2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

WAP에서의 새로운 종단간 인증 프로토콜

양종필*, 조현호*, 이경현**
*부경대학교 전자계산학과
**부경대학교 전자컴퓨터정보통신공학부
e-mail:bogus@unicorn.pknu.ac.kr

A New E2E Authentication Protocol in WAP

Jong-Phil Yang*, Hyun-Ho Cho*, Kyung-Hyune Rhee**
*Dept of Computer Science, PuKyong National University
**Division of Electronic, Computer & Telecommunication Engineering, Pukyong National University

요약

본 논문에서는 WAP 포럼에서 제시한 WAP 프로토콜의 종단간 안전성 취약성을 분석하고 새로운 종단간 보안 세션을 위한 인증 프로토콜을 제안한다.

I. 서론

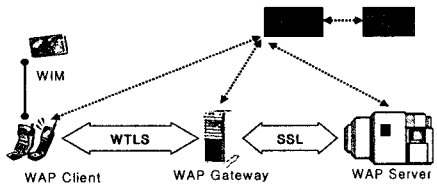
최근 무선 인터넷 시장의 급속한 성장에 힘입어 새로운 고객의 확보와 다양한 서비스 창출이 이루어지고 있다. 오퍼레이터와 생산자들은 진보된 서비스들의 도전에 부응하기 위해, WAP Forum을 구성하여 전송, 보안, 트랜잭션, 세션 및 응용 계층을 위한 프로토콜 집합인 WAP(Wireless Application Protocol)을 정의하였다. WAP Forum은 WAP 환경에서의 안전한 통신을 위해서 WTLS(Wireless Transport Layer Security)를 개발하였으며, 통신 개체들 사이에 프라이버시, 데이터 무결성 및 인증 서비스를 제공한다. WTLS는 기본적으로 SSL 3.0과 비슷한 기능을 지원하며, 부가적으로 데이터그램 지원, 최적화된 핸드쉐이크 및 동적인 키 재생 등의 몇 가지 새로운 특징을 가지고 있다. 또한, WTLS는 낮은 대역폭 기반 네트워크(low-bandwidth bearer network)에 최적화 되어있다[1][2].

현재, WAP의 심각한 보안적 문제점은 WTLS 자체의 문제라기 보다는 WAP Gateway라는 새로운 네트워크 요소를 추가함에 의해서 발생된다. WTLS는 기존의 유선 인터넷에서 사용되고 있는 SSL과 직접적으로 호환되지 못하므로, WAP Gateway는 WAP Client로부터 수신한 메시지를 복호화한 후, 다시 재암호화를 해서 WAP Server에게 전달하기 때문에, WAP Gateway내부에서는 전송하고자 하는 메시지의 평문(Plaintext)이 노출되는 문제점을 가지고 있다. 즉, WAP Client와 WAP Server간의 종단간의 보안(End-to-End Security)이 제공되지 못하고 있다. 따라서, 본 논문에서는 WAP 환경에서 WAP Client의 사용자 인증, 세션키 설정, 동적인 키 재생, 종단간의 보안 서비스를 제공하는 새로운 인증 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 WAP의 전체적인 보안 구조와 몇 가지 단점을 제시하여, 새로운 프로토콜 설계의 필요성을 제시한다. 3장에서는 제안 프로토콜의 모델 및 전반적인 동작 원리를 제안하고, 4장에서는 제안 프로토콜을 상술한다. 5장에서는 제안 프로토콜의 간략한 구현 방안을 제시하고 결론을 맺는다.

II. WAP 보안 구조 및 단점

1. WAP 보안 구조



[그림 1] WAP 환경에서의 보안 구조

WAP Forum에서 제안한 WTLS의 Handshake Protocol, Alert Protocol, Change Cipher Spec Protocol은 WTLS의 동작에 대한 관리를 위해 사용되며, 실질적인 보안 서비스는 Record Protocol에서 제공된다. 클라이언트와 서버가 WTLS를 이용해 연결을 할 경우, 가장 먼저 Handshake Protocol을 수행하여 한 세션동안 보안 서비스 제공에 사용될 세션키, 암호 알고리즘, 인증서등과 같은 암호 매개변수를 서로 공유하게 된다. 위의 매개변수들은 Record Protocol

에서 보안 서비스를 제공하기 위해 사용된다[1][2].

[그림 1]은 무선 환경에서의 보안 서비스를 위해서 WTLS가 사용되며, 유선 환경에서의 보안 서비스를 위해서 SSL이 사용됨을 보이고 있다.

2. WAP 보안 구조의 단점

현재의 WTLS를 기반한 WAP 보안 구조는 아래의 같은 몇 가지 문제점을 가지고 있다.

- WAP Client와 WAP Server간의 종단간의 보안 서비스를 제공하지 못한다.
- 안전한 보안 세션을 연결하기 위해서, 많은 공개키 인증서가 사용되며, 또한 수신한 인증서를 검증하기 위한 CRLs(Certificate Revocation Lists)의 질의 및 검증을 위한 자원 낭비가 심하다[3].
- WTLS 자체적인 보안 결함이 발표되고 있다[4][5][6][7].
- 공개키 인증서에 기반한 개체 인증은 단지 통신 개체의 적합성 판단을 위해서 사용되기 때문에, 접근 개체의 권한(Authorization)에 대한 정보를 충분히 제공하지 못한다[8].

본 논문에서 제안하는 E2ESP(End-to-End Shared Security Protocol)은 WAP 프로토콜 스택내의 WAE(Wireless Application Environment) 계층에서 운용될 수 있는 보안 프로토콜을 제안하며, 제안된 프로토콜은 기존의 WTLS/SSL과 동작되는 계층이 다르기 때문에, 함께 운용되는 것도 가능하다.

본 논문에서 WAP Gateway는 기본적으로 가장 일반적이라고 할 수 있는 Network operator의 서브네트워크 내부에 위치하는 상황을 고려한다. 현재 대부분의 사용자는 Network operator의 WAP Gateway에서 제공되는 WAP Portal에서 링크된 WAP 싸이트를 주로 검색할 것이므로, 사용자가 직접 URL을 입력하는 싸이트의 접근은 상대적으로 낮을 것이다[9]. 그러므로, 본 논문에서 제안하는 방안은 일반적인 WAP Portal 기반의 무선 인터넷 서비스에 초점을 맞추고 있다. 하지만, 어느 특정 회사 단위로 WAP Gateway를 운영하고 인트라넷과 같은 내부 네트워크에 접근하는 사용자들의 인증 및 접근 제한이 필요한 환경에서도, 제안하는 프로토콜 E2ESP는 적절히 활용될 수 있다.

III. CRL-Agent 및 전제 조건

사용자가 WAP Client로서 무선 인터넷을 시작할 때, 기본적으로 WAP Portal이 WAP 브라우저의 홈페이지로 설정되어 있다. WAP Portal에서 링크된 WAP Server들이 보안이 요구되는 싸이트, 즉 온라인 뱅킹, 주식 서비스, M-commerce라면, 이미 Network operator와 E2ESP를 지원하도록 협정한다. WAP Server의 입장에서는 E2ESP의 추가로 인해 발생되는 WAP Server 시스템의 변경으로는 기존의 환경에서 "협정된 WAP Gatewav로부터 페이지 요청이 오는가?"에 따라서, E2ESP를 on/off하면 된다.

CRL-Agent는 WAP Client와 WAP Server 사이의 E2ESP에서 사용되는 공개키 인증서에 대한 취소 및 변경 여부를 조사한다. CRL-Agent의 트래픽 오버헤드를 고려하여, E2ESP에서 사용될 공개키에 대한 인증서의 발행을 특정 몇몇의 CA(Certificate Authority)로 제한하는 방법도 고려할 수 있다. CRL-Agent는 CRLs(Certification Revocation Lists)를 정기적으로 조사하여, 취소 및 변경 여부에 대한 상태 및 그 WAP Server의 E2ESP를 위한 공개키를 저장하고 있다. WAP Client가 WAP Gateway에게 WAP Portal을 요청할 경우, WAP Gateway가 CRL-Agent에게 요청 WAP Client와 관련이 있는 WAP Server들의 공개키 상태 정보를 요청한다. 이 요청을 수신한 CRL-Agent는 그 WAP Client가 요구하는 WAP Server들의 공개키 상태 정보를 WAP Gateway에게 전송한다. WAP Gateway는 WAP Portal과 함께 수신한 공개키 상태 정보를 WAP Client에게 전송한다. CRL-Agent가 공개키 상태 정보를 WAP Gateway에게 전송할 때, 기밀성을 위한 암호처리 없이 전송된다. 만약, WAP Gateway와 CRL-Agent 사이에 SSL이 있다면, 당연히 세션키로 암호화된 리스트가 전송될 것이다. 또한, CRL-Agent와의 통신을 위한 WAP Client의 공개키는 단말기 초기 설정시에 이미 CRL-Agent는 알고 있으며, CRL-Agent는 정기적으로 가입된 WAP Client의 공개키 인증서를 검사하여 WAP Client의 공개키 취소 여부를 판단한다. 만약, CRL내에 WAP Client가 발견될 경우에는 CRL-Agent는 WAP Client에게 새로운 Certificate URL을 요청하며, 이에 따른 절차는 WAP Forum 스펙[3]에 자세히 언급되어 있다.

WAP Client는 Network operator(WAP Gateway와 CRL-Agent를 보유함)를 아래의 관점에서 신뢰한다.

- WAP Client는 CRL-Agent가 안전하다고 믿는다. 즉, CRL-Agent의 공개키는 안전하며, 단말기 초기 설정시 CRL-Agent의 공개키를 이미 알고 있다. 그러나, 주기적으로 CRL-Agent의 공개키 유효성 여부를 판단하기 위해서, CRLs를 질의할 수 있다.
- WAP Client는 CRL-Agent에 의해서 제공된 WAP Server의 공개키 상태 관련 정보를 서명의 검증을 통해서 확인할 수 있다.

IV. WAP 환경에서의 E2ESP 프로토콜

1. 표기법

본 논문에서 사용되는 표기법은 아래와 같다.

- C : WAP Client를 나타내는 식별자. 예를 들어, MIN(Mobile station's Identification Number)를 사용할 수 있다.
- S : WAP Server를 나타내는 식별자.
- G/W : WAP Gateway를 나타내는 식별자
- $CrlA$: CRL-Agent를 나타내는 식별자.
- U : WAP Client의 사용자를 나타내는 식별자(WAP Server의 사용자 계정).
- Pu_X : 통신 개체 X의 공개키(public key).
- Pr_X : 통신 개체 X의 비밀키(private key).
- $Hash(m)$: 메시지 m에 일방향 해쉬함수를 적용함. (예, SHA-1)
- $Ppass$: 공개 패스워드로서, WAP Server의 공개키를 해쉬처리한 값.
- $Upass$: 사용자 선택의 취약한 패스워드, WAP Server에 사용자의 계정(U)에 대응되는 패스워드
- r_X : 통신 개체 X의 랜덤 챌린지(Random challenge).
- seq_num : 메시지의 순차 넘버로서, 처음 0으로 초기화되어 있다.
- $refresh$: 얼마나 자주 세션키가 업데이트 되어서, 새로운 세션키들이 계산되는 가를 나타낸다. 이 값은 WAP Server가 설정하며, 새로운 키를 위한 n은 $n = 2^{refresh}$ 이며, 새로운 키를 가지는 메시지들의 순차넘버(seq_num)는 $0, n, 2n, 3n, \ldots$ 이 된다.
- ZZ : WAP Server와 WAP Client간의 E2ESP를 통한 키 협정에 의해서 유도된 공유 비밀값.
- SK : WAP Server와 WAP Client가 ZZ 를 계산한 뒤, WAP Server와 WAP Client 사이의 비밀 통신을

위해서 사용될 세션키.

- $E_A(m)$: 키 A 를 사용하여 메시지 m을 암호화.
- $D_A(m)$: 키 A 를 사용하여 메시지 m을 복호화.

2. E2ESP의 초기화

메시지 1 [C->G/W]
WAP Portal Request, C, $E_{Pr_C}(C, r_C)$
메시지 2 [G/W->CrlA] C, $E_{Pr_C}(C, r_C)$
메시지 3 [CrlA->G/W]
$CrlA$, $E_{Pr_{CrlA}}$(*Server Lists*, $CrlA$, C, r_C)
메시지 4 [G/W -> C]
WAP Portal Pages,
$CrlA$, $E_{Pr_{CrlA}}$(*Server Lists*, $CrlA$, C, r_C)

[그림 2] E2ESP의 초기화

WTLS 세션이 설정된 후에 WAP Client가 WAP Portal을 WAP Gateway에게서 수신받을 때의 메시지 흐름은 [그림 2]과 같으며, WAP Client와 WAP Gateway간의 WTLS의 활성화는 반드시 필요한 것은 아니다.

WAP Client는 WAP Gateway에게 WAP Portal을 수신하기 위해서 메시지 1을 보낸다. 메시지 1를 수신한 WAP Gateway가 WAP Portal을 WAP Client에게 전송하기 전에, 메지지 2를 전송함으로써, 해당 WAP Client가 E2ESP를 사용하는 WAP Server들의 공개키 상태 정보를 요청한다. CRL-Agent는 각각의 WAP Client에 대해서 "어떠한 WAP Server에 대한 공개키 검증이 필요한가?"를 데이터베이스화하고 있다. 이 데이터베이스를 "인증서 조회 리스트"라고 한다. CRL-Agent는 이미 검사된 WAP Server들의 공개키 상태 정보에서 요청 WAP Client(C)의 인증서 조회 리스트에 포함된 WAP Server들의 공개키 상태 정보만을 검출하고, 그 중에서 인증서가 취소된 서버들의 신원을 목록화하여 "*Server Lists*"를 생성한 후, 메시지 3를 WAP Gateway에게 전송한다. 메시지 3을 수신한 WAP Gateway는 메시지 4를 C에게 전송한다. 이를 수신한 C는 WAP Portal을 브라우저에 나타내며, 수신된 리스트를 CRL-Agent의 공개키로 서명을 검증하여, 어떠한 WAP Server의 공개키가 취소 및 변경되었는 지를 알게 된다.

만약 C가 접근하고자 하는 WAP Server의 E2ESP를 위한 인증서가 취소되거나 변경되었을 때, C는 해당되는 공개 패스워드를 갱신하기 위한 메시지 흐름은 [그림 3]과 같다. 공개 패스워드는 4.3절에서 좀 더 자세히 언급된다.

메시지 1 [C->G/W] C, $E_{Pr_C}(S, C, r_C)$
메시지 2 [G/W->CrlA] C, $E_{Pr_C}(S, C, r_C)$
메시지 3-1 [CrlA->G/W]
$CrlA$, $E_{Pr_{CrlA}}(CrlA, S, C, Pu_S, r_C)$
메시지 3-2 [CrlA->G/W]
$CrlA$, $E_{Pr_{CrlA}}(CrlA, S, C, Info, r_C)$
메시지 4-1 [G/W->C]
$CrlA$, $E_{Pr_{CrlA}}(CrlA, S, C, Pu_S, r_C)$
메시지 4-2 [G/W->C]
$CrlA$, $E_{Pr_{CrlA}}(CrlA, S, C, Info, r_C)$

[그림 3] 공개 패스워드 갱신

메시지 1는 WAP Client의 인증서 조회 리스트내에 S를 등록시키거나, 수신된 "*Server Lists*"내에 해당 S가 포함된 경우, S의 변경된 공개키 정보를 요청 또는 인증서 조회 리스트로부터 S의 신원을 삭제하는 요청을 나타낸다. 메시지 1을 수신한 WAP Gateway는 이를 CRL-Agent에게 전달하며, 이 메시지를 수신한 CRL-Agent는 다음의 동작을 수행한다.

- 만약, 수신된 S 가 "인증서 조회 리스트"에 포함되어 있을 경우,
  - 첫째로, S의 E2ESP를 위한 인증서가 취소되었을 경우에는 CRL-Agent는 그 S를 C의 "인증서 조회 리스트"로부터 삭제하고, 이를 나타내는 "*Info*"를 C에게 알린다. 즉, CRL-Agent는 WAP Gateway에게 메시지 3-2를 전송한다.
  - 둘째로, 해당 WAP Server의 E2ESP를 위한 인증서가 변경되었을 경우에는 CRL-Agent는 해당 WAP Server의 새로운 공개키 정보를 포함하는 메시지 3-1를 WAP Gateway에게 전송한다.
- 만약, 수신된 S 가 WAP Portal에서 링크된 것이며, 현재 C의 "인증서 조회 리스트"에 S 가 포함되어 있지 않을 경우, CRL-Agent는 S 를 C의 "인증서 조회 리스트"에 추가하고, S의 공개키에 대한 정보를 WAP Gateway에게 전송한다. 즉, 메시지 3-1를 전송한다.

WAP Gateway는 수신한 메시지 3-1이나 3-2를 C에게 단지 전달만을 한다. 메시지 4-1이나 4-2를 수신한 C는 다음과 같이 동작한다.

- 만약, C가 메시지 4-1를 수신했을 경우에는, 수신한 S의 공개키를 해쉬함수를 적용시켜서, 사용자가 공개 패스워드로 사용할 수 있게 한다.
- 만약, C가 S의 "인증서 조회 리스트"에서 삭제되었다는 정보인 메시지 4-2를 수신할 경우, 그 S와의 통신을 위해서는 E2ESP를 비활성화 시킨다. 그러나, 이는 기존의 WTLS/SSL 보안 세션과는 독립적이다.

위의 E2ESP의 초기화에서 time-stamp값을 각 메시지에 파라메터로 추가하여, 재생공격을 방어할 수 있다.

3. E2ESP

CRL-Agent로부터 WAP Server들의 공개키 상태 정보를 수신한 WAP Client의 사용자는 WAP Server에 로그인하기 위해서 사용자 정의의 취약한 패스워드를 기반으로 하는 E2ESP를 통해서 보안 세션을 위한 메시지 교환을 시작할 수 있다. E2ESP는 Diffie-Hellman 키 교환 방식을 기반한 EKE(Encrypted Key Exchange)방식을 채택하고 있다[10][11][12]. WAP Client의 사용자 및 WAP Server는 먼저 큰 소수(prime) p 과 g 를 "g가 mod p의 원시근(primitive)"으로 협정한다(p 과 g는 공개값). 사용자는 공개 패스워드로서 WAP Server의 공개키(Pu_S)를 일방향 해쉬처리한 값, $Ppass = Hash(Pu_S)$을 가지고 있다. 그리고, WAP Server는 사용자 인증을 위해서, 임의의 사용자 정의의 취약한 패스워드($Upass$)에 대한 관련된 정보를 $v = g^{Hash(Upass,\ U,\ S)}$ 형태로서 저장하고 있다.

WAP Client의 사용자가 WAP Server에 로그인을 하기 위한 절차는 [그림 4]와 같다.

메시지 1 [C->S] *User Login Request*
메시지 2 [S->C] r_S, Pu_S, g^x, $g^{x'}$, *key refresh list*
메시지 3 [C->S] $E_{Pu_S}(U, S, g^y, refresh, E_{SK}(r_S)), r_C$

[그림 4] E2ESP 보안세션 설정

WAP Client(C)의 사용자는 WAP Server(S)에게 단순히 로그인을 위한 메시지 교환 시작을 요청하는 메시지 1을 전송하며, 이때 WAP Gateway는 단지 수신한 메시지의 전달기능만을 수행한다.

C로 부터의 메시지 1을 수신한 S는 키 교환을 위하여, 임의의 랜덤한 큰 정수 x 와 x'를 생성하여, g^x, $g^{x'}$를 메시지 2내에 포함시킨다. 물론, x 와 x'는 S에서 비밀정보이다. Random challenge로서, r_S를 생성하고 세션의 기밀성을 좀더 높이기 위해서 S가 지원하는 세션키 재생성 주

기(*refresh*)를 목록화하여 메시지 2내에 포함시킨다.

메시지 2를 수신한 C는 S의 공개키(Pu_S)를 해쉬함수에 적용한 결과와 공개 패스워드(*Ppass*)가 일치하는 가를 검사한다. 수신한 S의 공개키와 공개 패스워드의 비교 방법은 5장에서 자세히 언급된다. 만약, 위의 두 값이 일치하지 않을 경우에는 C는 CRL-Agent에게 공개키 확인 요청을 한다. 또한, 위의 두 값이 일치할 경우, C는 사용자의 신분(U), 패스워드 (*Upass*)를 사용자에게 입력받고, S가 전송한 *key refresh list*내에서 적절한 *refresh*를 선택한 후, 랜덤하게 y를 생성하고, g^y 를 계산하여 메시지 3에 이를 포함시킨다. 이 때, y는 C의 비밀 정보이다.

C는 S와의 안전한 세션을 위한 공유된 비밀값(ZZ)과 세션키(SK)를 다음과 같이 계산한다.

$$ZZ = Hash((g^x)^y, (g^x)^{Hash(Upass,\ U,\ S)})$$

$$SK = Hash(ZZ, r_S, r_C, seq\_num)$$

위의 모든 절차가 끝난 후에 C는 메시지 3을 생성하여 S에게 전송한다. 메시지 3를 수신한 S는 C에서 로그인을 시도하는 사용자의 신원(U)과 g^y 값을 알게되며, 공유된 비밀정보(ZZ)와 세션키(SK)를 아래와 같이 계산한다.

$$ZZ = Hash((g^y)^x, v^x)$$

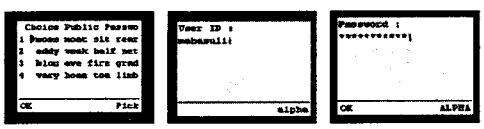
$$SK = Hash(ZZ, r_S, r_C, seq\_num)$$

S는 계산된 세션키로 암호화된 응답(response)을 복호화하여, 수신한 random-challenge 값(r_S)이 자신이 생성했던 값과 동일한 지를 비교하고, 두 값이 일치할 경우 세션키의 생성 및 C의 사용자가 정당함을 알게된다.

최초의 seq_num값은 0이며, S와 C간의 메시지 교환이 발생할 때 마다 순차 넘버는 1씩 증가한다. 만약, 세션키가 *refresh*협정을 따라서 새롭게 생성되어야 할 경우에는, 위의 세션키 생성식은 seq_num값만이 변한 채, 다시 계산되어 새로운 세션키를 생성한다.

V. E2ESP 프로토콜의 구현 방안 및 결론

E2ESP 프로토콜의 적용시에 공개 패스워드 입력의 인터페이스를 위해서, 다음의 방안을 고려할 수 있다.



[그림 5] 사용자 인증 인터페이스

공개 패스워드는 사용자가 암기를 하거나, 종이, 단말기내의 저장 옵션등의 다른 여러 방법을 사용하여 기억하도록 한다. 공개 패스워드는 WAP Server의 공개키를 해쉬처리한 값이기 때문에, 이진스트림 형태이다. 따라서, 사용자가 쉽게 이를 기억하는 것은 어렵다. 그러므로, S/Key의 이진 형태의 값을 문자사전내의 단어들로 매핑시키는 방법(예, moss mont sit rear rate pit)을 적용해서 사용자가 쉽게 기억하도록 한다[13]. WAP Client가 WAP Server의 공개키를 수신하면(E2ESP의 메시지 2), 사용자에게 [그림 5]와 같이 공개 패스워드를 확인하는 화면을 구성한다.

여기서, 올바른 공개 패스워드 값이 1번이라고 가정할 경우, WAP Client는 수신한 WAP Server의 공개키의 해쉬값을 1번에 출력하고, 나머지(2번,3번과 4번)는 임의적으로 출력하여 사용자가 선택하게 한다.

그리고, E2ESP에서 위의 WAP Server의 공개키 검증이 끝나면, 사용자는 WAP Server에 로그인을 수행하기 위하여, [그림 5]와 같이 사용자 신분(U)과 패스워드(*Upass*)를 입력한다.

위의 인터페이스 이후에, E2ESP의 메시지 3를 WAP Server에게 전송한다.

본 논문에서는 기존의 WTLS/SSL을 사용하는 WAP 보안 환경에서 제공되지 못했던, 종단간의 보안(End-to-End Security)을 지원하는 프로토콜을 제안하였다. 이 제안된 프로토콜은 EKE 방식을 적용함으로써 사용자의 신분을 노출시키지 않고, 접근하고자 하는 WAP Server에 안전하게 로그인이 가능하며, 사용자의 신분에 기반한 기본적인 권한 설정도 가능한 장점이 있다. 또한 기존의 환경에서 인증서 검증을 위해서 요구되는 단말기의 트래픽 오버헤드를 최소화하도록 설계하였다. 제안 프로토콜은 기존의 WTLS/SSL로 구성된 보안 구조와 더불어 응용 계층에서도 운용이 가능하며, 더욱이 기존의 개별적 보안기능 없이 독자적인 종단간의 보안 세션 연결이 가능하여 WAP에서의 종단간 정보보호에 활용될 수 있을 것으로 판단된다.

참고문헌

[1] WAP Forum, *"Wireless Application Protocol Architecture Specification"*, 1998.

[2] WAP Forum, *"Wireless Application Protocol Wireless Transport Layer Security Specification"*, 1999.

[3] WAP Forum, *"Wireless Application Protocol Public Key Infrastructure Definition"*, 2000.

[4] David Wagner, Bruce Schneier, *"Analysis Of The SSL 3.0 Protocol"* Proceedings of 2nd USENIX Workshop on Electronic Commerce 2104 USENIX Press, November 1997, pp.29-40.

[5] Markku-Juhani Saarinen, *"Attack Against The WAP WTLS Protocol"* Communications and Multimedia Security Joint working conference IFIP TC6 and TC11 Katholieke Universiteit Leuven, 1999, Belgium.

[6] Sami Jormalainen, Jouni Laine, *"Security In The WTLS"*. http://www.hut.fi/~jtlaine2/wtls/, 1999.

[7] Steven M. Bellovin, *"Problem Areas For The IP Security Protocols"* Proceedings of the Sixth USENIX Security Symposium, 1996, pp. 205-214.

[8] Rolf Oppliger, *"Security Technologies For The World Wide Web"* ARTECH HOUSE. INC, 2000.

[9] Charles Arehart, Nirmal Chidambaram etc, "Professional WAP", Wrox Press Ltd, 2000, pp.10-41.

[10] Peter Buhler, Thomas Eirich, Michael Stenier, Michael Waidner, *"Secure Password-Based Cipher Suite For TLS"* In Symposium on Network and Distributed Systems Security(NDSS '00), pages 129-142, San Diego, CA, Internet Society, 2000.

[11] S. Halevi and H. Krawczyk, *"Public-Key Cryptography And Password Protocols"* In 5th ACM Conference on Computer and Communication Security", San Francisco, California. ACM Press, 1998.

[12] Steven M. Bellovin, *"Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks"* Proceedings of the IEEE Symposium on research in Security and Privacy, Oakland, May 1992.

[13] N. Haller, *"The S/KEY One-Time Password System"*, RFC 1760, Feb 1995.