안하는 키 전달 프로토콜

본 장에서는 우리가 제안하는 키 전달 프로토콜에 대해 설명한다. 제안하는 프로토콜은 키 실시간 전달 방식을 사용하며, 응용 계층에서 동작한다. 우리는 제안하는 프로토콜을 Real-time Key Transport Protocol (RKTP) 이라고 정의하였다. RKTP의 메시지 헤더 구조는 〈그림 4〉와 같다. 〈그림 4〉를 보면, RKTP 헤더는 Ver. 필드, Session ID 필드, Security Association 필드, Encrypted Key 필드, Integrity Code 필드, 그리고 Certificate 필드로 분류된다. 〈표 1〉은 각 필드에 대한 요약 설명을 나타낸다.



〈그림 4〉 제안하는 키 전달 프로토콜 (RKTP)의 메시지 헤더 구조

〈표 1〉 제안하는 키 전달 프로토콜 (RKTP)의 메시지 헤더 설명

필드 명칭	크기 (bit)	내용
Ver.	4	RKTP 버전 표기
Session ID	16	각 세션의 구분자 (identifier)
Security Association	32	암호화 키의 전달에 사용되는 알고리즘 정보 Encryption: 암호화 알고리즘 종류 Auth.: 메시지 인증 코드의 알고리즘 종류 Cert. Type: 인증서 종류 Key Length: 암호화 키의 크기 Integrity Code Length: 메시지 인증 코드 크기 Cert. Length: 인증서의 크기
Encrypted Key	가변	암호화 된 키
Integrity Code	가변	키에 대한 메시지 인증 코드 혹은 전자서명
Certificate	가변	전자서명 사용 시, 서명자의 인증서

Ver. 필드는 4비트의 크기로 RKTP의 버전을 나타내며, 향후 RKTP가 확장되거나 기능이 향상될 경우, 버전을 표기함으로써 이전 버전과의 구분이 가능하고 하위 버전에 대한 호환성을 제공할 수 있다. 본 논문에서 제안하는 프로토콜은 RKTP 1.0 이므로 Ver. 에는 1이 표기된다.

Session ID 필드는 해당 패킷이 어떤 세션에 포함된 패킷인지 구분하는 역할을 수행한다. RKTP 1.0 에서는 최대 2^{16} 개의 세션을 표기할 수 있으며, 동시에 존재하는 세션의 개수가 이를 초과할 경우 프로토콜의 동작에 문제가 발생할 수 있다. RKTP의 향후 버전에서는 Session ID의 크기를 늘리거나, 혹은 확장 헤더를 도입함으로써 이러한 문제를 해결할 수 있다.

Security Association 필드는, 1) 현재 세션에서 사용중인 암호화 알고리즘 종류, 메시지 인증 코드의 알고리즘 종류, 인증서의 종류를 나타내고, 2) 암호화 된 키의 크기, 메시지 인증 코드의 크기, 인증서의 크기를 나타낸다. Security Association 필드의 크기는 총 32비트이다.

Security Association 필드 내의 Encryption 필드에는 키의 암호화에 사용된 알고리즘을 표기한다. 사용 가능한 알고리즘 종류에는 3-DES, AES-128, AES-256, RSA-1024 등 매우 다양하며, 2^6-1개의 범위 내에서 사용자가 임의로 추가할 수 있다. 이 필드를 0으로 설정하면, 암호화 알고리즘을 사용하지 않는 경우를 나타낸다.

Security Association 필드 내의 Auth. 필드에는 키에 대한 메시지 무결성 제공에 사용된 알고리즘을 표기한다. 사

용 가능한 메시지 인증 코드 알고리즘의 종류는 SHA-1, SHA-512, MD5, RSA-1024, DSA-2048 등 매우 다양하다. 암호화 알고리즘의 종류와 마찬가지로, 메시지 인증 코드 알고리즘의 종류도 사용자가 2^6-1개의 범위 내에서 임의로 추가할 수 있다. 또한, 이 필드를 0으로 설정하면, 메시지 인증 코드를 사용하지 않는 경우를 나타낸다.

Security Association 필드 내의 Cert. Type 필드에는 사용된 인증서의 종류를 표기한다. 사용 가능한 인증서의 종류는 X.509 등이 있다. 일반적으로 사용되는 인증서의 종류는 매우 적기 때문에, 2비트의 공간만을 사용한다.

한편, Security Association 필드 내의 Key Length 필드 (6비트), Integrity Code Length 필드 (6비트), Cert. Length 필드 (6비트)는 각각 암호화 된 키의 크기, 메시지 인증 코드의 크기, 인증서의 크기를 나타낸다. 각각의 필드는 32비트 단위로 크기를 표현한다. 예를 들어, Key Length 필드가 8이면 암호화 된 키의 크기가 256비트라는 것을 나타낸다. 만약 각 필드에 0이 표기되어 있다면, 이는 최대 크기 (2048비트)를 나타낸다. 암호화 알고리즘이나 메시지 인증 코드의 알고리즘을 사용하지 않을 경우, 이는 Encryption 필드 및 Auth. 필드를 0으로 설정하여 나타낼 수 있으며, 이 때 각 Length 필드는 어떤 값으로 설정해도 무방하다. RKTP 1.0 에서는 이 경우 각 Length 필드를 0으로 설정한다.

Encrypted Key 필드는 사용된 암호화 알고리즘에 따라 가변적인 크기를 가지며, 이 필드에는 메시지 암호화에 사용된 키가 암호화되어 있다. 이 키를 암호화할 때 사용한 알고리즘은 Encryption 필드에 표기되어 있다.

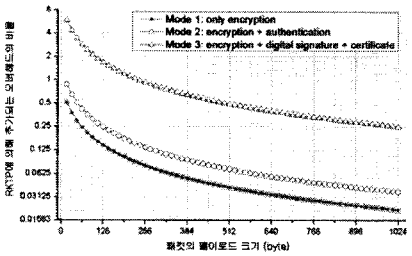
Integrity Code 필드는 필요 시 선택적으로 사용하는 필드로서, 전달하는 키에 대한 메시지 인증 코드를 나타내며, 이 때 사용한 알고리즘은 Auth. 필드에 표기된다. 이 알고리즘에 따라 Integrity Code 필드의 크기가 결정된다.

Certificate 필드는, 전달하는 키에 대한 메시지 인증 코드의 알고리즘이 RSA와 같이 인증서를 필요로 하는 알고리즘 (전자서명)인 경우, 메시지 송신자의 공개키에 대한 인증서를 첨부함으로써 전자서명에 대한 유효성을 제공할 수 있다. 이 때 첨부한 인증서의 종류는 Cert. Type 필드에 표기되어 있다. Certificate 필드는 Integrity 필드와 마찬가지로 필요에 따라 선택적으로 사용할 수 있으며, 사용하는 인증서의 종류에 따라 크기가 결정된다.

IV. 성능평가 및 분석

본 장에서는 제안하는 키 전달 프로토콜의 통신 오버헤드를 1) 메시지 무결성을 제공하지 않는 경우, 2) 대칭 키 기반의 메시지 인증 코드를 제공하는 경우, 3) 전자서명 및 인증서를 제공하는 경우로 나누어 분석한다. 한편, 모든 경우에 대해, 암호화 된 키와 대칭 키 기반의 메시지 인증 코드의 크기는 각각 128비트로 가정하고, 전자서명과 인증서의 크기는 각각 1024비트로 가정한다. 그리고 네트워크 계층과 전송 계층 프로토콜은 각각 IPv4와 UDP를 가정한다.

〈그림 5〉는 위에서 설명한 3가지 경우의 통신 오버헤드를 비교한 결과이다. 〈그림 5〉를 보면, 사용하는 보안 기능의 종류를 줄일 경우 오버헤드가 감소하는 것을 알 수 있다. 따라서 사용자는 상황에 따라 보안 기능과 성능 중 필요한 것을 선택하여 키 전달 과정을 수행할 수 있다. 예를 들어, 패킷의 페이로드가 1024바이트일 때, 암호화 된 키만 전달하고 키에 대한 메시지 무결성을 제공하지 않는 경우, RKTP에 의해 추가되는 패킷의 크기는 약 2%에 불과하므로, 통신의 성능에는 거의 영향을 미치지 않는다.



〈그림 5〉 운용 모드 별 RKTP에 의해 추가되는 패킷 크기의 비율

V. 결론

일반적인 응용 서비스와 달리, 법적 분쟁이 발생할 수 있는 응용이나, 미국의 VoIP 시스템과 같은 응용에서는 다양한 문제로 인해 신뢰할 수 있는 3자의 암호화 키 획득이 필수적이고, 이를 위해서는 통신 개체가 신뢰할 수 있는 3자에게 키를 전달하는 메커니즘이 필요하다.

우리는 신뢰할 수 있는 3자에게 키를 전달하는 방식을 키 사전 유도 방식, 키 사전 전달 방식, 키 실시간 전달 방식으로 분류하였고, 각 기법의 장단점을 설명하였다. 그리고 우리는 키 실시간 전달 방식에 해당되는 새로운 프로토콜을 제안하였다. 제안한 프로토콜은 어떠한 응용을 사용하더라도 키 전달이 가능하도록 설계하였다. 또한, 선택적인 보안 적용을 통해, 오버헤드를 줄일 수 있도록 구성하였다.

향후 연구과제는 키 사전 유도 방식 및 키 사전 전달 방식의 프로토콜을 개발하는 것이다. 그리고 각종 상황에 적합한 키 전달 프로토콜을 제시함으로써, all-IP 네트워크의 보안을 한층 향상시키는 기반을 마련할 수 있을 것이다.

참고문헌

[1] J. Y. Kim et al,, "Performance Evaluation of End-to-End Security Protocols in IEEE 802.16e," Accepted in The 8th Applications and Principles of Information Science, 2009.

[2] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, Nov. 1998.

[3] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, Aug. 2008.