무선 센서 네트워크에서 타원 곡선 암호를 이용한 공유키 설정에 기반한 보안 프로토콜

서석충*, 김형찬, R.S. Ramakrishna
광주과학기술원 정보통신공학과
e-mail : {gegehe*,kimhc,rsr}@gist.ac.kr

Security Protocols Based on Elliptic Curve Cryptographic Pairwise Key Setup for Wireless Sensor Networks

Seog Chung Seo*, Hyung Chan Kim, R.S. Ramakrishna
Dept. of Information & Communications,
Gwangju Institute of Science and Technology

요 약

무선 센서 네트워크 (Wireless Sensor Network)에서 기존에 존재하는 대부분의 보안 프로토콜들은 대칭적인 공유키(symmetric pairwise key) 설정에 기반하고 있다. 그러나 이러한 프로토콜들은 노드 전복(node compromising), 그리고 과중한 트래픽의 문제점을 안고 있다. 더욱이, 대칭키 방법을 이용한 브로드캐스트 메시지 인증은 자원이 제약된 센서네트워크에서 적용하기에는 너무 복잡하다. 본 논문은 공개키를 이용한 공유키(Pairwise Key) 설정에 기반한 보안 프로토콜들을 제안한다. 특히 경량성을 위하여 타원 곡선 암호 (Ellptic Curve Cryptography)를 채택하였다. 제안 프로토콜은 공유키 설정과 브로드캐스트 메시지 인증을 위하여 각각 Elliptic Curve Diffie-Hellman (ECDH)과 Elliptic Curve Digital Signature Algorithm (ECDSA)를 이용한다. 더욱이, 분산된 rekeying 메커니즘 (decentralized rekeying mechanism)을 도입함으로써 TinySec 의 성능을 향상시킨다.

1. 서론

최근, 무선 센서 네트워크는 매우 다양한 어플리케이션에서 이용되고 있다. 이러한 센서네트워크는 자원이 제한된 작은 센서 노드들로 구성되며, 물리적으로 안전하지 않은 환경에 배치되기 때문에, 안정적이고 오래 지속되는 네트워크 통신을 위해서는 인증, 무결성, 그리고 기밀성의 결합이 필요하다.

센서 네크워크에서는 도청, 악의적인 메시지의 삽입, 그리고 노드 전복과 같은 많은 위협들이 존재한다. 이러한 위협들은 센서의 제약된 자원과 외부로부터의 접근성에 기인한다 [7, 8]. 따라서 안전한 통신을 위하여 통합적인 보안 프로토콜이 필수적이다. 적지 않은 수의 대칭키 암호에 기반한 보안 프로토콜들이 제안되었다. 대칭키 암호는 암호화 및 복호화와 같은 일반적인 암호 동작의 경우에는 효율성을 제공하지만, 공유키 설정이나 브로드캐스트 메시지 인증의 경우에는 복잡한 절차를 요구하며 과중한 트래픽을 발생시킨다. 뿐만 아니라 대칭키 암호는 노드 전복에 취약하다.

본 논문은 효율적이고 견고한 공유키 설정과 브로드캐스트 메시지 인증을 위하여 타원 곡선 암호 (ECC)에 기반한 보안 프로토콜을 제안한다. ECC 는 RSA 와 비슷한 레벨의 보안성을 제공하면서도 훨씬 적은 키 사이즈를 요구하기 때문에, 센서 네트워크에 충분히 적용 가능하다. 공유키를 설정한 이후에, 제안 프로토콜은 노드간의 기밀성과 인증을 제공하기 위하여 링크 보안 프로토콜인 TinySec [3]을 이용하며, TinySec 의 동작을 더욱 견고하게 만들기 위하여 분산된 rekeying 메커니즘을 적용한다. 본 논문은 다음과 같이 구성되어있다. 2 절에서 센서네트워크에서의 보안 목표와 이를 달성하기 위한 기존의 보안 스킴에 대해서 논의한 후에 3 절에서 기존의 보안 프로토콜들의 문제점들을 보완하는 제안 프로토콜들을 제시한다. 4 절에서 실험결과를 제시하고, 마지막 5 절에서는 본 논문의 결론을 맺는다.

2. 기존의 보안 프로토콜들

센서 네트워크에서의 보안 목표는 외부로부터의 공격에 대

한 방어, 노드 전복과 같은 내부 공격의 충격 최소화, 그리고 베이스 스테이션 (Base Station)으로부터 브로드캐스팅되는 메시지의 인증을 달성하는 것이다. 이를 위하여 여러 가지 보안 프로토콜들이 제안되었으나 센서네트워크에 적용하기에는 여러 가지 문제점을 안고 있다.

2.1 공유키 설정 (Pairwise Key Establishment)

노드들간의 공유키 설정은 노드간의 기밀성 및 인증과 같은 보안 목표를 달성하기 위한 근본적인 단계이다. 공유키 설정을 위하여 여러 가지 스킴들이 제안되었다.

가장 단순한 스킴은 네트워크 전체에 공유된 하나의 키를 사용하는 것이다. 이 방법은 외부에 있는 공격자가 네트워크의 내부 트래픽을 도청하는 것을 막을 수 있다. 하지만 만약, 하나의 노드가 외부 공격자에 의해 전복되면, 네트워크 전체의 통신이 복호화되어 공격자에게 알려질 위험이 있으며, 또한 키를 얻은 공격자가 인증된 노드인 것 처럼 다른 노드들과 통신할 수 있다.

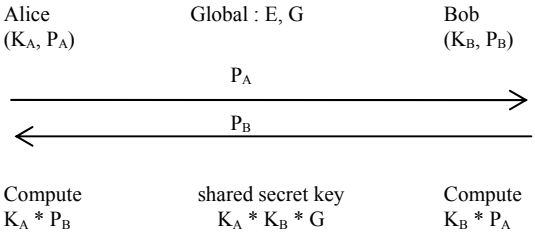
모든 가능한 노드의 짝에 대하여 대칭적으로 유일한 공유키를 미리 설정 (pre-configure)하는 방법도 제안되었다. 이 스킴을 적용하기 위해서는 각각의 노드가 최소한 (n - 1) 개의 키를 저장하여야 하며, 네트워크의 전체적인 측면에서는 (n - 1) * n / 2 개의 키를 저장하여야 한다. 따라서 이 스킴 역시 확장성이 떨어지는 단점을 가지고 있다.

또 다른 접근은 베이스 스테이션을 신뢰된 삼자(Trusted Third Party) 혹은 키 분배 센터 (Key Distribution Center)로써 이용하는 것이다. 이 프로토콜은 베이스 스테이션이 요청 메시지와 노드의 유효성을 검증할 수 있지만, 여러 가지 문제점들을 가지고 있다. 첫째로, 베이스 스테이션이 공격자에 의해 전복되면 전체 네트워크의 동작이 중단된다 (One point of failure). 두 번째로, 네트워크의 노드들이 다른 노드들과 통신하기 위한 공유키를 얻기 위해서는 베이스 스테이션과 통신해야만 하기 때문에 과중한 트래픽을 발생시킨다. 실제로 하나의 bit 를 전송시키는데 드는 전력은 800-1000 개의 명령어를 실행시키는데 드는 전력과 비슷하기 때문에 [3] 과중한 트래픽은 많은 에너지의 소모를 의미한다. 세 번째로 베이스 스테이션 근처의 노드들은 요청 메시지와 응답 메시지를 운반하는 책임을 맡고 있다. 따라서 이 노드들은 베이스 스테이션에서 멀리 떨어진 노드들에 비하여 에너지 소비가 심하다. 마지막으로 공격자가 전복된 노드로부터 베이스 스테이션과 안전하게 통신할 수 있는 마스터 키를 얻는다면, 이 공격자는 데이터 집합 노드 (aggregator)와 같은 중요한 역할을 하는 노드와 통신을 하기 위하여 베이스 스테이션으로 공유키 요청 메시지를 전송할 수 있다.

최근의 연구들은 random key predistribution 을 유망한 공유키 설정 스킴으로 간주하고 있다 [9, 10, 11]. 하지만, 이 방법은 노드 전복 공격에 취약하다. 만약 공격자가 충분히 많은 수의 노드를 전복시킨다면, 완전한 키 링을 복원해낼 수 있다. 뿐만 아니라, 최적의 라우팅 루트로 통신할 수 있는 두 개의 노드가 동일한 키를 가지고 있지 않은 경우에는 비효율적인 루트로 돌아가야만 한다.

2.2 기밀성 그리고 인증 (Confidentiality and Authentication)

센서 노드 사이에 공유키를 설정한 후에는, 이 키들을 암호 알고리즘(Cryptographic algorithm) 그리고 메시지 인증 코드 (Message Authentication Code)에 적용하여 기밀성과 인증을 제공할 수 있다. TinySec 은 이와 같은 노드간에 기밀성과 인증을 제공하는 검증된 링크 레이어 보안 아키텍처이다 [3]. 하지만 TinySec 역시 2byte 의 상대적으로 짧은 IV



(그림 1) ECDH with Precomputation.

(Initial Vector)로 인한 문제점을 가지고 있다. 만약 같은 IV 가 같은 키와 사용되면 평문이 공격자에게 노출될 수 있다. 예를 들어, $C = (IV, G_K(IV) \oplus P)$, $C' = (IV, G_K(IV) \oplus P')$, then $C \oplus C' = P \oplus P'$ 가 된다. 따라서, 이러한 문제를 해결하기 위해 TinySec 은 기존의 키를 업데이트 시키는 rekeying 프로토콜을 요구한다.

2.3 브로드캐스팅 메시지 인증

센서 네트워크의 기본적인 통신 방법은 브로드캐스팅이며, 베이스 스테이션은 자신의 인터레스트 (Interest Query) 혹은 커멘드 (Command)를 노드들에게 브로드캐스팅 함으로써 전달한다. 따라서, 센서 노드들은 베이스 스테이션으로부터 브로드캐스트된 메시지들의 무결성과 유효성을 검증하는 메커니즘이 필요하다. 만약 검증이 제대로 실행되지 않으면, 공격자가 베이스 스테이션인 것처럼 행동할 수 있다. 브로드캐스트 메시지 인증을 위하여 μTESLA 가 제안되었다 [1]. μTESLA 는 연기된 대칭키 노출 (Delayed key disclosure)을 통하여 비대칭성을 달성하지만, 이것을 위하여 시간 동기와 추가적인 키 노출 패킷을 요구하기 때문에 과중한 트래픽을 네트워크에 발생시킨다.

3. 제안 프로토콜들

이번 절에서는 2 절에서 기술된 문제점들을 완화시킬 수 있는 제안 프로토콜들 (공유키 설정, 브로드캐스트 메시지 인증, 그리고 TinySec Rekeying)을 기술한다.

3.1 Elliptic Curve Diffie-Hellman

공개키 알고리즘은 그것의 확장성, 노드 전복에의 복원력 그리고 단순한 메시지 프로토콜로 인하여 공유키 설정에 적합하다. 특히, Diffie-Hellman 키 교환 알고리즘은 센서네트워크에서도 적용 가능하다. 하지만, pure Diffie-Hellman 의 키 사이즈는 센서 네트워크에서 사용하기에 너무 길다. 이것의 권고된 키 사이즈는 1024bit 여서 하나의 키를 보내기 위해서는 적어도 다섯 개의 TinySec 메시지를 요구한다. 이것의 변형으로써 Elliptic Curve Diffie-Hellman (ECDH)이 그것의 작은 키 사이즈와 pure Diffie-Hellman 과 비슷한 보안 레벨의 제공으로 인하여 가장 유망한 접근으로 고려된다. 키 사이즈 측면에서 ECDH 의 권고된 키 사이즈는 163 bit [2, 12]이지만, 현실적으로 지금까지 붕괴된 가장 큰 키 사이즈는 109 bit 이기 때문에 (2004 년 4 월, 2600 대의 컴퓨터로, 17 개월이 소모됨) 제안 프로토콜에서는 113 bit 키를 채택하였다 [5]. 이것은 센서 노드의 생존 시간을 고려해볼 때 충분히 타당성 있다. ECDH protocol 은 다음과 같이 동작한다.

A, B: 센서 노드들, n: large prime

1. A: [1, n – 1]의 범위에서 비밀키 K_A 를 선택한다. 그리고 public key ($P_A = K_A * G$)를 계산한다.

2. B: [1, n – 1]의 범위에서 비밀키 K_B 를 선택한다. 그리고 public key ($P_B = K_B * G$)를 계산한다.
3. A: 자신의 public key (P_A)를 담은 요청 메시지를 B 에게 보낸다.
4. B: A 의 public key 를 받은 후에, 자신의 public key (P_B)를 담은 응답 메시지를 A 에게 보낸다.
5. A and B: 각자 공유키를 계산한다.
$K_{AB} = K_A * P_B = K_B * P_A = K_A * K_B * G$.

위의 절차에서 볼 수 있듯이, 하나의 센서 노드는 네트워크의 어떤 노드와도 public key 를 교환함으로써 공유키를 설정할 수 있다. 이것은 ECDH 는 과중한 트래픽을 발생시키지 않는 다는 것을 의미한다. 더욱이 각각의 노드들이 자신의 private/public 키 쌍만 유지하기 때문에 노드 전복에 대하여 탄력성을 제공한다.

Private/public key 쌍을 네트워크의 배치 전에 미리 계산하여 각각의 센서 노드에 저장함으로써 ECDH 의 성능을 향상시킬 수 있다 [그림. 1]. 이 메커니즘을 이용하여, 위의 절차에서 스텝 1 과 2 를 제거함으로써 private/public key 쌍을 계산해야 하는 CPU 의 오버헤드를 줄일 수 있다.

3.2 브로드캐스팅 메시지 인증

2.3 절에서 기술되었듯이, μTESLA 는 복잡한 대칭 키 아키텍쳐로 인한 많은 단점이 있다. 따라서 제안프로토콜에서는 브로드캐스팅 메시지 인증을 위하여 μTESLA 와 대조되는 공개키 아키텍쳐를 이용한다.

제안 프로토콜은 시그너쳐의 생성과 검증을 위하여 Elliptic Curve Digital Signature Algorithm (ECDSA)를 채택하였다 [4, 6]. ECDSA 는 ECC 에 기반하기 때문에 시그너쳐의 길이가 상대적으로 짧다는 장점이 있다. ECDSA 에서 30byte 의 패킷은 200bit 길이의 시그니쳐에 대응된다. 200bit 는 하나의 TinySec 패킷으로 전송될 수 있기 때문에 브로드캐스팅 메시지 인증을 위하여 오직 하나의 추가적인 메시지 전송이 필요하다. 이것은 μTESLA 가 시간 동기 메시지, 키 노출 패킷 그리고 노드에서의 메시지 버퍼링을 요구하는 것에 비하여 훨씬 적은 오버헤드를 요구한다.

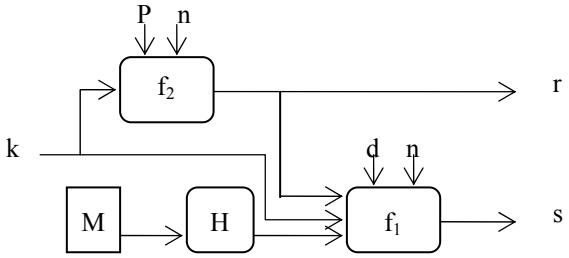
ECDSA 를 위한 필요조건은: 베이스 스테이션의 public key: (E, P, n, Q), 베이스 스테이션의 private key: d ε [1, n – 1], E: Elliptic Curve, P: a point ε E, Q = dP, 베이스 스테이션의 public key 와 E 그리고 n 은 네트워크 전체에 공개된 정보이다.

ECDSA 의 시그너쳐 생성 절차는 [그림. 2]에 기술되어 있다. 베이스 스테이션은 메시지 m 에 대한 시그너쳐를 생성하기 위하여

1. [1, n – 1]의 범위에서 예측하기 힘든 유일한 정수 k 를 선택한다.
2. $kP = (x_1, y_1)$과 $r = x_1 \bmod n$ 을 계산하고, 결과인 r 이 0 이 되면 다시 step1 로 돌아간다.
3. $k^{-1} \bmod n$ 을 계산한다.
4. $s = k^{-1}\{h(m) + dr\} \bmod n$ 을 계산한다 (h = SHA-1). 만약 결과인 s 가 0 이면 다시 step1 로 돌아간다.
5. 메시지 m 에 대한 시그너쳐는 (r, s)이다.

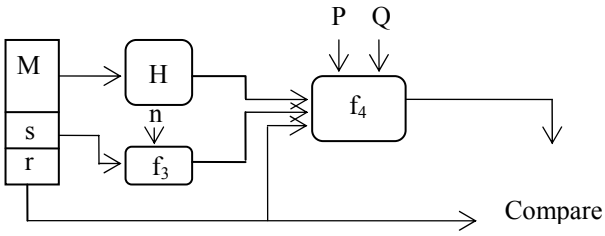
ECDSA 의 시그너쳐 검증 절차는 [그림. 3]에 기술되어 있다. 각각의 센서노드에서 베이스 스테이션이 보낸 브로드캐스팅 메시지를 검증하기 위하여 다음과 같은 절차를 수행한다.

1. r 과 s 가 [1, n – 1]의 범위에 있는지 검사한다.
2. $w = s^{-1} \bmod n$ 그리고 h(m)을 계산한다.
3. $u_1 = h(m)w \bmod n$ 그리고 $u_2 = rw \bmod n$ 을 계산한다.
4. $u_1P + u_2Q = (x_0, y_0)$ 그리고 $v = x_0 \text{ mon d}$ 을 계산한다.



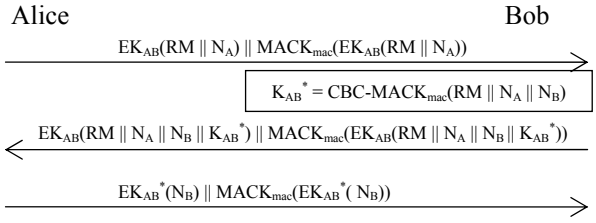
$kP = (x_1, y_1)$, $r = x_1 \bmod n$,
$s = k^{-1}\{h(m) + dr\} \bmod n$, h: SHA-1

(그림 2) ECDSA generation.



$w = s^{-1} \bmod n$, $u_1 = h(m)w$, $u_2 = rw \bmod n$,
$u_1P + u_2Q = (x_0, y_0)$, $v = x_0 \bmod n$.

(그림 3) ECDSA verification.



(그림 4) Rekeying.

5. v = r 을 만족하면 시그너쳐를 인증한다.

3.3 TinySec Rekeying

TinySec 의 상대적으로 짧은 IV 문제를 보완하기 위하여 기존의 키를 새로운 키로 업데이트하는 rekeying 메커니즘이 필요하다. 또한, 공유키를 새로운 키로 자주 업데이트할수록, 더욱 견고한 보안성이 보장된다. 따라서, 이러한 요구를 충족시키기 위하여 rekeying 메커니즘은 가볍고 효율적이어야 한다. Rekeying 메커니즘을 제공하기 위하여 ECDH 가 제안되었다 [2]. ECDH 에서는 새로운 session 키를 생성할 때마다 새로운 private/public 키 쌍을 생성해야만 한다. 이것은 CPU 에 많은 부담을 주기 때문에 rekeying 메커니즘의 필수 조건인 경량성과 실용성에 위배된다.

분산된 키 분배 스킴 (decentralized key distribution scheme)은 rekeying 메커니즘의 요구에 가장 부합한다. 이 메커니즘은 새로운 세션 키의 생성과 분배를, 이것을 요청하는 노드와 이 노드의 이웃 노드들로 지역화함으로써 오버헤드를 줄인다.

Rekeying 절차는 [그림. 4]에 기술되어있다.

K_{AB}: 바꿀 세션 키, K_{AB}^{*}: 새로운 세션 키, N: 넌스 값, RM:

<표 1> EccM 2.0 과 제안 프로토콜의 비교.

	EccM 2.0 [2]	Proposed
CPU	1.5145 J	0.3509 J
Radio	2.5795 J	1.1079 J
Total	4.0940 J	1.4588 J
Public key	34.764 sec	12.296 sec
Pairwise key	29.788 sec	8.7452 sec

<표 2> Rekeying 에 소모된 에너지.

	Decentralized Rekeying
CPU	0.01822 J
Radio	1.1079 J
Total	1.1261 J

요청 메시지.

1. A → B: 요청 메시지를 보낸다.

$EK_{AB}(RM \| N_A) \| MACK_{mac}(EK_{AB}(RM \| N_A))$

2. B: 새로운 세션 키의 생성.

$K_{AB}^{*} = CBC\text{-}MACK_{mac}(RM \| N_A \| N_B)$

3. B → A: 새로운 세션 키의 전송.

$EK_{AB}(RM \| N_A \| N_B \| K_{AB}^{*}) \| MACK_{mac}(EK_{AB}(RM \| N_A \| N_B \| K_{AB}^{*}))$

4. A → B: 인증을 위한 넌스 값 전송.

$EK_{AB}^{*}(N_B) \| MACK_{mac}(EK_{AB}^{*}(N_B))$

마지막의 인증을 위한 선택적인 메시지 전송을 제외하면 이 스킴은 단지 두 개의 메시지 전송만을 요구한다. 이것은 이 프로토콜이 가볍고 실용적이라는 것을 의미한다.

4. 실험 결과

제안 프로토콜의 타당성을 검증하기 위하여 ECDH 와 rekeying 메커니즘을 TinyOS 1.1.14 버전에서 구현하였다. 113 bit 키를 사용하는 ECDH 를 구현하기 위하여 [14]에서 권고한 113bit Elliptic Curve domain parameter 를 사용하였다. [2]의 EccM 2.0 에서는 163bit 키에 기반한 ECDH 를 구현하였기 때문에 공개키의 x, y 좌표를 각각 다른 패킷에 담아 보내야만 했다. 따라서, 공유키를 설정하기 위해서는 적어도 4 개의 메시지 전송이 필요했다. 제안된 ECDH 는 113bit 키에 기반하기 때문에 두 개의 좌표를 하나의 메시지에 담아 전송할 수 있으므로 공유키를 설정하는 것은 두 번의 메시지 전송만을 요구한다.

[표. 1]은 EccM 2.0 과 제안된 ECDH 의 에너지 소비와 키 생성시간을 비교한다. 통신에 소비된 에너지를 고려해볼 때, EccM 2.0 은 제안된 ECDH 보다 1.4725J 을 더 소비하였다. 이것은 EccM 2.0 이 공유키를 설정하기 위해 제안 프로토콜보다 더 많은 메시지 전송을 요구하기 때문이다. Private/public 키 쌍을 precomputation/preloading 함으로써 public 키를 생성하는 시간을 절약할 수 있다. 이 메커니즘을 이용함으로써, EccM 2.0 과 제안 프로토콜에 대하여 각각 54%, 67%의 CPU 에너지를 절약할 수 있었다.

분산된 rekeying 메커니즘은 단지 두 개의 메시지 전송만을 요구하기 때문에 [표. 2]와 같이 요청메시지와 응답메시지를 전송하는데 1.1079J 의 에너지가 소모되고 CBC-MAC 을 이용하여 새로운 세션키를 생성하는 데에는 0.01822J 이 소모된다. 실험의 결과로 미루어볼 때 제안된 rekeying 메커니즘은 센서 네트워크에 적용하기에 충분한 경량성을 가지는 것을 알 수 있다. 또한 EccM 2.0 은 단순히 public 키를 전송하고 이것을 이용하여 공유키를 계산하는 부분만 구현하였지만, 제안된 프로토콜에서는 더 나아가 공유된 키를 TinySec 에 적용하여 데이터 통신을 하고, 키를 업데이트시킬 필요가 있을 때, rekeying 메커니즘을 이용하여 새로운 세션키를 공유하는 것까지 구현하였다. 시뮬레이션 결과는 제안된 보안 프로토콜이 센서네트워크에서 충분히 적용 가능하다는 것을 보여준다.

5. 결론 및 향후 과제

절 3.2 에서 제안된 ECDSA 에 대한 이론적 근거를 제시하였으나 실험을 통하여 이것을 증명하지는 않았다. 향후 과제는 이것을 실제로 TinyOS 에 구현하여 다른 가능한 접근과 비교하는 것이다. 또한 현재 주를 이루고 있는 대칭 키를 이용한 보안 프로토콜을 실제로 구현하여 비교할 예정이다.

본 논문은 기존의 센서 네트워크 보안 프로토콜의 문제점을 보완할 수 있는 프로토콜을 제시하였다. 이론적 근거를 제시함으로써 공유키 설정과 브로드캐스팅 메시지 인증을 위하여 ECC 를 이용한 ECDH 와 ECDSA 가 각각 유망한 방법임을 확인하였으며, 실험을 통하여 제안된 ECDH 와 분산 rekeying 메커니즘의 경량성과 센서네트워크에 적용가능성을 검증하였다.

참고문헌

[1] A. Perrig, et al., “SPINS: security protocols for sensor networks. Wireless Networks,” 8(5):521-534, 2002.

[2] D. J. Malan, M. Welsh , and M.D. Smith, “A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography,” First IEEE International Conference on Sensor and Ad Hoc Communications and Networks, 2004.

[3] C. Karlof, N. Sastry, and D. Wagner, “TinySec: Link Layer Security Architecture for Wireless Sensor Networks,” SenSys 2004, Baltimore, MD, 2004.

[4] G. Gaubatz, et al., “State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks,” 2005.

[5] M. Zitterbart, et al., “Efficient Implementation of Elliptic Curve Cryptography for Wireless Sensor Networks,” 2005.

[6] D. B. Johnson, A. J. Menezes, “Elliptic Curve DSA (ECDSA): An Enhanced DSA”.

[7] A. Perrig, J. Stankovic, and D. Wagner, “Security in Wireless Sensor Networks,” 2004.

[8] E. Shi, and A. Perrig, “Designing Secure Sensor Networks,” IEEE Wireless Communication, 2004.

[9] W. Du, et al., “A pairwise Key Pre-distribution Scheme for Wireless Sensor Networks,” Proc. 10th ACM Conf. Comp. and Commun. Security, pp. 42-51, 2003.

[10] H. Chan, A. Perrig, and D. Song, “Random Key Pre-distribution Schemes for Sensor Networks,” IEEE Symp. Security and Privacy, 2003.

[11] L. Eschenauer and V. D. Gligor, “A Key-Management Scheme for Distributed Sensor Networks,” Proc. 9th ACM Conf. Comp. and Commun. Security, pp. 41-47, 2002.

[12] National Institute of Standards and Technology, “Recommended Elliptic Curves For Federal Government Use,” “http://csrc.nist.gov/CryptoToolkit/dss/-ecdsa/NISTReCur.pdf,” July 1999.

[13] W. Diffie, M. E. Hellman, “New Directions in Cryptography,” IEEE Transactions on Information Theory, 1976.

[14] Certicom Research, “SEC 2 : Recommended Elliptic Curve Domain Parameters,” http://secg.org.