Low-Rate WPAN 환경에서 효율적이고 인증 가능한 비밀키 설정 메카니즘

허 준° 홍충선 김동규

경희대학교 컴퓨터공학과

{heojoon°, cshong, dkkim}@khu.ac.kr

Efficient and Authenticated Secret Key Agreement Mechanism in Low-Rate WPAN Environment

Joon Heo°, Choong Seon Hong, Dong Kyu Kim

Dept. of Computer Engineering, Kyung Hee University

요 약

현재 IEEE 802.15.4(Low-Rate WPAN)를 구성하는 디바이스들은 파워, 메모리 용량, 연산처리 등의 물리적 제약으로 인해 보안을 위한 공유키 생성은 대칭키 기반의 메카니즘을 사용하고 있다. 그러나 대칭키 기반 메카니즘은 키 관리, 키 분배, 디바이스의 이동성지원과 같은 부분에 문제점을 가지게 된다. 또한 LR-WPAN 환경은 해당 어플리케이션에 따라 차별된 보안 정책을 가질 수 있으며, 디바이스의 물리적 상태에 따라 보안강도를 결정하는 경우도 발생할 수 있다. 본 논문에서는 타원곡선 기반의 EC-MQV 알고리즘을 사용하여 공개키 기반의 공유키 생성 메카니즘을 제안한다. 또한, 코디네이터를 활용하여 각 디바이스의 연산을 최소화하여 디바이스의 에너지 소비를 줄이는 방법을 제시하고, 보안 레벨에 따른 키 생성에 관하여 설명한다.

1. 서 론

산업기기의 자동 제어, 홈 오토메이션, 조명 시스템의 통합적 제어와 같은 어플리케이션들은 센서들의 상호작용에 의한 종합적인 제어 시스템의 구축을 필요로 하며 이러한 사용자 요구를 충족할 수 있는 기술로 IEEE 802.15.4 표준이 제안되었다. 이 표준은 low-rate, low-cost, short range의 특징을 가지며 디바이스들은 파워, 메모리 용량, 연산처리 능력 등과 같은 물리적 제약을 가지게 된다[1]. 이러한 물리적 제약으로 인해 보안을 위한 공유키 생성 메카니즘은 대칭키 기반의 시스템에 초점을 맞추어 진행되고 있는데, 그 이유는 일반적인 공개키 기반 기술은 키의 길이와 연산의 오버헤드가 크기 때문이다. 그러나, 대칭키 시스템이 가지는 키 관리, 키 분배, 구현의 복잡성 등의 문제점은 궁극적으로 공개키 기반 시스템의 적용을 요구하고 있으며, 이 문제를 해결하기 위한 다양한 관점에서의 노력이 이루어지고 있다[2]. 그 중에서도 EC-DH, EC-MQV와 같은 타원 곡선 기반의 공개키 기술의 적용이 중요한 이슈로 부각되고 있으나, 네트워크를 구성하는 디바이스의 상황과 어플리케이션의 특징에 따라 활용할 수 있는 키 설정 메카니즘의 개발이 미비하고 기존 알고리즘을 그대로 적용할 경우 디바이스의 물리적 제약을 해결하는데 충분하지 못하다. 따라서 본 논문에서는 EC-MQV 알고리즘을 기반으로 효율적이고 상호 인증 가능한 새로운 메카니즘을 제안한다. 이 메카니즘은 공개키 시스템의 특징을 사용하고 디바이스의 인증과 오버헤드를 최소화하기 위해 보안 메니져 (Security Manager)를 사용한다. 또한, 네트워크를 구성하는 디바이스의 물리적 상태와 해당 어플리케이션의 보안 정책에 따라 두 가지 보안레벨로 나누어 각각의 메카니즘을 정의한다. 제안된 메카니즘의 성능평가를 위해 먼저 타원곡선 암호화 알고리즘(ECC)을 기반으로 DH(Diffie-Hellman)와 MQV(Menezes-Qu-Vanstone)의 연산을 현재의 대칭키 시스템과 비교하여 디바이스가 가지게 되는 오버헤드를 측정한다. 또한 디바이스의 연산, 프레임 전송을 위한 네트워크 시간, 프레임 암호화에 따른 추가적 연산 등을 종합적으로 비교한다.

본 논문은 다음과 같이 구성되었다. 2장에서는 관련연구로서 타원곡선 암호와 알고리즘을 사용하는 EC-DH, EC-MQV에 관하여 살펴본다. 3장에서는 보안 레벨에 따른 공개키 기반 키설정 메카니즘을 제안하고 세부 동작과 메시지에 관하여 기술한다. 4장에서는 ECC기반의 알고리즘과 대칭키 기반 알고리즘을 비교하고, 제안된 공유키 설정 메카니즘을 기존의 메카니즘 및 대칭키 시스템과 상호비교한다. 5장에서는 결론과 향후 과제에 관하여 언급한다.

2.1 Elliptic Curve Menezes-Qu-Vanstone(EC-MQV)

EC-MQV는 개인키와 상대방의 공개키만을 이용하는 EC-DH와는 달리 Ephemeral Private Key 와 Ephemeral Public Key가 추가적으로 사용되어 EC-DH에서 취약점으로 지적된 Man-in-the-Middle 공격을 해결할 수 있다. 두 디바이스간에 Elliptic Curve coefficients값 E와 Base point P 를 동일하게 설정한 후 통신을 하고자 하는 디바이스1이 자신의 개인키 d와 공개키 Q=(a,b)를 생성하고, 디바이스2에게 자신의 공개키 Q=(a,b) 를 전송한다. 디바이스1로부터 공개키 Q=(a,b) 와 함께 연결요청을 받은 디바이스2도 개인키 d'와 공개키 Q'=(a',b') 를 생성하여, 디바이스2에게 공개키 Q'=(a',b')를 전송한다. 디바이스2에게 공개키 Q=(a,b)를 전송한 이후 디바이스1 은 Ephemeral 개인키 k를 Elliptic Curve coefficients 값과 Base point으로 주어진 타원곡선 상에서 임으로 선택하고 이를 이용해 Ephemeral 공개키

This work was supported by MIC

R=(x,y)을 생성하여 디바이스2 에게 R=(x,y)을 전송한다. 마찬가지로 디바이스2도 자신의 Ephemeral 개인키 k'를 선택한 후 생성한 공개키 R'=(x',y')을 디바이스1에게 전송한다. 이후 디바이스1은 개인키 d, 공개키 Q=(a,b), Ephemeral 개인키 k, Ephemeral 공개키 R=(x,y), 상대방의 공개키 Q'=(a',b'), 상대방의 Ephemeral 공개키 R'=(x',y')을 사용하여 타원곡선상의 연산을 수행한다. 또한 디바이스2도 자신의 개인키 d', 공개키 Q'=(a',b'), Ephemeral 개인키 k', Ephemeral 공개키 R'=(x',y'), 상대방의 공개키 Q=(a,b), 상대방의 Ephemeral 공개키 R=(x,y)을 사용해 타원곡선상의 연산을 수행한다. 이와 같은 과정으로 두 디바이스 간에 서로 다른 값을 통해 만들어낸 동일한 값(Shared secret Key)를 만들게 된다 [3][4][5].

3. 제안된 메카니즘

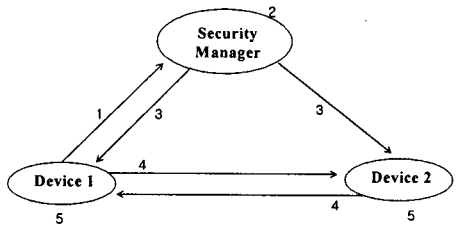
앞서 설명한 EC-DH, EC-MQV의 경우 공유키 생성과정이 디바이스-대-디바이스 형식이다. 다시 말해, 공개키 생성, 키관리, 임시키 생성 및 전송과 같은 과정을 모두 디바이스가 처리해야 한다. LR-WPAN 환경에 이러한 메카니즘을 그대로 적용할 경우 다음과 같은 문제가 발생하게 된다. 첫째, 디바이스간의 인증에 관한 문제이다. 디바이스들은 네트워크안의 모든 디바이스들과의 인증을 위한 정보를 가지고 있지 않다. 따라서, 특정 디바이스가 공유 보안키 설정을 요구할 경우 그 디바이스가 인증된 디바이스인지 여부를 확인할 수가 없다. 공개키 기반의 시스템에서는 두 디바이스가 동일한 키를 사용하지 않으므로, 디바이스 상호간의 인증을 보증할 수 있는 절차가 필요하다. 둘째, 타원곡선 기반의 알고리즘을 적용하기 위해서는 모든 디바이스들이 공통적으로 사용할 파라미터 값을 관리할 필요가 있다. 따라서 반복적으로 수행되는 값들은 믿을 수 있는 관리자에게 등록하고, 키설정을 원하는 디바이스가 관리자로부터 공개된 키를 받도록 하는 시스템이 효율적이라고 할 수 있다.

3.1 보안 메니저(Security Manager)를 통한 공개키 관리

본 논문에서는 보안 메니저 (Security Manager)를 정의하여 사용한다. 보안 메니저의 요구 기능은 IEEE 802.15.4 표준에서 정의하는 디바이스 중 코디네이터(coordinator)에 의해서 수행된다[1]. 보안 메니저는 해당 네트워크의 타원곡선 알고리즘을 위한 공통 파라미터를 생성하고 새롭게 네트워크로 들어오는 디바이스들에게 이 값을 알려주어 공개키를 생성한 후 자신에게 등록하도록 한다. 뒤에서 좀 더 자세히 설명하게 될 보안 정책에 따라 공개키와 임시 공개키 값을 등록받아 관리한다. 또한, 인가 받은 디바이스만이 보안 메니저와의 초기 연결 설정과정을 통해 파라미터 값을 받을 수 있다고 가정한다. 디바이스와 보안 메니저 간의 등록 과정이 끝나면 크게 두 시나리오로 동작한다. 보안이 많이 요구되고, 이에 적합한 물리적 능력을 가지는 디바이스의 High-Security 레벨 시나리오와 적은 수준의 보안이 요구되고 디바이스들의 연산능력이 작은 디바이스는 Middle-Security 레벨의 시나리오로 Shared Secret Key를 생성한다.

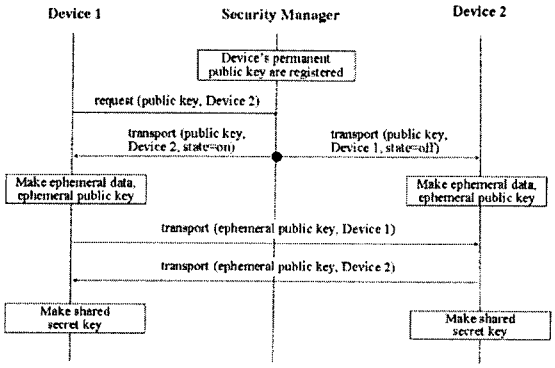
3.2 High-Security 레벨

네트워크 초기 구성 단계에서 위와 같은 방법으로 공개키가 등록된 후 High-Security 레벨에서의 공유 비밀키 생성 시나리오는 그림1과 같다.



<그림1> High-Security 레벨의 키 설정과정

a. 디바이스2와의 공유 비밀키 생성을 원하는 디바이스1은 SM에게 디바이스2의 공개키를 요청한다.
b. SM은 디바이스1과 디바이스2의 공개키를 테이블 목록에서 찾고 결과가 없는 경우 해당 디바이스와 공개키 등록을 위한 절차를 수행한다.
c. 디바이스1에게는 디바이스2의 공개키와 먼저 디바이스2에게 통신요청을 하라는 State=On 플래그를 보내고, 디바이스2에게는 디바이스1의 공개키와 통신요청을 기다리라는 State=Off 플래그를 전송한다.
d. State=On를 받은 디바이스1은 자신의 Ephemeral 임시 데이터를 생성하여 Ephemeral 공개키를 디바이스2에게 보내고 이 패킷을 받은 디바이스2는 디바이스1에게 같은 과정으로 Ephemeral 공개키를 보낸다.
e. 상대방의 공개키와 Ephemeral 공개키를 받은 각 디바이스는 Shared Secret Key를 생성한다.
위의 각 과정과 이에 따른 결과는 <그림2>와 같다.



<그림2> High-Security 레벨의 키 설정 과정

3.3 Middle-Security 레벨

Middle-Security level 모드의 시나리오는 아래의 설명과 같다.
a. 디바이스2와의 공유 비밀키 생성을 원하는 디바이스1

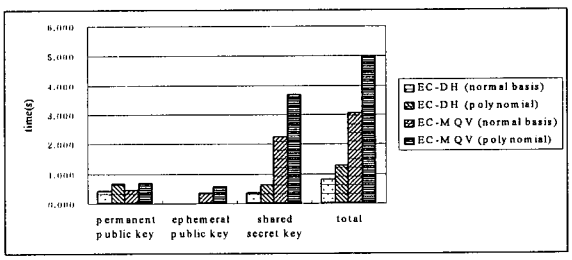
은 SM에게 디바이스2의 공개키와 임시 공개키를 요청한다.
b. SM은 디바이스1과 디바이스2의 공개키와 임시 공개키를 테이블 목록에서 찾고 결과가 없는 경우 해당 디바이스와 공개키 등록을 위한 절차를 수행한다.
c. 디바이스1에게는 디바이스2의 공개키, 임시 공개키와 먼저 디바이스2에게 통신요청을 하라는 State=On 플래그를 디바이스2에게는 디바이스1의 공개키, 임시 공개키와 통신요청을 기다리라는 State=Off플래그를 전송한다.
d. 상대방의 공개키와 Ephemeral 공개키를 받은 각 디바이스는 Shared Secret Key를 생성한다.

4. 성능평가

위 시나리오로 기존 EC-MQV를 C코드와 소켓을 이용해 구현하고 CPU: Pentium 200Mhz, 메모리: Ram 64 Mb, O/S: Windows 98 환경에서 각 메카니즘의 연산 시간을 측정하고 기존 대칭키 방식과 비교하였다.
우선 EC-DH의 normal basis(n)와 polynomial basis(p), EC-MQV의 normal basis 와 polynomial basis 그리고 Symmetric Key Agreement(SKA)에서 각각 공유 비밀키를 위한 생성 시간을 측정하면 그 결과는 <표 1>,<그림 3>와 같다.

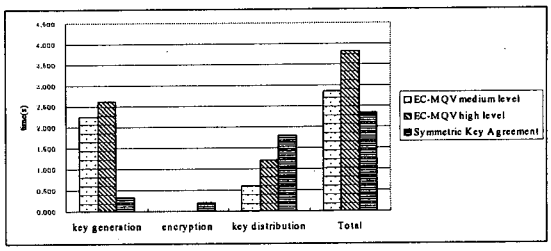
<표 1>EC-DH, EC-MQV, SKA 비교표 (단위: sec)

	EC-DH (n)	EC-DH (p)	EC-MQV (n)	EC-MQV (p)	SKA	
Private Key	0.000	0.000	0.000	0.000	Link Key	0.020
Public Key	0.426	0.667	0.444	0.703	Mac Tag1	0.122
ephemeral Key	-	-	0.000	0.000	Mac Tag2	0.180
ephemeral data	-	-	0.372	0.584	-	-
Share Key	0.372	0.628	2.246	3.680	-	-
Total Time	0.798	1.295	3.062	4.967	Total Time	0.322



<그림 3> EC-DH, EC-MQV, SKA 비교 (단위:sec)

또한, 250kbps 미만의 통신속도를 가지는 LR-WPAN의 구성된 네트워크를 통해 메시지 및 프레임 전송시간을 감안하여 전송 지연율 300ms를 가정하여 연산소요시간 + 네트워크 소요시간 을 계산해 보면 아래 <그림 4>과 같다.



<그림 4> 키 생성 및 분배에 소요되는 시간비교

위의 결과에서 EC-MQV는 공개키 알고리즘의 장점을 가지고 그 안정성도 입증이 되었다는 것을 전제하에 기존 대칭키 방식과 근소한 차이를 보이는 것을 확인할 수 있으며 향후 EC알고리즘 자체의 연산 경량화와 제안된 메카니즘을 디바이스에 최적화된 알고리즘으로 개선해 실제 LR-WPAN 환경이나 기타 모바일/센서네트워크에서 사용될 수 있도록 테스트를 진행할 것이다.

5. 결론 및 향후과제

본 논문에서는 대칭키 정도의 키 사이즈, 파라미터, 연산 과정을 통해서도 공개키 시스템과 같은 보안강도를 제공할 수 있는 EC-MQV 알고리즘을 기반으로 노드의 현재 상황과 해당 네트워크의 보안 정책에 따라 적용될 수 있는 2개의 보안 레벨을 정의하고 각각의 메카니즘를 제안하였다. 또한 이러한 시나리오를 바탕으로 키를 생성하고 대칭키 시스템과 보안 레벨의 따른 성능평가를 진행하였다. 향후 과제로는 실제 무선 및 센서 노드에 본 논문에서 제안한 메카니즘을 구현하고 구체적인(메모리, 연산시간, 네트워크 지연) 성능평가를 통한 메카니즘의 검증이 필요하다고 할 수 있겠으며, 연산의 고속화 방안에 초점을 두어 연구를 진행하는 것이다.

참고문헌

[1] Jose A. Gutierrez, Edgar H. Callaway Jr, Raymond L. Barrett Jr, "Low-Rate Wireless Personal Area Networks", IEEE Std 802.15.4.
[2] Tom Messerges, Johnas Cukier, Tom A.M.Kevenaar, Larry Puhl, "A Security Design for a General Purpose, Self-Organizing, Multihop Ad Hoc Wireless Network", TR2003-114, 2004
[3] ANSI X9.63-2001, "Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography, 2001
[4] Miller, Victor S., "Use of Elliptic Curves in Cryptography", Lecture Note in Computer Science, no.218, pp.417-426, Springer-Verlag, 1986
[5] ISO/IEC 9798-2, Information Technology - Security Techniques - Entity Authentication Mechanism - Part 2, "Mechanisms Using Symmetric Encipherment Algorithm", International Standardization Organization, 1994