情報保護學會論文誌
第12券 第3號, 2002. 6

타원 곡선에 기반한 표준 키 분배 프로토콜의 안전성 분석 및 응용 분야에 관한 연구*

오 수 현**, 이 승 우**, 심 경 아***, 양 형 규****, 원 동 호**

A Study on the Security analysis and Applications of Standard Key agreement protocols based on Elliptic curve cryptosystem

Soo-hyun Oh**, Seung-Woo Lee**, Kyung-Ah Shim***,
Hyung-Kyu Yang****, Dong-ho Won**

요 약

네트워크 상에 전송되는 메시지에 대한 기밀성을 제공하기 위해 암호 시스템의 사용이 점차 증가하고 있으며, 키 분배 프로토콜은 안전한 암호 시스템을 구현하는데 가장 필수적인 요소이다. 최근 들어 국내·외에서는 암호화 기능을 제공하는 정보 보안 제품들이 많이 개발되고 있으나, 각 제품에서 사용되는 키 분배 프로토콜의 안전성에 대한 정확한 증명없이 산발적으로 제안되고 있는 실정이다. 따라서, 본 논문에서는 최근에 발표된 표준 키 분배 프로토콜 중 타원 곡선 암호 시스템을 이용하는 ANSI X9.63의 키 분배 프로토콜들에 대해 세션키 설정 과정과 특징을 자세히 분석하고, 이를 기반으로 하여 여러 공격자 모델에 대한 각 프로토콜의 안전성을 분석하고자 한다. 또한, 키 분배 프로토콜의 주요 응용분야의 요구사항을 분석하고 각 분야에 가장 적합한 프로토콜을 제안한다.

ABSTRACT

To provide the privacy of transmitted message over network, the use of cryptographic system is increasing gradually. Because the security and reliability of the cryptographic system is totally rely on the key, the key management is the most important part of the cryptographic system. Although there are a lot of security products providing encryption, the security of the key exchange protocols used in the product are not mostly proved yet. Therefore, we have to study properties and operation of key agreement protocols based on elliptic curve in ANSI X9.63. Furthermore, we analyze the security of their protocols under passive and active attacker models and propose the most suitable application field taking the feature of the protocols into account.

Keyword : *Key agreement protocol, Elliptic curve cryptosystem, Diffie-Hellman primitive, Active attack*

Ⅰ. 서 론

개방형 네트워크상에서 전송되는 디지털 정보들은 실제 생활의 문서에 비해 무단 절취, 위·변조 또는 정보의 파기 등과 같은 위험이 훨씬 증가하게 된다. 따라서 네트워크 상에서 발생하는 안전성과 관련된

* 본 연구는 한국정보보호진흥원 위탁과제(2001-S-092)의 지원에 의해 수행하였습니다.
** 성균관대학교 정보통신공학부 정보통신보호연구실({shoh, swlee, dhwon}@dosan.skku.ac.kr)
*** 한국정보보호진흥원(KISA) 암호 기술팀(kashim@kisa.or.kr)
**** 강남대학교 컴퓨터공학과

문제를 해결하기 위해 최근 들어 암호 시스템의 사용이 점차 증가하고 있다.

암호 시스템은 송신자가 전송하는 메시지를 키(key)라는 상대적으로 작은 비밀 정보를 이용하여 암호화하여 전송하면 수신자는 대응하는 키를 이용하여 수신한 암호문을 복호하여 평문을 얻게 되는 것이다. 그러므로 암호 시스템의 안전성 및 신뢰성은 키에 절대적으로 의존하며 암호 시스템의 설계 및 구현 시 키 관리(key management)는 가장 중요한 분야 중의 하나이다.

키 관리는 키의 생성(generation), 저장(store), 분배(distribution), 파기(destroy), 폐기(revoke), 등록(register) 및 해제(deregister), 등록된 키의 확인(certification)등을 포함한다. 이 중 키 관리의 가장 근본적인 문제는 통신하는 두 당사자간에 효율적인 방법으로 비밀키를 공유하기 위한 키 분배(key distribution) 문제이다.

최근 들어 국내·외에서는 암호화 기능을 제공하는 정보 보안 제품들이 많이 개발되고 있으며 각 제품에서 사용되는 키 분배 프로토콜은 안전성에 대한 정확한 증명없이 산발적으로 제안되고 있는 실정이다. 따라서, 본 논문에서는 최근에 발표된 표준 키 분배 프로토콜 중, ANSI X9.63의 키 분배 프로토콜들에 대해 세션키 설정 과정과 특징을 자세히 분석하고 이를 기반으로 하여 몇 가지 능동적 공격자 모델에 대한 각 프로토콜의 안전성을 분석하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 키 분배 프로토콜의 개념 및 타원 곡선 암호 시스템의 개요에 대해 간단히 언급하고 3장에서는 본 논문에서 고려할 X9.63의 키 분배 프로토콜들의 세션키 설정 과정 및 특징을 분석한다. 다음으로 4장에서는 각 키 분배 프로토콜에 대해 능동적 공격자 환경에서의 안전성을 분석하고 5장에서는 키 분배 프로토콜의 특징을 고려하여 가장 적합한 응용 분야를 도출하고 마지막으로 6장에서 결론을 맺는다.

II. 연구 배경

2.1 키 분배 프로토콜의 개요

키 분배 프로토콜(key distribution protocol)이란 안전하지 않은 통신로를 이용하여 두 사용자간에 안전하고 효율적인 비밀 세션키(session key)를 공유하기 위한 메커니즘이다. 이러한 키 분배 프로토콜은 1976년 Diffie-Hellman[4]에 의해 처음으로 제안된 이후, 많은 연구가 진행되어 현재까지 다양한 프로토콜들이 제안되었다.

키 분배 프로토콜은 세션키를 설정하는 유형에 따라 크게 두 가지로 나눌 수 있다. 먼저, 사용자 U와 사용자 V가 공유하고자 하는 비밀 세션키를 어느 누구도 미리 결정하지 않고 두 사용자간의 합의에 의해 설정하는 키 동의(key agreement) 방식과 사용자 U가 비밀 세션키를 일방적으로 선택하여 세션키를 공유하고자 하는 사용자 V에게 안전하게 전송하는 키 전송(key transport)방식이 있다.

또한, 키 분배 방식은 세션키 설정을 위해 사용하는 암호 방식에 따라, 관용 암호 시스템을 이용한 방식과 공개키 암호 시스템을 이용하는 방식으로 나눌 수 있다. 관용 암호 시스템을 이용하기 위해서는 사전에 안전한 통신로를 이용하여 두 사용자간에 미리 비밀키를 공유해야하므로 개방된 통신로를 이용하는 컴퓨터 네트워크에 적용하는 데에는 어려움이 많다. 따라서, 공개키 암호 시스템을 이용한 키 분배 방식이 많이 사용되고 있다.

본 논문에서는 공개키 암호 시스템을 이용하여 세션키를 공유하는 키 분배 방식 중, ANSI X9.63[2]에서 제안한 타원 곡선 상에서의 이산대수 문제에 기반한 표준 키 분배 프로토콜들의 특징 및 안전성을 분석하고 각 프로토콜의 특징에 적합한 응용 분야에 대해 살펴보도록 한다.

2.2 타원 곡선 암호 시스템의 개요

타원 곡선(Elliptic curve)에 기반한 암호 시스템은 1985년 Kobliz와 Miller에 의해 제안되었으며[8], 최근 들어 무선 인터넷 환경이나 스마트 카드와 같은 제한된 계산능력을 갖는 하드웨어에 적합한 방식으로 주목받으면서 활발히 연구되고 있는 분야이다. 타원 곡선 암호 시스템은 동일한 안전성을 제공하는 유한체 상의 이산대수 문제에 기반한 시스템에 비해 사용하는 키의 길이가 짧고 효율적이라는 장점이 있다.

유한체 상에서의 타원 곡선이란 무한 원점(point at infinity)이라 불리는 특별한 점 O와 다음과 같은 체(field) F_q상에서 정의되는 discriminant가 0이 아닌 점들로 이루어진 집합을 말한다.

$$E(F_q) = \{(x, y) | y^2 = x^3 + ax + b \bmod q\} \cup \{O\}$$

타원 곡선에서의 이산대수 문제는 유한체 상의 이산대수 문제와 정의되는 유한군이 다를 뿐 본질적으로 동일하므로 유한체 상의 이산대수를 이용한 암호 시스템은 그대로 타원곡선의 암호 시스템으로 변환할 수 있다.

즉, Diffie-Hellman 방식과 DSA(Digital Signature Algorithm)는 각각 ECDH(Elliptic Curve Diffie-Hellman)와 ECDSA로 변환될 수 있다. 최근 들어 국내에서도 부가형 디지털 서명 방식 표준인 KCDSA(Korean Certificate-based Digital Signature Algorithm)를 타원 곡선에 적용한 EC-KCDSA의 표준화 작업이 진행중이다.

타원 곡선 상에서의 이산대수 문제와 본 논문에서 분석할 X9.63의 키 분배 프로토콜의 안전성 기반이 되는 ECDH 문제에 대한 정의는 다음과 같다.

〔정의 1〕 타원곡선 상에서의 이산대수 문제

타원 곡선 $E(F_q)$상에서 어떤 x에 대해 $A=xG$를 만족하는 점 $A(A\neq G)$를 선택하고 q, a, b, G, A로부터 자연수 x를 구하는 문제를 타원곡선 상에서의 이산대수 문제(discrete logarithm problem)라 한다.

〔정의 2〕 타원곡선 상에서의 Diffie-Hellman 문제

타원 곡선 $E(F_q)$상에서 $A=xG$, $B=yG$인 임의의 두 점 A, B가 주어졌을 때 $C=xyG$를 구하는 문제를 타원곡선 상에서의 Diffie-Hellman 문제라 한다.

Ⅲ. 타원 곡선에 기반한 표준 키 분배 프로토콜

3.1 ANSI X9.63의 키 분배 프로토콜

3.1.1 기호 정의

각각의 키 분배 프로토콜에 대한 분석에 앞서 ANSI X9.63에서 사용하는 구성 요소에 대해 살펴본다. 프로토콜 설명에 사용하는 기호는 다음과 같이 정의한다.

〔ANSI X9.63의 기호 정의〕

- p : 사용되는 기반 필드 F_q의 크기 ($q=p$)
- a, b : F_q의 타원 곡선 상의 임의의 원소 ($4a^3+27b^2\neq 0$)
- E : a와 b에 의해서 정의된 F_q 상의 타원 곡선 ($E : y^2=x^3+ax+b$)
- n : G의 위수(order)
- h : cofactor, $h=\#E(F_q)\ /\ n$
- 점 Q : $(x_Q, y_Q)=dG$ (d: 비밀키, Q:공개키)
- P : 타원 곡선상의 한 점
- x_P, y_P : 점 P의 x, y coordinate
- z : 공유 비밀 정보
- $x \| y$: 스트링 x와 y의 연접
- U : 키 분배 프로토콜의 시행자(initiator)
- V : 키 분배 프로토콜의 응답자(recipient)
- $(d_{e,u}, Q_{e,u})/(d_{e,v}, Q_{e,v})$: U/V의 일회용 키 쌍
- $(d_{s,u}, Q_{s,u})/(d_{s,v}, Q_{s,v})$: U/V의 고정된 키 쌍
- QEU/QEV : U/V의 일회용 공개키

3.1.2 사용하는 프리미티브

타원 곡선에 기반한 키 동의 프로토콜의 시스템 설정을 위해서는 도메인 파라미터 생성 프리미티브와 키 생성 프리미티브가 필요하다. 도메인 파라미터 생성 프리미티브는 타원 곡선이 정의되는 체의 형태에 따라 F_p(p는 3보다 큰 소수)와 F_{2^m}으로 나누어진다.

그리고 두 사용자 U, V사이에 세션키를 설정하기 위해 Diffie-Hellman 프리미티브, MQV 프리미티브와 보조 함수로 Associate value 함수(avf), 해쉬 함수, 키 유도 함수(kdf) 등을 사용한다. 본 논문에서는 Diffie-Hellman 프리미티브를 이용하여 세션키를 설정하는데 필요한 프리미티브와 보조 함수에 대해서만 언급하기로 한다.

(가) 도메인 파라미터 생성 프리미티브

도메인 파라미터 생성 프리미티브는 키 동의 프로토콜의 시스템 설정 과정에 필요한 소수 p와 타원 곡선 E, 위수가 n인 점 G 등을 생성하는 프리미티브이다.

① F_p 상에서의 도메인 파라미터 생성 프리미티브

- 입력 : 없음
- 실행 과정
  - 3보다 큰 소수 p 선택(field size)
- 출력 : 길이가 'keydatalen'인 KeyData
  - field size $q=p$
  - 타원 곡선 E : $y^2=x^3+ax+b$를 정의하는 F_p 상에서 임의의 원소 a, b
  - E상에서 위수가 소수인 점 $G=(x_G, y_G)$를 정의하는 두 원소 x_G, y_G

- n : 점 G의 위수(order)
- cofactor $h=\#E(F_p) / n$

② F_{2^m}상에서의 도메인 파라미터 생성 프리미티브

- 입력 : 없음
- 실행 과정
  - 2^m 선택(field size)
  - F_{2^m}상의 원소를 표현할 기저(basis)를 TPB (Trinomial basis), PPB(Pentanomial basis), GNB(Gaussian nomal basis) 중에서 선택
- 출력
  - field size $q= 2^m$
  - 원소를 표현하는데 사용할 기저(TPB와 PPB를 선택한 경우에는 F_2상의 m차 기약 다항식 f(x)도 출력)
  - 타원곡선 E : $y^2+xy=x^3+ax^2+b$를 정의하는 F_{2^m}상에서 임의의 원소 a, b
  - E상에서 위수가 소수인 점 $G=(x_G, y_G)$를 정의하는 두 원소 x_G, y_G
  - n : G의 위수(order)
  - cofactor $h=\#E(F_{2^m}) / n$

(나) 키 생성 프리미티브

- 입력
  - 도메인 파라미터 q, a, b, x_G, y_G, n, h
  - 사용할 기저($q= 2^m$인 경우)
- 실행 과정
  ① 랜덤한 정수 $d\in[1, n-1]$ 선택
  ② 점 $Q=(x_Q, y_Q)=dG$ 계산
- 출력 : (비밀키, 공개키)=(d, Q)

(다) Diffie-Hellman 프리미티브

Diffie-Hellman 프리미티브는 자신의 비밀키와 상대방의 공개키를 이용하여 공유 비밀 정보(shared secret value) z를 생성하는 프리미티브이다. 공유 비밀 정보 z의 계산에 cofactor 곱셈 이용하게 되는데, 이는 small subgroup attack을 방지하기 위하여 사용된다.[7] Diffie-Hellman 프리미티브의 수행 과정은 다음과 같다.

- 도메인 파라미터 : q, a, b, G, n, h
- 입력 : 비밀키 d, 공개키 Q
- 실행 과정
  ① 점 $P=hdQ$를 계산
  ② $P\neq 0$인지 검사하고 0이면 중단(stop)
  ③ $z=x_P$ (x_P는 P의 x-coordinate)
- 출력 : z

(라) 보조 함수

① 해쉬 함수

해쉬 함수는 임의의 비트 스트링에 대해 고정된 해쉬 값을 출력하는 함수로 키 유도 함수에서 사용한다.

- 전제 사항
  - hashmaxlen : 해쉬 함수의 최대 입력길이
  - hashlen : 해쉬 함수의 출력 길이
- 입력
  - Data : hashmaxlen 이하의 비트 스트링
- 실행 과정
  - hash = H(Data) 계산
- 출력
  - 길이가 hashlen인 비트 스트링 hash

② 키 유도 함수(kdf)

키 유도 함수는 공유 비밀 정보(z)로부터 키 생성 데이터를 만들 때 사용하는 함수이다.

- 입력
  - z : 공통의 비밀 정보
  - 정수 keydatalen : keying data의 길이
  - SharedInfo (선택 사항)
- 실행 과정
  1) counter=00000001(16진수)로 설정
  2) For I=1 to $\lceil$ keydatalen / hahslen $\rceil$
     i) $Hash_i=H(Z||$ counter $||$ [SharedInfo])
     ii) Increment counter
     iii) Increment I
  3) keydatalen / hahslen=정수이면,
     $Hash!_{\lceil keydatalen/hahslen \rceil}=Hash_{\lceil keydatalen/hahslen \rceil}$
     정수가 아니면,
     $Hash!_{\lceil keydatalen/hahslen \rceil}=Hash_{\lceil keydatalen/hahslen \rceil}$
     의 최상위(keydatalen-(hashlen× $\lfloor$keydatalen / hahslen $\rfloor$)) 비트
  4) $KeyData=Hash_1||Hash_2||...||Hash_{\lceil keydatalen/hahslen-1 \rceil}||Hash!_{\lceil keydatalen/hahslen \rceil}$
- 출력 : 길이가 keydatalen인 KeyData

〔표 1〕 ANSI X9.63의 키 동의 프로토콜

	고정 데이터 (static data)	일회용 데이터 (ephemeral data)	공유 비밀 값 (shared secret value)	세션키 (session key)
①	N/A	• 도메인 파라미터 :$q_e, a_e, b_e, G_e, n_e, h_e$ • 사용자 U의 키 쌍 :$(d_{e,u}, Q_{e,u})$ • 사용자 V의 키 쌍 :$(d_{e,v}, Q_{e,v})$	$z_e = DH(d_{e,u}, Q_{e,v})$ $= DH(d_{e,v}, Q_{e,u})$	KeyData = kdf(z_e, 〔ShredInfo〕)
②	• 도메인 파라미터 :q, a, b, G, n, h • 사용자 V의 키 쌍 :$(d_{s,v}, Q_{s,v})$	• 사용자 U의 키 쌍 :$(d_{e,u}, Q_{e,u})$	$z_e = DH(d_{e,u}, Q_{s,v})$ $= DH(d_{s,v}, Q_{e,u})$	KeyData = kdf(z_e, 〔ShredInfo〕)
③	• 도메인 파라미터 :$q_s, a_s, b_s, G_s, n_s, h_s$ • 사용자 U의 키 쌍 :$(d_{s,u}, Q_{s,u})$ • 사용자 V의 키 쌍 :$(d_{s,v}, Q_{s,v})$	N/A	$z_s = DH(d_{s,u}, Q_{s,v})$ $= DH(d_{s,v}, Q_{s,u})$	KeyData = kdf(z_s, 〔ShredInfo〕)
④	• 도메인 파라미터 :q, a, b, G, n, h • 사용자 U의 키 쌍 :$(d_{s,u}, Q_{s,u})$ • 사용자 V의 키 쌍 :$(d_{s,v}, Q_{s,v})$	• 사용자 U의 키 쌍 :$(d_{e,u}, Q_{e,u})$	$z_e = DH(d_{e,u}, Q_{s,v})$ $= DH(d_{s,v}, Q_{e,u})$ $z_s = DH(d_{s,u}, Q_{s,v})$ $= DH(d_{s,v}, Q_{s,u})$ $Z = z_e \mid\mid z_s$	KeyData = kdf(Z, 〔ShredInfo〕)
⑤	• 도메인 파라미터 :$q_s, a_s, b_s, G_s, n_s, h_s$ • 사용자 U의 키 쌍 :$(d_{s,u}, Q_{s,u})$ • 사용자 V의 키 쌍 :$(d_{s,v}, Q_{s,v})$	• 도메인 파라미터 :$q_e, a_e, b_e, G_e, n_e, h_e$ • 사용자 U의 키 쌍 :$(d_{e,u}, Q_{e,u})$ • 사용자 V의 키 쌍 :$(d_{e,v}, Q_{e,v})$	$z_e = DH(d_{e,u}, Q_{e,v})$ $= DH(d_{e,v}, Q_{e,u})$ $z_s = DH(d_{s,u}, Q_{s,v})$ $= DH(d_{s,v}, Q_{s,u})$ $Z = z_e \mid\mid z_s$	KeyData = kdf(Z, 〔ShredInfo〕)
⑥	• 도메인 파라미터 :$q_s, a_s, b_s, G_s, n_s, h_s$ • 사용자 U의 키 쌍 :$(d_{s,u}, Q_{s,u})$ • 사용자 V의 키 쌍 :$(d_{s,v}, Q_{s,v})$	• 도메인 파라미터 :$q_e, a_e, b_e, G_e, n_e, h_e$ • 사용자 U의 키 쌍 :$(d_{e,u}, Q_{e,u})$ • 사용자 V의 키 쌍 :$(d_{e,v}, Q_{e,v})$	$z_s = DH(d_{s,u}, Q_{s,v})$ $= DH(d_{s,v}, Q_{s,u})$ $z_e = DH(d_{e,u}, Q_{e,v})$ $= DH(d_{e,v}, Q_{e,u})$	KeyData =kdf(z_e, 〔ShredInfo〕)
⑦	• 도메인 파라미터 :$q_s, a_s, b_s, G_s, n_s, h_s$ • 사용자 U의 키 쌍 :$(d_{s,u}, Q_{s,u})$ • 사용자 V의 키 쌍 :$(d_{s,v}, Q_{s,v})$	• 도메인 파라미터 :$q_e, a_e, b_e, G_e, n_e, h_e$ • 사용자 U의 키 쌍 :$(d_{e,u}, Q_{e,u})$ • 사용자 V의 키 쌍 :$(d_{e,v}, Q_{e,v})$	$z_e = DH(d_{e,u}, Q_{e,v})$ $= DH(d_{e,v}, Q_{e,u})$ $z_s = DH(d_{s,u}, Q_{s,v})$ $= DH(d_{s,v}, Q_{s,u})$ $z = z_e \mid\mid z_s$	KeyData! =kdf(z, 〔ShredInfo〕) =Mackey || KeyData 세션키 = KeyData
⑧	• 서명용 도메인 파라미터 :$q_{sig}, a_{sig}, b_{sig}, G_{sig}, n_{sig}, h_{sig}$ • 사용자 U의 서명키 쌍 : :$(d_{sig,u}, Q_{sig,u})$ • 사용자 V의 서명키 :$(d_{sig,v}, Q_{sig,v})$	• 도메인 파라미터 :$q_e, a_e, b_e, G_e, n_e, h_e$	$z_e = DH(d_{e,u}, Q_{e,v})$ $= DH(d_{e,v}, Q_{e,u})$	KeyData! =kdf(z_e, 〔ShredInfo〕) =Mackey || KeyData 세션키 = KeyData

① Ephemeral unified model
② 1-pass Diffie-Hellman
③ Static unified model
④ 1-pass unified model
⑤ Full unified model
⑥ Combined unified model with Key confirmation
⑦ Full unified model with Key confirmation
⑧ Station-to-Station

3.1.3 키 분배 과정

ANSI X9.63에서는 11개의 키 동의 프로토콜과 3개의 키 전송 프로토콜을 정의하고 있다. 11개의 키 동의 프로토콜 중, Diffie-Hellman 프리미티브를 사용하는 방식에는 Ephemeral unified model, 1-pass Diffie-Hellman, Static unified model, Combined unified model with Key confirmation, 1-pass unified model, Full unified model, Full unified model with key confirmation, Station-to-Station 등 8개의 프로토콜이 있으며, MQV 프리미티브를 이용하는 방식에는 1-pass MQV, Full MQV, Full MQV with key confirmation 등 3개의 프로토콜이 있다. 본 논문에서는 Diffie-Hellman 프리미티브를 이용하는 8가지 키 동의 프로토콜에 대해서만 언급한다.

〔표 1〕은 8개의 키 동의 프로토콜의 세션키 설정에 필요한 고정 데이터, 일회용 데이터, 공유 비밀 값 및 세션키 계산 과정을 간략히 정리한 것이다.

먼저, Ephemeral unified model은 사용자 U, V가 일회용 키 쌍을 이용하여 세션키를 설정하는 방식으로 X9.63에서 정의하는 프로토콜 중 가장 기본이 되는 방식이다. 1-pass Diffie-Hellman 프로토콜은 사용자 U는 일회용 키 쌍을 이용하고 사용자 V는 고정된 키 쌍을 이용하는 방식이며, Static unified model은 두 사용자 모두 고정된 키 쌍을 이용하여 세션키를 설정하는 방식이다.

Combined unified model with Key confirmation 방식은 두 사용자가 일회용 키 쌍과 고정된 키 쌍을 모두 이용하며, MAC(Message Authentication Code)을 이용하여 두 사용자 사이에 공통된 키가 설정되었음을 확인하는 키 확인을 제공하는 방식이다. 1-pass unified model은 사용자 U는 일회용 키 쌍과 고정된 키 쌍을 모두 이용하고 사용자 V는 고정된 키 쌍만을 이용하는 방식이고, full unified model은 두 사용자 모두 일회용 키 쌍과 고정된 키 쌍을 이용하여 세션키를 설정하는 방식이다. 또한, Full unified model with key confirmation 방식은 Combined unified model with Key confirmation 방식을 변경한 것으로 세션키 설정 과정이 거의 유사하다. 그러나 공유 비밀 정보인 KeyData를 생성할 때, 고정된 공유 비밀 정보와 일회용 공유 비밀 정보를 연접하여 생성하는 것이 다르다.

마지막으로, Station-to-Station 프로토콜은 Ephemeral unified model에 개체 인증과 명시적 키 인증을 제공하기 위해 디지털 서명을 추가한 방식이다.

〔표 2〕는 프로토콜 ⑥, ⑦, ⑧에서 키 확인이나 개체 인증을 제공하기 위해 생성하거나 검증하는 인증 데이터의 형태를 나타낸 것이다.

3.2 키 동의 프로토콜의 특징 분석

본 논문에서는 앞 절에서 설명한 각 키 동의 프로토콜에 대해 세션키 설정에 필요한 통신 회수와 개체 인증(entity authentication), 키 확인(key confirmation), 묵시적 키 인증(implicit key authentication), key freshness 등을 제공하는지에 대해 분석하였다.

각각에 대한 정의는 다음과 같고, X9.63의 키 동의 프로토콜의 특징을 분석한 결과는 〔표 3〕과 같다.

- **n-pass 프로토콜** : 한 사용자 입장에서 세션키를 설정하기 위해 상대방과 n번의 통신이 필요한 프로토콜

〔표 2〕 키 확인 및 개체 인증을 위한 데이터의 형태

	사용자 V	사용자 U
⑥	$MacData_1 = 02_{16}\|\|V\|\|U\|\|QEU\|\|QEV\|\|text_1$ $MacTag_1 = MAC_{MacKey}(MacData_1)$	$MacData_2 = 03_{16}\|\|U\|\|V\|\|QEV\|\|QEU\|\|text_2$ $MacTag_2 = MAC_{MacKey}(MacData_2)$
⑦	$MacData_1 = 02_{16}\|\|V\|\|U\|\|QEV\|\|QEU\|\|text_1$ $MacTag_1 = MAC_{MacKey}(MacData_1)$	$MacData_2 = 03_{16}\|\|U\|\|V\|\|QEV\|\|QEU\|\|text_2$ $MacTag_2 = MAC_{MacKey}(MacData_2)$
⑧	$Data_1 = QEV\|\|QEU\|\|U\|\|text_1$ $Data_1$에 대한 디지털 서명 생성 $= (rsig_1, ssig_1)$ $MacTag_1 = MAC_{MacKey}(Data_1)$	$Data_2 = QEU\|\|QEV\|\|V\|\|text_2$ $Data_2$에 대한 디지털 서명 생성 $= (rsig_2, ssig_2)$ $MacTag_2 = MAC_{MacKey}(Data_2)$

⑥ Combined unified model with Key confirmation
⑦ Full unified model with Key confirmation
⑧ Station-to-Station

〔표 3〕 ANSI X9.63의 타원곡선 기반 키 분배 프로토콜의 특징

	통신 회수	개체 인증	키 확인	묵시적 키인증	Key freshness
①	2	-	-	-	양방향
②	1	-	-	일방향	일방향
③	0	-	-	양방향	-
④	3	-	양방향	양방향	양방향
⑤	1	-	-	양방향	일방향
⑥	2	-	-	양방향	양방향
⑦	3	-	양방향	양방향	양방향
⑧	3	양방향	양방향	양방향	양방향

① Ephemeral Unified Model
② 1-pass Diffie-Hellman
③ Static unified model
④ Combined unified model with Key confirmation
⑤ 1-pass unified model
⑥ Full unified model
⑦ Full unified model with Key confirmation
⑧ Station-to-Station

- **개체 인증(Entity authentication)** : 키 분배 프로토콜에 참여하고 있는 상대방의 신원을 확인하는 것으로 명시적 키 인증(explicit key authentication)에 의해 제공될 수 있음
- **키 확인 (key confirmation)** : 키 분배 프로토콜에 참여한 합법적인 사용자가 자신이 의도한 상대방과 실제로 공통의 비밀 세션키를 공유하였음을 확인
- **묵시적 키 인증(implicit key authentication)** : 키의 소유 여부는 알려져 있지 않다고 하더라도 키 분배 프로토콜에 참여한 상대방만이 세션키를 계산할 수 있음을 보장
- **Key freshness** : 세션마다 설정된 키가 바뀜

Ⅳ. 안전성 분석

4.1 공격자 모델

암호 프로토콜의 안전성을 증명하기 위해서는 먼저 대상이 되는 공격자 모델을 정의해야 한다. 암호 프로토콜에 대한 공격자는 크게 수동적 공격자(passive attacker)와 능동적 공격자(active attacker)로 나눌 수 있으며, 각각의 특징은 다음과 같다.

- **수동적 공격자** : 프로토콜의 참가자와 실제로 통신에 참여하지 않고 두 참가자 사이의 통신 내용을 도청(eavesdropping)함으로써 공격을 수행하는 공격자
- **능동적 공격자** : 단순히 참가자들의 통신 내용을 도청하는 것뿐만 아니라 전송되는 메시지를 위·변조하거나 새로운 메시지를 삽입하거나 하는 등 실제 통신에 참여하는 보다 강력한 공격자

본 논문에서 고려할 능동적 공격자 모델은 active impersonation 공격자, forward secrecy에 대한 공격자, key compromise impersonation 공격자, known key security에 대한 공격자이며 각각에 대한 정의는 다음과 같다.

[정의 3] Active Impersonation(AI) attack

공격자가 자신을 임의의 다른 사용자로 위장하여 프로토콜에 참여하고, 정당한 사용자 U와 키 분배를 성공적으로 수행하는 경우에 *active impersonation*이 가능하다고 한다.

[정의 4] Forward Secrecy(FS)

사용자 U와 V의 비밀키가 노출되더라도, 공격자가 두 사용자 사이에 설정된 과거 세션키를 계산할 수 없는 경우에 *forward secrecy*를 만족한다고 한다. forward secrecy는 노출되는 사용자의 키에 따라 다음과 같이 나눌 수 있다.

- *Half Forward Secrecy* : 한 사용자의 비밀키가 노출된 경우에만 세션키가 안전
- *Full Forward Secrecy* : 두 사용자의 비밀키가 모두 노출된 경우에도 세션키 안전

[정의 5] Key-Compromise Impersonation(KCI) attack

사용자 U의 비밀키가 노출되었을 때, 공격자 E가 누구에게나 사용자 U로 위장할 수 있고 사용자

U에게 임의의 사용자 V로 위장할 수 있을 때 *key-compromise impersonation* 가능하다고 한다. 그러나, 공격자 E가 누구에게나 사용자 U로 위장할 수 있지만 사용자 U에게 임의의 다른 사용자로 위장할 수는 없는 경우에는 키 분배 프로토콜이 *key-compromise impersonation resilience* 특성을 갖는다고 한다.

[정의 6] Known Key Security(KKS)

두 사용자 U, V 사이의 과거 세션키가 노출되더라도 현재 세션키의 안전성에는 아무런 영향을 미치지 않는 경우에 *Known Key Security*를 만족한다고 한다. Known Key Security에 대한 공격은 다음과 같이 두 가지로 나눌 수 있다.

- *KKP(Known Key Passive)* 공격 : 과거의 세션키와 전송 정보 및 현재 세션의 전송 정보를 이용하여 현재의 세션키를 획득하려는 공격 방법
- KKI(Known Key Impersonation) 공격 : 세션에 직접 참여하여 과거의 세션키와 전송 정보 그리고 현재 세션의 전송 정보를 이용하여 사용자 U에게 사용자 V로 위장하여 세션키를 설정하려는 공격 방법

4.2 안전성 분석 결과

4.2.1 수동적 공격자에 대한 안전성

X9.63에서 제안하고 있는 키 동의 프로토콜은 기본적으로 그 안전성이 ECDH 문제에 기반하므로 수동적 공격자가 공개 정보와 전송 정보를 이용하여 세션키를 구하는 어려움은 ECDH 문제를 푸는 어려움과 동일하다.

4.2.2 능동적 공격자에 대한 안전성

■ **Active Impersonation에 대한 안전성**

Ephemeral unified model에서는 두 사용자 사이에 세션키를 설정하기 위해 매 세션마다 다르게 선택한 일회용 키 쌍을 이용하므로 상대방에 대한 어떠한 인증도 제공하지 않는다. 따라서, 공격자가 임의의 다른 사용자로 위장하는 것이 가능하므로 active impersonation 공격에 대해 안전하지 않다.

그리고 1-pass Diffie-Hellman 프로토콜은 1-pass 프로토콜로 사용자 U에게는 묵시적 키 인증을 제공하지만 사용자 V에게는 어떤 형태의 인증도 제공하지 않는다. 따라서 공격자 E가 사용자 V에게 사용자 U로 위장하여 세션키를 설정하는 공격이 가능하므로 역시 active impersonation에 대해 안전하지 않다.

반면에, Static unified model은 두 사용자 사이에 전송 정보의 교환없이 공개키 인증서를 이용하여 세션키를 설정하는 방식이므로, 공격자가 세션에 직접 참여하여 다른 사용자로 위장하는 active impersonation 공격을 수행할 수 없다.

또한, Combined unified model with Key confirmation 프로토콜은 상대방의 개체 인증을 위해 사용자들의 고정된 키 쌍을 이용하고 세션키의 생성에는 매 세션마다 다르게 선택한 일회용 키 쌍을 이용한다. 따라서, 정당한 사용자 V의 비밀키를 알지 못하는 공격자가 프로토콜에 직접 참여하여 사용자 U에게 사용자 V로 위장하기 위해서는 ECDH 문제를 해결해야 한다. 즉, 사용자 V로 위장하여 프로토콜에 직접 참여한 공격자는 일회용 키 쌍을 이용하여 생성한 세션키는 계산할 수 있지만, 사용자 V의 비밀키를 알지 못하므로 인증 정보인 MacTag를 생성할 수 없다. 정당한 MagTag를 생성하기 위해서는 두 사용자의 고정된 공개키 $Q_{s,u}$, $Q_{s,v}$로부터 z_s를 계산해야 하므로 ECDH 문제를 해결해야 한다. 따라서, Combined unified model with Key confirmation 프로토콜에 대한 AI 공격자의 어려움은 ECDH 문제의 어려움과 동치이다.

다음으로, 1-pass unified model에서는 세션키를 설정하기 위해 사용자 U는 일회용 키 쌍과 고정된 키 쌍을 모두 이용하고 사용자 V는 고정된 키 쌍만을 이용한다. 따라서 이 방식에 대한 AI 공격자는 $z_e = DH(d_{e,u}, Q_{s,v})$는 계산할 수 있지만 사용자 U의 비밀키를 모르므로 $z_s = DH(d_{s,u}, Q_{s,v})$를 계산할 수 없게 된다. 따라서, 1-pass unified model에 대한 AI 공격자의 어려움 역시 ECDH 문제의 어려움과 동치이다.

Full unified model과 Full unified model with Key confirmation 프로토콜은 공격자가 생성한 일회용 키 쌍에 대해서는 z_e를 계산할 수 있지만 공격자는 사용자 V의 비밀키를 알지 못하므로 z_s를 계산하기 위해서는 ECDH 문제를 해결해야 한다. 그러므로 1-pass unified model과 같이 이들 방식에 대한 AI 공격자의 어려움도 ECDH 문제의 어려움과 동치이다.

마지막으로, Station-to-Station 프로토콜은 세션키 생성에 참여한 상대방의 개체 인증을 위해 디지털 서명을 이용하므로, 안전한 디지털 서명 방식을 이용하면 공격자가 정당한 사용자로 위장하는 active impersonation 공격은 불가능하다.

■ **Key Compromise Impersonation에 대한 안전성**

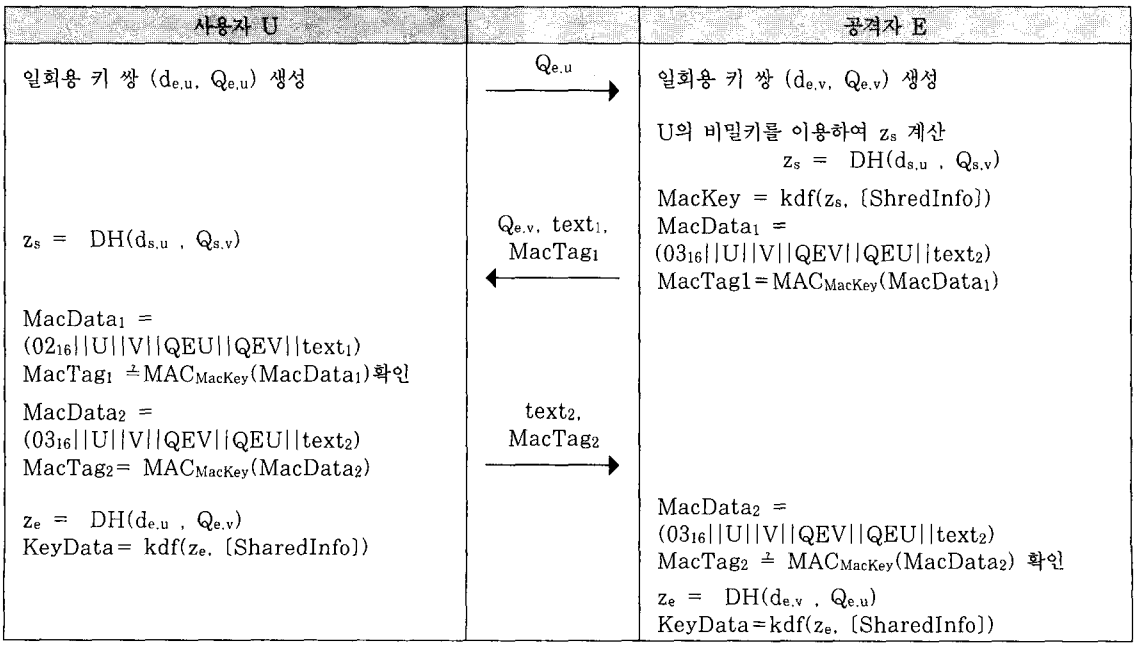
Ephemeral unified model에서는 사용자의 고정된 비밀키가 세션키 생성에 사용되지 않으므로 사용자의 비밀키가 노출되더라도 공격자에게는 아무런 도움이 되지 않는다. 그러나 이 방식은 본래 공격자가 다른 사용자로 위장하는 것이 가능하므로 key compromise impersonation resilience 특성을 갖지 못한다.

1-pass Diffie-Hellman 프로토콜 역시 본래 공격자가 정당한 사용자로 위장하는 것이 가능하므로 key compromise impersonation resilience 특성을 갖지 못한다.

Static unified model은 세션키를 생성하기 위해 두 사용자의 고정된 비밀키만을 이용하므로 사용자 U의 비밀키가 노출되는 경우, 누구든지 사용자 U로 위장할 수 있을 뿐만 아니라 공격자는 사용자 U에게 임의의 다른 사용자로 위장할 수 있으므로 key compromise impersonation resilience 특성을 갖지 못한다.

Combined unified model with Key confirmation 프로토콜에서는 사용자 U의 비밀키가 노출되는 경우에 공격자는 사용자 U로 위장할 수 있을 뿐만 아니라 사용자 U에게 임의의 정당한 사용자로 위장하는 것이 가능하다. 사용자 U의 고정된 비밀키 $d_{s,u}$를 획득한 공격자가 사용자 V로 위장하여 세션키를 설정하는 공격 과정은 [그림 1]과 같다. 따라서, 이 방식은 key compromise impersonation resilience 특성을 갖지 못한다.

1-pass unified model에서는 key compromise impersonation 공격에 대한 안전성은 두 가지 경우로 나누어 생각할 수 있다. 첫 번째 경우는 공격자가 세션을 시작하는 경우로, 공격자가 일회용 키 쌍을 생성하여 사용자 V로 위장하여 사용자 U에게 전송하는 경우이다. 이러한 경우에는 공격자는 자신이 생성한 일회용 비밀키와 사용자 U의 공개키를 이용하여 z_e를 계산하고 사용자 V의 공개키와 획득한 사용자 U의 비밀키를 이용하여 z_s를 계산할 수 있으므로 사용자 V로 위장하는 것이 가능하다. 그러나 사용자 U가 먼저 세션을 시작하여 일회용 키 쌍을 생성하여 전송하는 경우에는 공격자가 사용자 V의 비밀키를 알지 못하므로 z_e를 계산할 수 없다. 따라서, key compromise impersonation resilience



[그림 1] Combined unified model with Key confirmation 프로토콜에 대한 KCI 공격 과정

사용자 U		공격자 E
일회용 키 쌍 $(d_{e,u}, Q_{e,u})$ 생성	$Q_{e,u}$ →	
	← $Q_{e,v}$	일회용 키 쌍 $(d_{e,v}, Q_{e,v})$ 생성
$z_e = DH(d_{e,u}, Q_{e,v})$		$z_e = DH(d_{e,v}, Q_{e,u})$
$z_s = DH(d_{s,u}, Q_{s,v})$		$d_{s,u}$를 이용하여 다음 계산
$Z = z_e \mid\mid z_s$		$z_s = DH(d_{s,u}, Q_{s,v})$
KeyData = kdf(Z, [ShredInfo])		$Z = z_e \mid\mid z_s$
		KeyData = kdf(Z, [ShredInfo])

[그림 2] Full unified model에 대한 KCI 공격 과정

특성을 갖게 된다.

Full unified model과 Full unified model with Key confirmation 프로토콜에서는 사용자 U의 비밀키가 노출되는 경우, 공격자는 이를 이용하여 사용자 U로 위장할 수 있을 뿐만 아니라 사용자 U에게 임의의 사용자로 위장할 수도 있다. 따라서, key compromise impersonation resilience 특성을 갖지 못한다. Full unified model에 대한 KCI 공격 과정은 [그림 2]와 같고, Full unified model with Key confirmation 프로토콜에 대한 공격 과정도 이와 유사하다.

Station-to-Station 프로토콜은 사용자 U의 비밀키가 노출되는 경우에 공격자가 사용자 U로 위장하는 것은 가능하다. 그러나 공격자가 사용자 U에게 임의의 정당한 사용자로 위장하기 위해서는 그 사용자의 디지털 서명을 위조할 수 있어야 하므로 안전한 디지털 서명 방식을 이용하는 경우에는 불가능하게 된다. 따라서, 이 프로토콜은 key compromise impersonation resilience 특성을 갖는다.

■ **Forward secrecy에 대한 안전성**

Ephemeral unified model에서는 세션키를 설정하기 위해 각 사용자들이 매 세션마다 다르게 선택한 일회용 키 쌍을 이용하고, 사용자의 고정된 비밀키가 세션키 생성에 사용되지 않으므로 두 사용자의 고정된 비밀키가 모두 노출되더라도 공격자에게는 아무런 도움이 되지 않는다. 즉, 두 사용자의 비밀키가 모두 노출되더라도 공격자가 현재의 세션키를 구하는 어려움은 수동적 공격자와 동일하므로 full forward secrecy를 제공한다.

1-pass Diffie-Hellman 프로토콜은 사용자 U의 비밀키가 세션키를 생성하는데 포함되지 않으므로 사용자 U의 비밀키가 노출되더라도 세션키의 안전성에는 아무런 영향이 없지만, 사용자 V의 비밀키가 노출되는 경우에는 누구든지 두 사용자 사이의 세션키를 계산할 수 있으므로 half forward secrecy만을 제공한다.

Static unified model의 세션키 형태는 $z_s = DH(d_{s,u}, Q_{s,v}) = DH(d_{s,v}, Q_{s,u})$이므로, 두 사용자 중 한사람의 비밀키만 노출되더라도 두 사용자 사이의 세션키를 누구든지 쉽게 계산할 수 있다. 따라서, 어떠한 형태의 forward secrecy도 제공하지 않는다.

Combined unified model with Key confirmation 프로토콜에서는 사용자들의 고정된 키 쌍은 개체 인증을 수행하는데 사용하고, 실제 세션키를 생성하기 위해서는 매 세션마다 각 사용자들이 다르게 선택한 일회용 키 쌍을 이용한다. 그러므로 사용자 U, V의 고정된 비밀키가 모두 노출되더라도 현재의 세션키의 안전성에는 아무런 영향을 미치지 않으므로, full forward secrecy를 제공한다.

1-pass unified model은 1-pass Diffie-Hellman 프로토콜과 같이 사용자 V의 비밀키가 노출되는 경우에는 누구든지 이를 이용하여 사용자 U, V 사이의 세션키를 계산할 수 있지만, 사용자 U의 고정된 비밀키가 노출된 경우에도 U의 일회용 비밀키를 알지 못하므로 이를 이용하여 세션키를 계산할 수 없다. 따라서, half forward secrecy를 제공한다.

Full unified model과 Full unified model with Key confirmation 프로토콜은 세션키를 설정하기 위해 사용자 U, V의 일회용 키 쌍과 고정된 키 쌍을 모두 이용하므로 사용자 U, V의 고정된 비밀키가 모두 노출되더라도 세션키의 안전성에는 영향이 없다. 즉, 사용자 U, V의 비밀키가 노출된 경우에 공격자는 이를 이용하여 z_s를 쉽게 계산할 수 있지만 일회용 키로부터 생성된 z_e를 구하기 위해서는 ECDH 문제를 해결해야 한다. 그러므로 이들

방식은 full forward secrecy를 제공한다.

Station-to-Station 프로토콜에서는 두 사용자의 고정된 비밀키는 상대방의 개체 인증을 위해서만 사용하고, 실제 세션키의 설정에는 두 사용자가 매 세션마다 다르게 선택한 일회용 키 쌍을 이용한다. 따라서 사용자 U, V의 비밀키가 모두 노출되더라도 과거 세션키의 안전성에는 아무런 영향을 미치지 않으므로 full forward secrecy를 제공한다.

■ **Known Key attack에 대한 안전성**

Ephemeral unified model에서는 사용자들이 매 세션마다 다르게 선택한 일회용 키 쌍을 이용하여 세션키를 생성하므로 이전 세션의 전송 정보와 세션키가 노출되더라도 KKP 공격자가 현재의 세션키를 구하는데는 아무런 도움이 되지 않는다. 즉, KKP 공격자의 어려움은 아무런 정보가 주어지지 않은 수동적 공격자의 어려움과 동일하다. 그러나, 이 방식은 프로토콜에 참여하는 상대방에 대한 인증을 제공하지 않으므로 KKI 공격에 대해서는 안전하지 않다.

1-pass Diffie-Hellman 프로토콜 역시 매 세션마다 다른 랜덤 수를 이용하여 세션키를 생성하므로, 이전 세션의 전송 정보와 세션키가 알려지더라도 KKP 공격자가 현재의 세션키를 구하는 어려움은 수동적 공격자의 어려움과 동일하다. 그러나 사용자 V에게 사용자 U에 대한 인증을 제공하지 않으므로 누구든지 사용자 U로의 위장이 가능하므로 KKI 공격에 대해서는 안전하지 않다.

Static unified model은 두 사용자 사이에 설정되는 세션키가 항상 일정하므로 이전 세션의 세션키가 노출되는 경우, 공격자는 현재 세션의 사용자 U, V 사이의 세션키도 알게 된다. 따라서, KKP 공격에 대해 안전하지 않다. 또한, 이 방식에서는 공격자가 이전의 세션키를 획득하는 경우 정당한 사용자로 위장할 수 있으므로 KKI 공격에 대해서도 안전하지 않다.

Combined unified model with Key confirmation 프로토콜에서는 매 세션마다 다르게 선택된 일회용 키 쌍을 이용하여 세션키를 설정하므로 공격자가 과거의 전송 정보와 세션키를 획득하더라도 현재의 세션키를 구하는데는 아무런 도움이 되지 않는다. 즉, KKP 공격자의 어려움은 수동적 공격자의 어려움과 동일하다. KKI 공격자의 경우에도 과거의 전송 정보를 재 사용하여 사용자 V로 위장하려는 경우에, 사용자 U가 새로운 일회용 키 쌍을 생성하게 되면 이에 따라 $MacData_1$의 내용이 변경되므로 공격자는 정당한 MacTag를 생성할 수 없게 된다. 그러므로 공격자가 과거의 전송 정보와 세션키를 획득하더라도 KKI 공격자의 어려움은 아무런 정보도 주어지지 않은 AI 공격자의 어려움과 동일하다.

1-pass unified model에서는 세션키를 생성하기 위해 매 세션마다 다르게 선택된 사용자 U의 일회용 키를 이용하므로 이전 세션의 전송 정보와 세션키가 노출되더라도 이를 이용하여 현재의 세션키를 구하기 위해서는 ECDH 문제를 해결해야 한다. 즉, KKP 공격자의 어려움은 수동적 공격자의 어려움과 동일하다. 그러나 과거의 전송 정보와 세션키를 획득한 공격자가 프로토콜에 직접 참여하는 경우에는 사용자 V로 위장하여 사용자 U와 세션키를 설정하는 공격이 가능하다. 1-pass unified model에 대한 KKI 공격자의 공격 과정은 [그림 3]과 같다.

Full unified model과 Full unified model with Key confirmation에서는 두 사용자 사이에 세션키를 설정하기 위해 매 세션마다 다르게 선택한 일회용 키 쌍을 이용하므로 공격자가 과거의 전송 정보와 세션키를 획득하더라도 현재의 세션키를 계산하기 위해서는 ECDH 문제를 해결해야 한다. 그러므로 KKP 공격자의 어려움은 ECDH 문제의 어려움과 동치이다. KKI 공격자도 이전 세션의 전송 정보를 이용하여 재 전송 공격을 하는 경우에, 사용

공격자 E		사용자 U
이전 세션의 $Q_{e,v}$와 세션키 획득	$Q_{e,v}$ →	
		$z_e = DH(d_{s,u}, Q_{e,v})$
		$z_s = DH(d_{s,u}, Q_{s,v})$
		$Z = z_e \mid\mid z_s$
KeyData = 이전의 세션키		KeyData = kdf(Z, [ShredInfo])

[그림 3] 1-pass unified model에 대한 KKI 공격 과정

자 U가 다른 일회용 키 쌍을 선택하여 전송하면 공격자는 이를 이용하여 세션키를 계산할 수 없다. 즉, 과거의 전송 정보와 세션키는 공격에 아무런 도움이 되지 못하므로 KKI 공격자의 어려움은 AI 공격자의 어려움과 동일하다.

Station-to-Station 프로토콜은 세션키를 설정하기 위해 사용자들이 매 세션마다 각각 다르게 선택한 일회용 키를 이용하므로 KKP 공격자의 어려움은 수동적 공격자의 어려움과 동일하다. KKI 공격의 경우에도, 공격자가 이전 세션의 전송 정보와 세션키를 모두 획득하더라도 사용자 U가 이전 세션과 다른 일회용 키를 이용하여 키 토큰을 생성하는 경우에는 공격자는 이에 대한 디지털 서명을 생성할 수도 없고 사전에 공유한 두 사용자사이의 비밀키를 알지 못하므로 사용자 V로 위장하는 것이 불가능하다. 그러므로 안전한 디지털 서명을 사용하는 경우에 Station -to-Station 프로토콜은 KKI 공격에 안전하다.

〔표 4〕는 ANSI X9.63 키 분배 프로토콜의 안전성 분석 결과를 정리한 것이다. 표에서 $\equiv^{p}_{m}$ECDH는 해당 공격자 환경에서 세션키를 구하는 문제가 ECDH 문제로 *polynomial time many-one* 귀착 가능하고 그 역도 성립함을 의미하고, $\equiv^{p}_{tt}$ECDH는 *polynomial time truth-table* 귀착 가능하고 그 역도 성립함을 의미한다[9][10].

V. 응용 분야

■ 전자 상거래

최근 활성화되고 있는 전자 상거래는 공개 채널인 인터넷을 통해 이루어지므로 사용자들이 전송하는 정보가 의도하지 않은 사람들에게도 공개될 수 있다는 단점이 있다. 이러한 문제를 해결하기 위해 사용자들이 웹 상점에 전송하는 정보를 암호화하여 전송하는데, 주로 서버와 사용자 사이에 공유된 비밀키를 이용하여 대칭키 암호 방식으로 암호화하여 전송하는 방식을 이용하고 있다. 또한, 대칭키 암호 방식을 이용하기 위해서는 사전에 서버와 사용자 사이에 공유된 비밀 정보를 설정해야하므로 키 분배 프로토콜이 반드시 필요하다.

그리고 전자 상거래는 기존의 상거래 방식과 달리 비 대면이라는 특징을 가지므로 정보를 전송하는 사용자의 신분에 대한 인증을 제공하는 방식이 적합하며, 일회용 키 쌍을 이용하여 세션키가 매 세션마다 다르게 설정되는 방식이 적합할 것이다.

이러한 요구사항을 만족하는 키 분배 방식으로는 해당 사용자만이 세션키를 계산할 수 있다는 묵시적 키 인증을 제공하고 매 세션마다 사용되는 키가 변경되도록 key freshness를 제공하는 X9.63의 1-pass Diffie-Hellman 프로토콜이 있다. 그러나 이 프로토콜은 사용자가 서버에 대한 묵시적 키 인

〔표 4〕 능동적 공격자에 대한 ANSI X9.63 키동의 프로토콜의 안전성 분석 결과

	Ephemeral Unified Model	1-pass Diffie-Hellman	Static unified model	Combined unified model with Key confirmation
AI	안전하지 않음	안전하지 않음	공격 불가능	안전함($\equiv^{p}_{m}$ ECDH)
KCI	안전하지 않음	안전하지 않음	안전하지 않음	안전하지 않음
FS	Full FS 제공	Half FS 제공	안전하지 않음	Full FS 제공
KKP	안전함($\equiv^{p}_{m}$ ECDH)	안전함($\equiv^{p}_{m}$ ECDH)	안전하지 않음	안전함($\equiv^{p}_{m}$ ECDH)
KKI	안전하지 않음	안전하지 않음	안전하지 않음	안전함($\equiv^{p}_{m}$ ECDH)

	1-pass unified model	Full unified model	Full unified model with Key confirmation	Station-to-Station
AI	안전함($\equiv^{p}_{tt}$ ECDH)	안전함($\equiv^{p}_{tt}$ ECDH)	안전함($\equiv^{p}_{tt}$ ECDH)	*
KCI	안전하지 않음	안전하지 않음	안전하지 않음	*
FS	Half FS 제공	Full FS 제공	Full FS 제공	Full FS 제공
KKP	안전함($\equiv^{p}_{tt}$ ECDH)	안전함($\equiv^{p}_{tt}$ ECDH)	안전함($\equiv^{p}_{tt}$ ECDH)	안전함($\equiv^{p}_{m}$ ECDH)
KKI	안전하지 않음	안전함($\equiv^{p}_{tt}$ ECDH)	안전함($\equiv^{p}_{tt}$ ECDH)	*

* 안전한 디지털 서명을 사용하면 공격 불가능

증을 제공받지 못하므로, 이러한 문제점을 해결하기 위해서는 full unified model이 적합하다. 또한, 1-pass unified model의 경우에는 통신 회수는 1이지만, 양방향 묵시적 키 인증과 key freshness를 제공하므로 위의 요구사항을 모두 만족하는 가장 적합한 방식이라 할 수 있다.

그리고 사용자의 신분에 대한 정확한 개체 인증을 제공하기 위해서는 디지털 서명을 이용하는 Station-to-Station 프로토콜을 이용할 수도 있다. 그러나 디지털 서명을 이용하는 키 분배 방식은 세션키 설정에 많은 계산이 요구된다는 단점이 있다.

■ **전자 금융 거래**

최근 들어, 전자 상거래와 함께 인터넷 뱅킹과 같이 네트워크를 통한 금융 거래가 널리 이용되고 있다. 전자 금융 거래는 전자 상거래와 비슷한 특징을 가지지만 전송되는 정보가 사용자의 계좌 번호나 금액 등과 관련되어 있으므로 보다 강력한 사용자 인증 기능을 제공해야 한다. 따라서, 묵시적 키 인증이나 개체 인증을 제공할 수 있는 키 분배 프로토콜이 적합하며 각 세션마다 세션키가 다르게 설정될 수 있도록 key freshness를 제공하는 방식이 적합하다. 이러한 요구사항을 만족하는 방식으로는 해당 사용자만이 세션키를 계산할 수 있다는 묵시적 키 인증을 제공하고 key freshness를 제공하는 1-pass Diffie-Hellman, full unified model, 1-pass unified model 등이 적합하다.

또한, 사용자에 대한 명확한 개체 인증을 제공하기 위해서는 디지털 서명을 이용하는 Station-to-Station 프로토콜이 적당하며, 사용자들이 계좌 번호나 금액 등과 같은 주요한 메시지를 서버로 전송하기 전에 서버와 동일한 키가 설정되었는지를 확인하는 키 확인 기능을 제공하기 위해서는 Full unified model with key confirmation 프로토콜을 사용하여야 한다.

■ **전자 우편 시스템**

전자 우편 시스템은 가장 많이 이용되고 있는 인터넷 서비스 중의 하나이다. 전자 우편 시스템에서는 사용자들이 상대방의 이름이나 전자 메일 주소와 같은 공개된 정보만을 가지고 자신이 메시지를 보내고자 하는 사용자에게 암호화된 메시지를 전송할 수 있도록 해야한다.

즉, 전자 우편을 보내기 전에 별도의 키 분배 프로토콜을 수행하지 않고 추가적인 통신 없이 세션키를 계산하고 이를 이용하여 암호화하는 방식이 적합하다. 또한, 메시지를 보낸 사람 외에는 다른 사람은 동일한 세션키를 계산할 수 없다는 것을 보장할 수 있도록 묵시적 키 인증을 제공하는 방식이 적합하므로, 상대방의 공개키 인증서와 자신의 비밀키를 이용하여 세션키를 계산하고 이를 이용하여 메시지를 암호화하는 방식이 바람직할 것이다.

Static unified model은 이러한 요구사항을 모두 만족하지만 메시지를 암호화하는데 항상 동일한 세션키가 이용된다는 단점이 있다. 따라서, 이러한 문제를 해결하기 위해 세션키를 생성하는데 필요한 통신 회수는 1이지만, 양방향 묵시적 키 인증과 key freshness를 제공하는 1-pass unified model을 사용하는 것이 바람직하다. 즉, 송신자가 암호화된 메시지와 일회용 키를 함께 전송하면 통신량의 추가 없이 암호화된 전자 우편을 전송할 수 있게 된다.

■ **무선 환경에서의 전자 상거래**

무선 환경에서의 전자 상거래는 유선 네트워크 환경에 비해 다양한 장소에서 서비스를 이용할 수 있다는 장점이 있다. 하지만 사용자의 이동이 빈번하게 발생하므로 사용자 인증 기능이 중요하고, 단말기의 계산 능력이 상대적으로 작기 때문에 이러한 특징을 고려하여 사용하는 키 분배 프로토콜을 선택해야 한다. 따라서, 타원 곡선에 기반한 키 분배 프로토콜은 같은 안전성을 보장하면서 상대적으로 사용하는 키 길이가 짧다는 장점이 있으므로 무선 환경에 적합한 키 분배 프로토콜이라 할 수 있다.

그리고 서버와 사용자 사이의 메시지의 교환을 최소화하기 위해 1-pass 프로토콜을 사용하는 것이 바람직할 것이다.

1-pass unified model의 경우에는 한번의 메시지 교환으로 양방향 묵시적 키 인증과 key freshness를 제공한다는 장점이 있으므로 위의 요구사항을 모두 만족한다.

■ **인증 시스템을 위한 비밀 공유키 설정**

패스워드를 이용한 기존의 인증 시스템은 항상 고정된 패스워드를 이용하므로 도청을 통한 재 전송 공격에 대해 안전하지 않다는 문제점이 있다. 최근에 이러한 문제점을 해결하기 위해 여러 가지 일회용 패스워드 시스템이 제안되었으며, 그중 대표적인 방식이 challenge-response 형태의 인증 방식이다.

이 방식은 사전에 사용자와 서버 사이에 비밀키를 공유하고 사용자가 서버에 접속하는 경우에 서버는 랜덤 수를 선택하여 사용자에게 전송한다. 그리고 나서, 랜덤 수를 수신한 사용자는 사전에 공유한 키를 이용하여 이를 암호화하여 서버에게 전송하고, 서버는 이를 복호하여 자신이 처음에 보낸 랜덤 수와 같은지를 확인함으로써 사용자에 대한 인증을 수행하는 것이다.

따라서, 사용자가 처음에 서버에 등록하는 과정에서 사용자와 서버 사이에 비밀키를 공유하기 위한 키 분배 프로토콜이 필요하다. 인증 시스템을 위한 비밀키를 설정하기 위해서는, 서버에게 사용자에 대한 인증을 제공해야 하므로 묵시적 키 인증을 제공하는 키 분배 프로토콜을 사용하는 것이 적합하다. 또한, 사용자의 long-term 키가 노출되더라도 사용자가 등록한 인증 시스템의 비밀키의 안전성에는 영향을 미치지 않도록 하기 위해 key freshness를 제공하는 방식이 적합하다. 이러한 요구사항을 만족하는 방식에는 1-pass Diffie-Hellman, full unified model, 1-pass unified model 등이 있다.

또한, 인증 시스템에 등록하기 전에 사용자의 신분에 대한 정확한 개체 인증이 필요한 경우에는 디지털 서명 을 이용하는 Station-to-Station 프로토콜이 적합하고 사용자들과 서버사이에 사용할 비밀 공유키가 제대로 설정되었는지를 확인하기 위해서는, 키 확인 기능을 제공하는 Full unified model with key confirmation 프로토콜이 적합하다.

앞장의 〔표 5〕는 지금까지 설명한 키 분배 프로토콜의 각 응용 환경에서 만족해야 할 요구 사항과 적합한 키 분배 프로토콜을 간략히 정리한 것이다.

〔표 5〕 응용 환경의 요구사항 및 적합한 키 분배 프로토콜

응용 환경	요구 사항	적합한 프로토콜
전자 상거래	• 해당 사용자만이 세션키를 계산할 수 있어야 함 ➡ 묵시적 키 인증 제공 • 세션마다 다른 키를 사용하는 것이 바람직 ➡ key freshness 제공	Full unified model, 1-pass unified model
	• 사용자의 신분에 대한 확실한 인증을 제공(선택 사항) ➡ 디지털 서명 포함	Station-to-Station
전자 금융 거래	• 해당 사용자만이 세션키를 계산할 수 있어야 함 ➡ 묵시적 키 인증 제공 • 세션마다 다른 키를 사용하는 것이 바람직 ➡ key freshness 제공	1-pass Diffie-Hellman Full unified model 1-pass unified model
	• 사용자의 신분에 대한 확실한 인증을 제공(선택 사항) ➡ 디지털 서명 포함	Station-to-Station
	• 메시지를 전송하기 전에 동일한 키가 설정되었음을 확인(선택 사항) ➡ 키 확인 제공	Full unified model with key confirmation
전자 우편	• 추가적인 통신량 없이 세션키를 설정할 수 있어야 함 ➡ 고정된 키 쌍 이용 • 송신자 외에 다른 사람은 동일한 세션키를 계산할 수 없음을 보장해야 함 ➡ 묵시적 키 인증 제공	Static unified model
	• 세션마다 다른 키를 사용하는 것이 바람직 ➡ key freshness 제공	1-pass unified model
무선 환경에서의 전자 상거래	• 무선 단말기의 계산 능력을 고려하여 서버와 사용자 사이의 메시지 교환 및 사용자의 계산량을 최소화해야 함 • 서버는 사용자만이 세션키를 계산할 수 있음을 보장받아야 함 ➡ 묵시적 키 인증 제공	1-pass Diffie-Hellman 1-pass unified model
인증 시스템의 비밀 공유키 설정	• 인증 시스템에 등록한 해당 사용자만이 세션키를 계산할 수 있어야 함 ➡ 묵시적 키 인증 제공 • 사용자의 long-term 키가 노출되더라도 인증 서버와의 공유키는 안전해야 함 ➡ Forward secrecy 제공	1-pass Diffie-Hellman Full unified model 1-pass unified model
	• 사용자의 신분에 대한 확실한 인증 제공 ➡ 디지털 서명 포함	Station-to-Station

Ⅵ. 결 론

키 분배 프로토콜은 안전한 암호 시스템의 구현에 있어 가장 필수적인 요소이다. 따라서 키 분배 프로토콜에 대한 많은 연구가 진행되었으며, 최근 들어 키 분배 프로토콜의 표준화 작업이 활발히 진행되고 있다.

그리고, 타원 곡선 암호 시스템은 유한체 상의 암호 시스템에 비해 사용하는 키의 길이가 짧고 효율적이라는 장점이 있어, 타원 곡선에 기반한 키 분배 프로토콜 또한 응용 분야가 다양하다.

본 논문에서는 ANSI X9.63에서 제안한 8개의 타원 곡선에 기반한 키 동의 프로토콜의 세션키 설정 과정 및 특징을 분석하고, 몇몇 능동적 공격자 모델에 대해 각 프로토콜의 안전성을 분석하였다. 본 논문에서 고려한 능동적 공격자 모델은 active impersonation 공격, forward secrecy에 대한 공격, key compromise impersonation 공격, known key security에 대한 공격 등이다.

또한, 키 분배 프로토콜의 주요 응용 분야의 요구사항을 분석하고 각 환경에 적합한 키 분배 방식을 제안하였다.

참 고 문 헌

[1] ANSI X9.42, "Agreement of symmetric Key on Using Diffie-Hellman Cryptography", 2001.

[2] ANSI X9.63, "Public Key Cryptography for the financial services industry : key agreement and key transport using elliptic curve cryptography", 2001.

[3] S. Blake-Wilson, D. Johnson, A. Menezes, "Key agreement protocols and their security analysis", Cryptography and Coding, Lecture Notes in Computer Science 1355, pp. 30~45, 1997.

[4] W. Diffie, M.E. Hellman, "New directions in cryptography", IEEE Transaction of Information Theory, IT-22, 6, pp. 644~654, 1976.

[5] W. Diffie, P.C. Oorschot, M.J. Wiener, "Authentication and Authenticated Key Exchange", Designs, Codes and Cryptography, pp. 107-125, 1992.

[6] IEEE P1363, "Standard for Public-Key Cryptography", Working draft D13, 1999.

[7] D. Johnson, "Diffie-Hellman Key Agreement Small Subgroup Attack", a Contribution to X9F1 by Certicom, July, 1996.

[8] N. Kobliz, "Elliptic curve cryptosystems", Mathematics of Computation, 48, pp. 203~209, 1987.

[9] S. J. Kim, M. Mambo et al, "On the security of the Okamoto-Tanaka ID-Based Key Exchange scheme against Active attacks", IEICE Trans, pp. 231~238, Jan. 2001.

[10] M. Mambo and H. Shizuya, "A note on the complexity of breaking exchange scheme", IEICE Trans. Okamoto-Tanaka ID-based key Fundamentals, Vol. E82-A, No. 1, pp. 77~80, Jan, 1999.

[11] R.A. Rueppel, P.C. van Oorschot, "Modern Key Agreement Techniques", Computer Communications, pp. 458-465, 1994.

[12] K. Sakurai and H. Shizuya, "Relationships among the computational powers of breaking discrete log cryptosystems", Proc. Eurocrypto '95 LNCS 921, pp. 341~355, Springer -Verlag,1995.

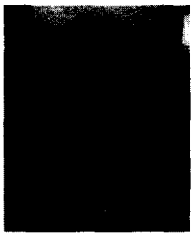
[13] SECG SEC1 : Elliptic Curve Cryptography, V1.0, 2000. 9

[14] 이동훈, 황효선, 임채훈, "타원 곡선 암호의 기초와 응용", Technical Report, FS-TR01-03

〈著者紹介〉

오 수 현 (Soo-hyun Oh) 학생회원
1998년 2월 : 성균관대학교 정보공학과 졸업(공학사)
2000년 2월 : 성균관대학교 대학원 전기전자 및 컴퓨터 공학부 졸업(공학석사)
2000년 3월~현재 : 성균관대학교 정보통신공학부 박사 과정

이 승 우 (Seung-Woo Lee) 학생회원
2001년 2월 : 강남대학교 전자계산학과 졸업(공학사)
2001년 3월~현재 : 성균관대학교 정보통신공학부 석사 과정



심 경 아 (Kyung-Ah Shim)
1992년 2월 : 이화여자대학교 수학과 졸업(이학사)
1994년 2월 : 이화여자대학교 대학원 수학과 졸업(이학석사)
1999년 2월 : 이화여자대학교 대학원 수학과 졸업(이학박사)
2000년 2월~현재 : 한국정보보호진흥원(KISA) 암호 기술팀 선임연구원

양 형 규 (Hyung-Kyu Yang) 정회원
1983년 2월 : 성균관대학교 전자공학과 졸업(공학사)
1985년 2월 : 성균관대학교 대학원 전자공학과 졸업(공학석사)
1983년 12월~1990년 2월 : 삼성전자 선임연구원
1994년 2월 : 성균관대학교 대학원 정보공학과 졸업(공학박사)
1995년 3월~현재 : 강남대학교 컴퓨터공학과 교수

원 동 호 (Dong-Ho Won) 정회원
성균관대학교 전자공학과 졸업 (학사, 석사, 박사)
1978년 ~ 1980년 : 한국전자통신연구소 전임연구원
1992년 ~ 1994년 : 성균관대학교 교학처장
1996년 ~ 1998년 : 국무총리실 정보화추진위원회 자문위원
1999년 ~ 2001년 : 성균관대학교 전기전자 및 컴퓨터공학부장 정보통신대학원장
현재 : 성균관대학교 정보통신공학부 교수
한국정보보호학회 회장
정통부 지정 정보보호인증기술연구센터 센터장
〈관심분야〉 암호이론,