情 報 保 護 學 會 誌
第 13 券 第 5 號, 2003. 10

ISO/IEC 키 관리 표준 규격 분석

조 은 성*, 정 영 석*, 오 수 현*, 양 형 규**, 원 동 호***

요 약

정보통신 기술의 발전으로 네트워크를 통한 정보의 공유가 보편화되어 다양한 형태의 서비스가 제공되면서 정보 보호와 정보를 이용하는 사용자 인증이 중요한 이슈가 되고있으며 이를 위한 암호기술의 사용이 급증하고 있다. 이에 현대의 암호기술의 안전성과 관련된 가장 핵심적인 요소인 키를 안전하게 관리하고자 하는 키 관리 기술이 중요하게 연구되고 있으며 이를 위한 다양한 서비스들이 제공되고 있다. 키 관리 기술은 키의 생성에서부터 키를 폐기하기까지 여러 가지 단계로 정의되는데, 본 고에서는 대칭키와 공개키를 기반으로 키의 생성과 키 분배에 대하여 기술하고 있는 ISO/IEC 11770과 타원곡선 기반의 키 분배 프로토콜에 대하여 기술하고 있는 ISO/IEC 15946-3에 대하여 분석하고자 한다.

Ⅰ. 서 론

정보통신 기술의 발전은 컴퓨터의 보급이 보편화되고 네트워크를 통하여 정보를 주고받을 수 있는 기반 시설이 확충됨으로써 더욱 급속한 성장세를 보이고 있다. 정보 시스템을 이용하여 다양한 정보의 공유와 서비스가 제공되면서 전송되는 정보에 대한 보호의 필요성이 증가하게 되었다. 또한 인터넷과 같은 네트워크가 일반적인 사용자들이 보편적으로 사용할 수 있는 환경이 갖추어지면서 사용자의 요구에 맞는 컨텐츠를 유료로 제공하는 등의 서비스가 등장하게 되었으며, 사용자의 상태나 올바른 권한을 가진 사용자임을 확인할 수 있는 기술이 요구되고 있다. 특히, 정보화 사회가 발전함에 따라 개인의 프라이버시, 기업의 경영 비밀 등 제 3자로부터 보호해야 할 정보의 급증으로 정보보호를 위해 암호기술의 사용이 요구되고 있다. 이에 정보 시스템 내에서의 정보보호 및 사용자 합법성 확인을 위한 방법으로 암호기술이 크게 주목받고 있다. 이러한 현대 암호기술의 핵심적인 요소 중에 하나가 키(key) 관리이다. 보편적으로 현대의 암호기술이 키에 의존하여 정보를 보호하기 때문에 사용자는 자신의 비밀키가 외부에 노출되지 않도록 보관하여야 한다. 최근에는 키의 분실이 발생하였거나 합법적인 이유로 키를 복구하고자 하는 기술이 요구되면서 이러한 서비스를 제공하는 등의 키 관리 기술이 표준으로 제정되었다.

최근 키 관리 서비스의 제공자, 키 관리 기술 개발자 및 사용자의 수가 증가함에 따라 다양한 키 관리 요소에 대한 일괄적인 관리가 필요하게 되었으며, 각 요소들이 기반하고 있는 서로 다른 표준들 간에 호환성을 제공하고 사용자의 편리성을 고려한 통합 키 관리 시스템이 요구되고 있다.

따라서, 본 고에서는 키 관리 기술의 표준 규격으로써 ISO/IEC 11770, 15946-3에 정의된 키 관리 표준 규격을 키 관리 요소별로 나누어 분석하였다. 2장에서는 키 관리 기술에 대한 개요와 각 서비스의 정의에 대하여 간략하게 논하였으며, 3장에서는 키 관리 요소별로 ISO/IEC에 기술된 키 생성과 키 분배에 대하여 표준을 분석하고 4장에서는 결론을 맺는다.

Ⅱ. 키 관리 기술

1. 키 관리의 개요

키 관리는 키 및 키 재료(key material)의 생성, 등록, 인증, 말소, 분배, 설치, 저장, 보관, 취소, 파

* 성균관대학교 정보통신공학부 정보통신보호연구실({escho, yschung, shoh}@dosan.skku.ac.kr)
** 강남대학교 컴퓨터·미디어 공학부 교수(hkyang@kangnam.ac.kr)
*** 성균관대학교 정보통신공학부 교수(dhwon@dosan.skku.ac.kr)

생, 파기 등과 같은 서비스의 관리를 총칭한다. 키 관리의 목적은 키를 이용하는 모든 과정에서 안전성을 보장하기 위한 모든 절차를 규정하는 것이며 키 관리 절차는 사용된 알고리즘, 키의 의도적인 사용, 그리고 사용에 대한 보안정책에 의해 좌우된다. 또한 암호화 장비에서 발생하는 사항도 포함한다.

2. 키 관리 서비스

키 관리는 생성, 등록, 인증, 분배, 설치, 저장, 파생, 보관, 취소, 말소, 폐기 등의 기본적인 키 관리 서비스를 포함한다. 이러한 서비스는 키 관리 시스템에 의해 제공되거나 신뢰성 있는 제 3의 신뢰기관에 의해 제공될 수 있다. 이 때 신뢰기관은 모든 객체가 신뢰할 수 있도록 보안 요구사항을 만족시키는 범위 내에서 서비스를 제공해야 한다[6,8,9]. ISO/IEC 11770-1에서 기술된 키 관리 요소에 대한 정의는 다음과 같다.

2.1 키 생성

키 생성은 암호학적으로 안전한 키를 생성하는 절차를 의미하며, 이 때 예측이 불가능하고 위조할 수 없는 랜덤 수(random number)를 사용해야 하며 재사용해서는 안 된다. 이는 하나의 키의 노출로 인해 그 키와 관련된 정보뿐만 아니라 노출된 키로부터 파생되었거나 관련되어 있는 다른 모든 키들에 대한 접근 권한도 임의의 사용자에게 제공될 수 있기 때문이다[3,6,8].

2.2 키 등록

키 등록은 생성된 키를 정당한 사용자와 관련시키는 것으로 등록 기관(RA: Registration Authority)에 의해 이루어지며, 등록 기관은 키와 관련된 정보의 기록을 안전하게 유지해야 한다. 또한, 키 등록기관은 키 등록뿐만 아니라 키를 말소시키는 역할도 수행한다.

2.3 키 인증서 생성

키 확인서는 보통 공개키와 객체의 연관성을 보장하는 인증서(certificate)라 하며 인증기관에 의해 생성된다. 인증 기관(CA: Certification Authority)은 사용자로부터 키 인증에 대한 요구를 받은 경우 키 인증서를 생성한다.

2.4 키 분배

키 분배는 인가된 객체들 사이에 키 또는 키 재료가 안전하게 공유되는 것을 의미한다. 비대칭 암호 방식에서는 키를 분배하는 특정 메커니즘을 사용하고, 대칭 암호 방식에서는 키 전송 센터(KTC : Key transport center), 키 분배 센터(KDC : Key distribution center)에 의해 분배가 이루어진다[1,2,10].

2.5 키 설치

키 설치는 키를 사용하기 전에 필요한 절차로 키 관리 시스템 내에서 안전하게 제공되어야 한다.

2.6 키 저장

키 저장은 키를 사용하거나 복구를 위한 백업을 위해 키를 안전하게 저장하는 것을 의미한다. 이 때, 키는 물리적으로 안전한 장치에 저장되는 것이 바람직하며, 저장된 키 재료에 대하여 기밀성 및 무결성을 제공하여야 한다. 키 저장은 키의 생명주기(Key life-cycle)동안의 모든 상태(활성 준비, 활성, 활성 종료 등)에서 발생할 수 있다[6].

2.7 키 유도

키 유도는 원본 키(original key)로부터 파생키(derivation key)를 유도하는 것으로 유도된 키가 원본 키를 노출시키지 않도록 하기 위해 유도 연산은 역변환이 불가능하고 예측 불가능해야 한다.

2.8 키 보관

키 보관은 키의 일반적인 사용이 중단된 이후 그 키의 오용 등의 문제가 발생했을 경우 그러한 사실을 증명하는데 키가 사용될 수 있도록 하기 위해 이루어진다.

2.9 키 복구

키 복구는 합법적 상황에서 암호문을 복호화 하거나, 사용자가 자신의 비밀키를 분실했을 경우 등의 유사시에 허가된 사용자만이 복호화를 할 수 있는 기능을 제공하기 위해 수행된다[4].

2.10 키 취소

키 취소는 키의 오용이 의심되거나 알려진 경우 키 취소는 키의 안전한 비활성 상태를 유지하기 위해 이루어진다. 키 취소는 키 삭제라고도 하며, 유효기간이 만료된 키나 소유자의 환경이 변경된 경우 발생한다.

키가 취소된 후에는 단지 키와 관련된 정보의 복호화나 검증만을 위해 사용된다.

2.11 키 말소

키 말소는 키와 객체의 관계를 제거하는 것으로, 키 등록 기관에 의해 폐기 과정의 일부분으로 제공된다.

2.12 키 폐기

키 폐기는 더 이상 사용될 필요가 없는 키의 안전한 폐기를 위해 이루어진다. 키를 폐기한다는 것은 키와 관련된 모든 기록을 제거함으로써 폐기 후에 남아있는 어떠한 정보를 가지고도 폐기된 키를 다시 복구시킬 수 없도록 하는 것을 의미한다. 또한, 이는 사용되는 키뿐만 아니라 보관된 모든 복사본에 대한 폐기도 포함한다. 보관된 키를 폐기할 때는 보관된 키에 의해 보호된 자료가 더 이상 필요 없는지의 여부를 사전에 판단 하여야한다[6].

Ⅲ. 키 관리 요소별 표준 분석

1. 키 생성

ISO/IEC 11770에서 키 생성 서비스는 키 생성, 키 등록, 키 인증서 생성, 키 분배, 키 저장의 서비스를 포함한다. 키 생성 서비스의 요소는 다음과 같고 각 서비스에 대한 정의는 2.2절에서 정의되어 있다.

- 키 생성
- 키 등록
- 키 인증서 생성
- 키 분배
- 키 저장

2. 키 분배

2.1 ISO/IEC 15946-3

ISO/IEC 15946-3은 타원 곡선 기반 키 분배 프로토콜에 대하여 기술하고 있다. 이 표준에서는 키 교환 방법을 키 동의(key agreement)와 키 전송(key transport)으로 나누어서 기술한다.

■ 시스템 파라미터
- d_x : 비밀키
- P_X : 공개키 ($P_X = d_X G$)
- SK_X : 비밀 서명키
- VK_X : 공개 검증키
- h, I :
  - 인수의 곱셈이 사용되고, 인수의 곱셈이 사용되지 않은 키와 호환될 필요가 없는 경우
    → $h = \#E/n$, $I = 1$
  - 인수의 곱셈이 사용되고, 인수의 곱셈이 사용되지 않은 키와 호환되어야 하는 경우
    → $h = \#E/n$, $I = h^{-1}$
  - 인수의 곱셈이 사용되지 않은 경우
    → $h = 1$, $I = 1$

본 고의 각 프로토콜 분석에서 사용된 용어의 정의는 다음과 같다.

- **통신 회수**: 사용자와 상대방이 공통의 비밀 세션키를 공유하기 위하여 필요한 통신의 수
- **개체 인증**: 키 분배 프로토콜에 참여하고있는 상대방의 신원을 확인하는 것으로 명시적 키 인증에 의해 제공될 수 있음
- **키 확인**: 키 분배 프로토콜에 참여한 합법적인 사용자가 자신이 의도한 상대방과 실제로 비밀 세션키를 공유하였음을 확인
- **묵시적 키 인증**: 키의 소유 여부는 알려져 있지 않다고 하더라도 키 분배 프로토콜에 참여한 상대방만이 세션키를 계산할 수 있음을 보장
- **Key freshness**: 세션마다 설정된 키가 바뀜

2.1.1 키 동의 프로토콜

키 동의 프로토콜은 두 사용자 사이에서 어느 하나가 세션키를 결정하지 않고 두 사용자가 모두 참여하여 합의하에 세션키를 공유하는 과정이다.

① 상호 대화가 없는 Diffie-Hellman 형태의 키 동의 프로토콜(KANIDH)

1) 사용자 A, B는 자신의 비밀키를 이용하여 공개키 P_A, P_B를 생성, 공개 디렉토리에 등록시킨다.
2) 통신을 원하는 사용자 A, B는 각각 상대방의 공개키와 자신의 비밀키 d_A, d_B를 이용하여 세션키 K_{AB}를 생성한다.

$$K_{AB} = (d_A \cdot I)(hP_B) = (d_B \cdot I)(hP_A)$$

상호 대화가 없는 Diffie-Hellman 형태의 키 동의 프로토콜에 대한 수행 과정은 [그림 1]과 같다.

사용자 A	공개정보 P_A, P_B	사용자 B
$P_A = d_A G$ P_B 확인 $K_{AB} = (d_A \cdot 1)(hP_B)$		$P_B = d_B G$ P_A 확인 $K_{AB} = (d_B \cdot 1)(hP_A)$

[그림 1] KANIDH 프로토콜

KANIDH 프로토콜은 항상 동일한 세션키가 생성되기 때문에 다음과 같은 권고사항이 있으며 프로토콜 분석결과는 [표 1]과 같다.

※ 항상 동일한 세션키 생성에 대한 권고사항
→ 세션키와 time-varying information(ex. time-stamp)을 유도함수에 입력하여 새로운 세션키 생성 권고

[표 1] KANIDH 프로토콜 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	key freshness
KANIDH	0	-	-	양방향	-

② ElGamal 형태의 키 동의 프로토콜(KAEG)

1) 사용자 A는 자신의 공개키/비밀키 쌍을 생성하여 자신의 공개키를 공개 디렉토리에 등록시키고, 랜덤수 $r \in_{R/S} \{1, \cdots, n-1\}$을 선택하여 생성한 중간값 KT_{A1}을 사용자 B에게 전송해 준다.

$$KT_{A1} = rG$$

2) 사용자 A는 공개정보 P_B를 사용하여 아래와 같이 세션키 K_{AB}를 생성하고, 사용자 B는 자신의 비밀키 d_B를 사용하여 다음과 같이 세션키 K_{AB}를 생성한다.

$$K_{AB} = (r \cdot 1)(hP_B) = (d_B \cdot 1)(hKT_{A1})$$

사용자 A	공개정보 P_B	사용자 B
$r \in_{R/S} \{1, \cdots, n-1\}$ $KT_{A1} = rG$ $K_{AB} = (r \cdot 1)(hP_B)$	KT_{A1} →	$P_B = d_B G$ KT_{A1}확인 $K_{AB} = (d_B \cdot 1)(hKT_{A1})$

[그림 2] KAEG 프로토콜

[표 2] KAEG 프로토콜 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	key freshness
KAEG	1	-	-	일방향	일방향

ElGamal 형태의 키 동의 프로토콜에 대한 수행과정은 [그림 2]와 같으며 분석결과는 [표 2]에서 정리하였다.

③ Diffie-Hellman 형태의 키 동의 프로토콜(KADH)

1) 사용자 A는 랜덤수 r_A를 선택하여 아래와 같이 중간값 KT_{A1}생성한 후 사용자 B에게 전송해 준다. 사용자 B는 랜덤수 r_B를 선택하여 아래와 같이 중간값 KT_{B1}을 생성한 후 사용자 A에게 전송한다.

$$KT_{A1} = r_A G$$
$$KT_{B1} = r_B G$$

2) 사용자 B는 수신한 중간값 KT_{A1}과 자신이 선택한 랜덤수 r_B를 이용하여 아래와 같은 방식으로 세션키 K_{AB}를 생성한다.

$$K_{AB} = (r_B \cdot 1)(hKT_{A1})$$

3) 사용자 A는 수신한 중간값 KT_{B1}과 자신이 선택한 랜덤수 r_A를 이용하여 아래와 같이 세션키 K_{AB}를 생성한다

$$K_{AB} = (r_A \cdot 1)(hKT_{B1})$$

KADH 프로토콜에 대한 수행과정은 [그림 3]과 같다.

사용자 A		사용자 B
$r_{A \in_{R/S}} \{1, \cdots, n-1\}$ $KT_{A1} = r_A G$ KT_{B1} 확인 $K_{AB} = (r_A \cdot 1)(hKT_{B1})$	KT_{A1} → KT_{B1} ←	$r_{B \in_{R/S}} \{1, \cdots, n-1\}$ $KT_{B1} = r_B G$ KT_{A1} 확인 $K_{AB} = (r_B \cdot 1)(hKT_{A1})$

[그림 3] KADH 프로토콜

KADH 프로토콜은 2회의 통신회수를 가지며 key freshness를 제공한다. 분석결과는 [표 3]과 같다.

〔표 3〕 KADH 프로토콜 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	key freshness
KADH	2	-	-	-	양방향

④ 2개의 키 쌍을 갖는 Diffie-Hellman 형태의 키 동의 프로토콜(KADH2KP)

1) 사용자 A는 공개키, 비밀키 쌍을 생성하고 랜덤수 $r_{A \in R/S}\{1,\cdots,n-1\}$를 선택하여 아래와 같이 중간값 KT_{A1}을 생성하여 사용자 B에게 전송한다.

$$P_A = d_A G$$
$$KT_{A1} = r_A G$$

2) 사용자 B는 수신한 KT_{A1}을 검증 후 랜덤수 $r_{B \in R/S}\{1,\cdots,n-1\}$를 선택하여 아래와 같이 중간값 KT_{B1}을 생성하고 사용자 A에게 전송한다.

$$KT_{A1} \text{ 확인}$$
$$KT_{B1} = r_B G$$

3) 사용자 A는 전송 받은 중간값 KT_{B1}을 검증 후 아래와 같은 방법으로 세션키 K_{AB}를 생성한다.

$$K_{AB} = (d_A \cdot 1)(hKT_{B1}) \| (r_A \cdot 1)(hP_B)$$

4) 사용자 B는 중간값 KT_{A1}을 이용하여 아래와 같은 방법으로 세션키 K_{AB}를 생성한다.

$$K_{AB} = (r_B \cdot 1)(hP_A) \| (d_B \cdot 1)(hKT_{A1})$$

2개의 키 쌍을 갖는 Diffie-Hellman 형태의 키 동의 프로토콜에 대한 수행과정은 〔그림 4〕와 같으며, 이에 대한 분석결과는 〔표 4〕와 같다.

사용자 A	공개정보 P_A, P_B	사용자 B
$P_A = d_A G$ $r_{A \in R/S}\{1,\cdots,n-1\}$ $KT_{A1} = r_A G$	KT_{A1} →	$P_B = d_B G$
		KT_{A1} 확인 $r_{B \in R/S}\{1,\cdots,n-1\}$ $KT_{B1} = r_B G$ $K_{AB} = (r_B \cdot 1)(hP_A)$ $\|(d_B \cdot 1)(hKT_{A1})$
	← KT_{B1}	
KT_{B1} 확인 $K_{AB} = (d_A \cdot 1)(hKT_{B1})$ $\|(r_A \cdot 1)(hP_B)$		

〔그림 4〕 KADH2KP 프로토콜

〔표 4〕 KADH2KP 프로토콜 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	key freshness
KADH2KP	2	-	-	양방향	양방향

⑤ 2개의 서명과 키 확인이 있는 Diffie-Hellman 형태의 키 동의 프로토콜(KADH2SKC)

1) 사용자 A, B는 자신의 서명키 쌍을 생성한다. 사용자 A는 랜덤수 $r_{A \in R/S}\{1,\cdots,n-1\}$를 선택하고 중간값 KT_{A1}을 생성하여 사용자 B에게 전송한다.

$$KT_{A1} = r_A G$$

2) 사용자 B는 수신한 중간값을 검증 후 랜덤수 r_B를 생성하여 아래와 같은 방법으로 세션키 K_{AB}를 생성한다. 생성한 랜덤수 r_B와 수신한 중간값 KT_{A1}를 연접하여 생성한 값 DB_1에 자신의 서명키로 서명하여 사용자 A에게 전송한다.

$$K_{AB} = (r_B \cdot 1)(hKT_{A1})$$
$$DB_1 = r_B G \| KT_{A1} \| A \| [Text_1]$$
$$KT_{B1} = S_{SK_B}(DB_1) \| f_{K_{AB}}(DB_1)$$

3) 사용자 A는 수신한 값 $KT_{B1}, (DB_1)$을 검증 후 세션키 K_{AB}을 생성한다. 생성한 K_{AB}와 랜덤수 〔r_A, r_B〕, 사용자 B의 ID를 사용하여 DB_2를 생성하고, 생성된 DB_2에 자신의 서명키로 서명하여 사용자 B에게 전송해 준다.

$$V_{VK_B}(KT_{B1}) \text{ 확인}$$
$$K_{AB} = (r_A \cdot 1)(hr_B G)$$
$$f_{K_{AB}}(DB_1) \text{ 확인}$$
$$DB_2 = r_A G \| r_B G \| B \| [Text_2]$$
$$KT_{A2} = S_{SK_A}(DB_2) \| f_{K_{AB}}(DB_2)$$

4) 사용자 B는 서명의 검증을 수행한다.

$$V_{VK_A}(KT_{A2})$$
$$B, r_A G, r_B G \text{ 확인}$$
$$f_{K_{AB}}(DB_2) \text{ 확인}$$

KADH2SKC 프로토콜에 대한 수행 과정은 〔그림 5〕와 같다.

사용자 A	공개정보 VK_A, VK_B	사용자 B
SK_A, VK_A $r_A \in_{R/S} \{1,\cdots,n-1\}$ $KT_{A1} = r_A G$	KT_{A1} →	SK_B, VK_B KT_{A1} 확인 $r_B \in_{R/S} \{1,\cdots,n-1\}$ $K_{AB} = (r_B \cdot 1)(hKT_{A1})$ $DB_1 = r_B G \| KT_{A1} \| A [\| Text_1]$
$V_{VK_B}(KT_{B1})$ $A, r_A G, r_B G$ 확인 $K_{AB} = (r_A \cdot 1)(hr_B G)$ $f_{K_{AB}}(DB_1)$ 확인 $DB_2 = r_A G \| r_B G \| B \| [Text_2]$ $KT_{A2} =$ $S_{SK_A}(DB_2) \| f_{K_{AB}}(DB_2)$	KT_{B1}, (DB_1) ←	$KT_{B1} = S_{SK_B}(DB_1) \| f_{K_{AB}}(DB_1)$
	KT_{A2}, (DB_2) →	$V_{VK_A}(KT_{A2})$ $B, r_A G, r_B G$ 확인 $f_{K_{AB}}(DB_2)$ 확인

〔그림 5〕 KADH2SKC 프로토콜

KADH2SKC에 대한 분석결과는 〔표 5〕와 같다.

〔표 5〕 KADH2SKC 프로토콜 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	key freshness
KADH2SKC	3	양방향	양방향	양방향	양방향

2.1.2 키 전송 프로토콜

키 전송 프로토콜은 사용자 또는 신뢰받는 기관이 "세션키"를 다른 사용자에게 전송하는 과정이다.

■ 시스템 파라미터
- K : 세션키
- d_B : 비밀키
- P_B : 공개키 ($P_B = d_B G$)
- SK_A : 비밀 서명키
- VK_A : 공개 검증키

TVP : 타임스탬프(time stamp or sequence number)

$\pi(Q)$: 변환 π에의 점 Q로부터 얻어진 정수 값

$\pi^*(Q)$: $(\pi(Q) \bmod 2^{\lceil \rho/2 \rceil}) + 2^{\lceil \rho/2 \rceil}$, $(\rho = \lceil \log_2 n \rceil)$

① ElGamal 형태의 키 전송 프로토콜(KTEG)

1) 사용자 A는 랜덤수 r과 세션키 K를 생성한 후 키 유도 함수(kdf : key derivation function)를 사용하여 중간키 K′를 계산한다. 사용자 A, B의 세션키 KT_{A1}으로 K′를 암호화하여 전송한다.

$BE = (A \| K)$ XOR $kdf(\pi((r \cdot 1)(hP_B)),$ parameters)
$K' = kdf(\pi((r \cdot 1)(hP_B)), MAC$ parameters)
$KT_{A1} = BE \| rG \| MAC(K',\ BE)$

2) 사용자 B는 세션키 KT_{A1}을 사용하여 수신된 메시지를 복호화 한 후 키 유도 함수를 이용하여 중간값 K″를 계산한다. 검증 과정을 거쳐 세션키 K임을 확인한다.

$A \| K = BE$ XOR $kdf(\pi((d_B \cdot 1)(h \cdot rG)),$ parameters)
$K'' = kdf(\pi((d_B \cdot 1)(h \cdot rG)),$ MAC parameters)
$MAC(K',\ BE) \doteq MAC(K'',\ BE)$, A 확인

KTEG 프로토콜에 대한 과정은 〔그림 6〕과 같고, 1회의 통신 회수를 가지며 묵시적 키 인증과 key freshness를 제공한다. 〔표 6〕에서 결과를 정리하였다.

사용자 A	공개정보 P_B	사용자 B
P_B 확인 $r \in_{R/S} \{1,\cdots,n-1\}$ $BE = (A \| K)$ XOR $kdf(\pi((r \cdot 1)(hP_B)),$ parameters) $K' = kdf(\pi((r \cdot 1)(hP_B)),$ MACparameters) (중간키: intermediate key) $KT_{A1} = BE \| rG \| MAC(K', BE)$	KT_{A1} →	rG 확인 $A \| K = BE$ XOR $kdf(\pi((d_B \cdot 1)(h \cdot rG)),$ parameters) $K'' = kdf(\pi((d_B \cdot 1)(h \cdot rG)),$ MACparameters) (중간키: intermediate key) $MAC(K', BE) \doteq MAC(K'', BE)$, A 확인

〔그림 6〕 KTEG 프로토콜

〔표 6〕 KTEG 프로토콜 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	key freshness
KTEG	1	-	-	일방향	일방향

〔표 7〕 KTEGOS 프로토콜 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	key freshness
KTEGOS	1	일방향	-	양방향	일방향

② 서명이 있는 ElGamal 형태의 키 전송 프로토콜 (KTEGOS)

1) 사용자 A는 랜덤수 r을 선택하고 아래와 같은 방법으로 BE와 중간값 KT_{A1}을 생성한다. 생성된 KT_{A1}에 자신의 서명키로 서명하여 사용자 B에게 전송한다.

$r \in_{R/S} \{1, \cdots, n-1\}$
$BE = (A \| K) \ XOR \ kdf(\pi((r \cdot 1)(hP_B)), parameters)$
$KT_{A1} = B \| TVP \| rG \| BE$
$S_{SK_A}(KT_{A1})$

2) 사용자 B는 전송 받은 서명을 검증하고 중간 값을 확인한 후 연산을 통해 세션키 K를 생성한다.

$V_{VK_A}(KT_{A1})$
B, TVP, rG 확인
$A \| K = BE \ XOR \ kdf(\pi((d_B \cdot 1)(h \cdot rG)), parameters)$
A 확인

KTEGOS 프로토콜에 대한 자세한 과정은 〔그림 7〕과 같다.

KTEGOS 프로토콜은 개체인증, key freshness를 일방향 제공하며 양방향으로 묵시적 키 인증을 제공한다. 분석결과는 〔표 7〕과 같다.

2.2 ISO/IEC 11770

ISO/IEC 11770에서는 대칭키를 이용한 키 분배 방식과 공개키를 이용한 키 분배 방식에 대해 기술하고 있으며, 이를 이용한 방식으로 객체간 키 동의 방식(Point to Point), 신뢰기관을 이용한 분배 방식에 대해 설명하고 있다.

■ 시스템 파라미터
- x : 사용자 X의 ID
- T : KDC / KTC 의 ID
- F : 키 데이터
- K_{XY} : 사용자 X와 Y사이의 비밀키
- R : 랜덤수
- R_X : X에 의해 발행된 랜덤수
- T/N : 타임스탬프 또는 일련번호(time stamp /sequence number)
- T_X/N_X : X에 의해 발행된 타임스탬프
- TVP : 시간 변수 인자
- eK(Z) : 키 K를 사용한 대칭 알고리즘으로 데이터 Z를 암호화
- dK(Z) : 키 K를 사용한 대칭 알고리즘으로 데이터 Z를 복호화
- vK(Z) : 메시지 인증 코드(MAC), macK(Z)
- f : 키 생성 함수
- X||Y : 데이터 X, Y의 연접

사용자 A	공개정보 P_B, VK_A	사용자 B
P_B 확인 $r \in_{R/S} \{1, \cdots, n-1\}$ $BE = (A \| K) \oplus kdf(\pi((r \cdot 1)(hP_B)), parameters)$ $KT_{A1} = B \| TVP \| rG \| BE$ $S_{SK_A}(KT_{A1})$	$S_{SK_A}(KT_{A1})$ KT_{A1} ⟶	$V_{VK_A}(KT_{A1})$ B, TVP, rG 확인 $A \| K = BE \oplus kdf(\pi((d_B \cdot 1)(h \cdot rG)), parameters)$ A 확인

〔그림 7〕 KTEGOS 프로토콜

2.2.1 대칭키를 이용한 키 분배 방식

① Point-to-Point 키 동의 방식 1

1) 사용자 B는 랜덤수 R_B를 선택하여 사용자 A에게 전송한다.
2) 랜덤수를 수신한 사용자 A는 B의 ID와 랜덤수 그리고 키 데이터를 연접한 후 사용자 A, B간의 세션키로 암호화한다.

$$eK_{AB}(R_B||B||F||Text\ 1)$$

Point-to-Point 키 동의 방식 1의 수행과정은 [그림 8]과 같고 2회의 통신회수와 사용자 A에 대한 개체인증을 제공한다. 분석결과는 [표 8]에서 정리하였다.

사용자 A		사용자 B
세션키로 암호화 $eK_{AB}(R_B\|\|B\|\|F\|\|Text\ 1)$	R_B ← $eK_{AB}(R_B\|\|B\|\|F\|\|Text\ 1)$ →	R_B선택

[그림 8] point to point 프로토콜 1

[표 8] point to point 프로토콜 1 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key freshness
PtoP 1	2	A	-	-	-

② Point-to-Point 키 동의 방식 2

1) 사용자 B는 아래와 같은 방법으로 키 데이터와 A의 ID, 타임스탬프를 세션키 K_{AB}로 암호화하여 사용자 A에게 전송해 준다.

$$E_{B1} = eK_{AB}(T_B/N_B||A||F_B||Text\ 2)$$

2) 사용자 A는 아래와 같은 방법으로 키 데이터와 B의 ID, 타임스탬프를 세션키 K_{AB}로 암호화하여 사용자 B에게 전송해 준다.

$$E_{A1} = eK_{AB}(T_A/N_A||B||F_A||Text\ 1)$$

3) 사용자 A와 B는 키 생성 함수 f를 사용하여 다음과 같은 방법으로 키를 생성한다.

$$K = f(F_A, F_B)$$

[그림 9]는 point to point 프로토콜 2의 수행과정이며, 이 프로토콜은 2회의 통신회수를 갖으며 사용자 A,B 모두에 대한 개체인증과 일방향 묵시적 키 인증을 제공한다. 분석결과는 [표 9]와 같다.

사용자 A		사용자 B
$E_{A1} =$ $eK_{AB}(T_A/N_A\|\|B\|\|F_A\|\|Text\ 1)$	E_{A1} ← E_{B1} →	$E_{B1} = eK_{AB}(T_B/N_B\|\|A\|\|F_B\|\|Text\ 2)$

[그림 9] point to point 프로토콜 2

[표 9] point to point 프로토콜 2 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key freshness
PtoP 2	2	A,B	-	일방향	-

③ Point-to-Point 키 동의 방식 3

1) 사용자 B는 랜덤수 R_B를 선택해서 사용자 A에게 전송한다.

2) 사용자 A는 아래와 같이 자신의 랜덤값, B의 랜덤값, B의 ID 등을 세션키 K_{AB}로 암호화하여 사용자 B에게 전송한다.

$$E_{A1} = eK_{AB}(R_A||R_B||B||F_A||Text\ 1)$$

3) 사용자 B는 아래와 같이 자신의 랜덤값, A의 랜덤값, 자신의 키 자료정보 등을 세션키 K_{AB}로 암호화하여 사용자 A에게 전송한다.

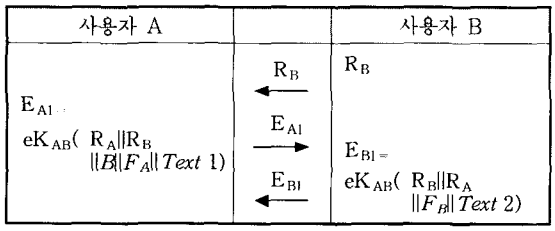
$$E_{B1} = eK_{AB}(R_B||R_A||F_B||Text\ 2)$$

4) 사용자 A와 B는 키 생성 함수 f를 사용하여 키를 생성한다.

$$K = f(F_A, F_B)$$

[그림 10]은 point to point 프로토콜 3의 수행과정이다. 이 프로토콜은 3회의 통신회수를 갖으며 사용

자 A,B 모두에 대한 개체인증과 양방향 묵시적 키 인증을 제공한다. 분석결과는 〔표 10〕과 같다.



〔그림 10〕 point to point 프로토콜 3

〔표 10〕 point to point 프로토콜 3 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key freshness
PtoP 3	3	A,B	-	양방향	-

④ KDC(Key Distribution Centre) 프로토콜 1

1) 사용자 A는 통신하고자 하는 사용자 B의 ID정보와 타임스탬프를 KDC에게 전송한다

$$TVP_A||B$$

2) KDC는 키 데이터와 타임스탬프, 사용자 A의 ID를 사용자 B와의 세션키로 암호화하고, 키 데이터와 타임스탬프, 사용자 B의 ID를 사용자 A와의 세션키로 암호화하여 아래와 같이 사용자 A에게 전송한다.

$$eK_{AT}(TVP_A||F||B||Text)||eK_{BT}(T_T/N_T||F||A||Text\ 2)$$

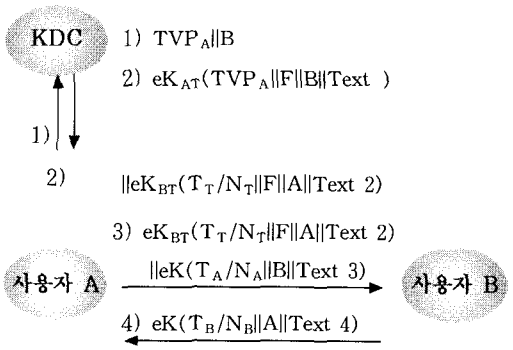
3) 사용자 A는 KDC와의 세션키로 복호화하여 검증한다. 나머지 암호문과 사용자 B의 ID와 타임스탬프 키 데이터를 생성한 세션키 K로 암호화한 암호문을 아래와 같은 방법으로 연접하여 사용자 B에게 전송한다.

$$eK_{BT}(T_T/N_T||F||A||Text\ 2)||eK(T_A/N_A||B||Text\ 3)$$

4) 사용자 B는 수신한 암호문을 복호화하여 검증 후 사용자 A의 ID와 타임스탬프를 키 데이터 F로 생성한 세션키 K로 아래와 같이 암호화하여 사용자 A에게 전송한다.

$$eK(T_B/N_B||A||Text\ 4)$$

KDC 프로토콜의 수행 과정은 〔그림 11〕과 같다.



〔그림 11〕 KDC 프로토콜 1

KDC 프로토콜 1의 분석결과는 〔표 11〕과 같다.

〔표 11〕 KDC 프로토콜 1 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key freshness
KDC 1	3(4)	opt	opt	양방향	-

* opt : optional

⑤ KDC 프로토콜 2

1) 사용자 A는 B의 ID와 타임스탬프, 키 데이터를 아래와 같이 KDC와의 세션키로 암호화하여 KDC에게 전송한다.

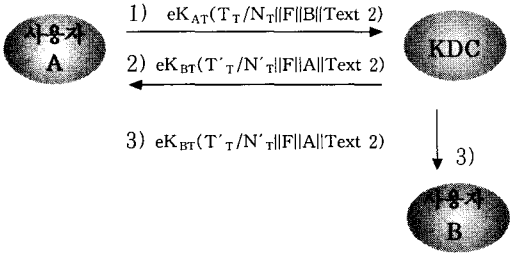
$$eK_{AT}(T_A/N_A||B||Text\ 1)$$

2) KDC는 사용자 A의 정보와 키 데이터, 타임스탬프를 사용자 A와의 세션키로 아래와 같이 암호화하여 사용자 A에게 전송한다.

$$eK_{AT}(T_T/N_T||F||B||Text\ 2)$$

3) KDC는 사용자 B의 정보와 키 데이터, 타임스탬프를 사용자 B와의 세션키로 아래와 같이 암호화하여 사용자 A에게 전송한다.

$eK_{BT}(T'_T/N'_T||F||A||Text\ 2)$



〔그림 12〕 KDC 프로토콜 2

〔표 12〕 KDC 프로토콜 2 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key freshness
KDC 2	3	-	-	양방향	-

⑥ KTC 프로토콜 1

1) 사용자 A는 KTC에게 아래와 같은 방법으로 사용자 B의 ID, 키 데이터, 타임스탬프를 연접하고 KTC와의 세션키로 암호화하여 전송한다.

$$eK_{AT}(TVP_A||B||F||Text\ 1)$$

2) KTC는 사용자 B의 ID와 타임스탬프를 B의 세션키 K_{BT}로 암호화한 암호문과 사용자 A의 ID, 키 데이터, 타임스탬프를 A의 세션키 K_{AT}로 암호화한 암호문을 아래와 같이 연접하여 사용자 A에게 전송한다.

$$eK_{AT}(TVP_A||B||Text\ 2)\ ||eK_{BT}(T_T/N_T||F||A||Text\ 3)$$

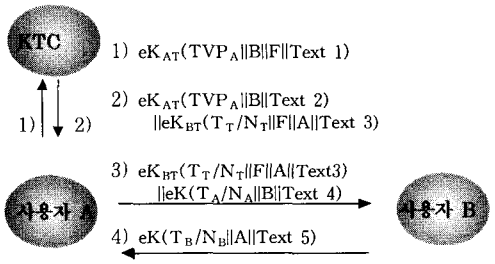
3) 사용자 A는 KTC에게 받은 암호문을 복호화 후 KTC로부터 수신한 암호문과 사용자 B의 정보와 타임스탬프를 사용자 A, B의 세션키로 암호화한 정보를 아래와 같이 연접하여 사용자 B에게 전송한다.

$$eK_{BT}(T_T/N_T||F||A||Text3)\ ||eK(T_A/N_A||B||eText\ 4)$$

4) 사용자 B는 키 확인을 위해 사용자 A의 ID와 타임스탬프를 세션키 K로 아래와 같이 암호화하여 사용자 A에게 전송한다.

KTC 프로토콜 1에 대한 수행과정은 다음 〔그림 13〕과 같다.

$$eK(T_B/N_B||A||Text\ 5)$$



〔그림 13〕 KTC 프로토콜 1

프로토콜의 4번째 연산을 수행함으로써 상호인증 및 키 확인을 제공한다. 분석결과는 〔표 13〕과 같다.

〔표 13〕 KTC 프로토콜 1 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key freshness
KTC 1	3(4)	opt	opt	양방향	-

*opt : optional

⑦ KTC 프로토콜 2

1) 사용자 B는 랜덤수 R_B를 선택하여 통신하고자 하는 A에게 전송한다.
2) 사용자 A는 KTC에게 랜덤수 R_B와 자신이 선택한 랜덤수 R_A, 사용자 B의 ID를 아래와 같이 KTC와의 세션키로 암호화하여 전송한다.

$$eK_{AT}(R_A||R_B||B||F||eText\ 1)$$

3) KTC는 수신한 암호문을 복호화한 후 아래와 같이 사용자들에게 전송해 줄 정보를 각각의 사용자 세션키 K_{AT}, K_{BT}로 암호화하여 사용자 A에게 전송해 준다.

$$eK_{AT}(R_A||B||Text\ 2)||eK_{BT}(R_B||F||A||Text\ 3)$$

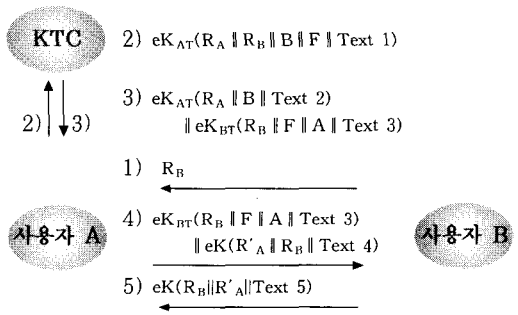
4) 사용자 A는 KTC에게 받은 암호문을 검증 후 KTC와 사용자 B의 세션키로 암호화된 암호문, 랜덤수 R_B와 랜덤수 R'_A를 사용자 A, B의 세션키로 암호화한 암호문을 아래와 같이 연접하여 사용자 B에게 전송해 준다.

$eK_{BT}(R_B||F||A||Text\ 3)||eK(R'_A||R_B||Text\ 4)$

5) 사용자 B는 암호문을 복호하여 검증 후 사용자 A에게 생성된 세션키로 랜덤수 R_B와 R'_A를 아래와 같이 암호화하여 전송해 준다.

$$eK(R_B||R'_A||Text\ 5)$$

프로토콜의 5) 연산을 수행함으로써 상호인증 및 키 확인을 제공한다. 5)를 수행하지 않으면 일방향 객체 인증을 제공한다.



〔그림 14〕 KTC 프로토콜 2

〔표 14〕 KTC 프로토콜 2 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key freshness
KTC 2	3(4)	opt	opt	양방향	-

*opt : optional

〔표 15〕는 지금까지 살펴본 ISO/IEC 11770의 대칭키 기반 키 분배 프로토콜의 특징을 간략히 정리한 것이다.

〔표 15〕 ISO/IEC 11770 대칭키를 이용한 키 분배 방식

구분	통신횟수	키 제어	키 인증	키 확인	객체 인증
1	2	A	일방향	-	일방향
2	2	A/B	양방향	-	양방향
3	3	A/B	양방향	-	양방향
4	3(4)	KDC	양방향	opt	opt
5	3	KDC	양방향	-	-
6	3(4)	A	양방향	opt	opt
7	4(5)	A	양방향	opt	opt

2.2.2 공개키를 이용한 키 분배 방식

① 키 동의 프로토콜 1

1) 사용자 A는 랜덤수 r_A를 선택하여 $F(r_A, g)$를 계산한 값 KT_{A1}을 사용자 B에게 전송해 준다.

$$KT_{A1} = F(r_A, g)||Text1$$

2) 사용자 B도 랜덤수 r_B를 선택하여 $F(r_B, g)$를 계산한 값 KT_{B1}을 사용자 A에게 전송해 준다.

$$KT_{B1} = F(r_B, g)||Text2$$

3) 사용자 A와 B는 KT_{A1}와 KT_{B1}에서 각각 $F(r_A, g)$를 분리하여 아래와 같은 방법으로 세션키 K_{AB}를 생성한다.

KT_{A1}에서 $F(r_A, g)$ 분리
$K_{AB} = NK$
KT_{B1}에서 $F(r_B, g)$를 분리
$K_{AB} = w(F(r_A, p_B), F(r_A, F(r_B, g)))$

사용자 A		사용자 B
H에서의 랜덤수 r_A생성 $F(r_A, g)$계산 $KT_{A1} = F(r_A, g)\|\|Text1$	KT_{A1} →	
		H에서의 랜덤수 r_B생성 $F(r_B, g)$계산 $KT_{B1} = F(r_B, g)\|\|Text2$
	← KT_{B1}	KT_{A1}에서 $F(r_A, g)$ 분리 $K_{AB} = w(F(r_B, F(r_A, g)), F(r_B, p_A))$
KT_{B1}에서 $F(r_B, g)$를 분리 $K_{AB} = w(F(r_A, p_B), F(r_A, F(r_B, g)))$		

〔그림 15〕 키 동의 프로토콜 1

키 동의 프로토콜 1의 과정은 [그림 15]와 같이 나타낼 수 있으며 이에 대한 분석은 [표 16]과 같다.

[표 16] 키 동의 프로토콜 1 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key 연산
키 동의 1	2	-	opt	양방향	2,2

② 키 동의 프로토콜 2

1) 사용자 A는 랜덤수 r_A를 생성하고 아래와 같은 방법으로 생성한 KT_{A1}을 사용자 B에게 전송해 준다.

$$KT_{A1} = r_A || Text1$$

2) 사용자 B는 랜덤수 r_B를 생성하고 아래와 같이 랜덤수 r_A, r_B와 사용자 A의 ID를 연접한 값에 서명키 S_B로 서명한 중간값 BS를 생성한다. 서명문 BS와 사용자 B의 ID를 아래와 같이 연접하고 사용자 A의 공개키 E_A로 암호화하여 사용자 A에게 전송한다.

$$BS = S_B(A||r_A||r_B||Text2)$$
$$KT_{B1} = E_A(B||BS||Text3)||Text4$$

3) 사용자 A는 비밀키로 암호문 KT_{B1}을 복호하고, 서명문 BS의 서명을 검증 후 BS에 포함되어 있는 사용자 B의 서명의 전부 또는 일부분으로 세션키를 구성한다.]

키 동의 프로토콜 2의 과정은 [그림 16]과 같고 이에 대한 분석결과는 [표 17]에 정리하였다.

[표 17] 키 동의 프로토콜 2 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key 연산
키 동의 2	2	일방향	opt	일방향	2,2

*opt : optional

③ 키 동의 프로토콜 3

1) 사용자 A는 랜덤수 r_A를 생성하고, 아래와 같이 함수 $F(r_A, g)$를 계산하여 사용자 B에게 전송한다.

$$KT_{A1} = F(r_A, g)||Text1$$

2) 사용자 B는 함수 $F(r_B, g)$와 $F(r_B, F(r_A, g))$의 계산을 통해 세션키 K_{AB}를 생성한 후 아래와 같은 방법으로 토큰 DB_1과 KT_{B1}를 생성하여 KT_{B1}을 사용자 A에게 전송해 준다.

$F(r_B, g)$계산
$K_{AB} = F(r_B, F(r_A, g))$
$DB_1 = F(r_B, g)||F(r_A, g)||A||Text2$
$KT_{B1} = S_B(DB_1)||f_{K_{AB}}(DB_1)||Text3$

3) 사용자 A는 수신된 메시지를 검증하고, 아래와 같은 방식으로 DB_2과 KT_{A2}를 생성하여 KT_{A2}를 사용자 B에게 전송해 준다.

$K_{AB} = F(r_A, F(r_B, g))$
$f_{K_{AB}}(DB_1)$ 확인
$DB_2 = F(r_A, g)||F(r_B, g)||B||Text4$
$KT_{A2} = S_A(DB_2)||f_{K_{AB}}(DB_2)||Text5$

사용자 A		사용자 B
H에서의 랜덤수 r_A생성 $KT_{A1} = r_A||Text1$	KT_{A1} →	H에서의 랜덤수 r_B생성 $BS = S_B(A||r_A||r_B||Text2)$ $KT_{B1} = E_A(B||BS||Text3)||Text4$
$D_A(KT_{B1})$, $V_B(BS)$ BS에 포함되어 있는 B의 서명의 전부 또는 일부분으로 세션키 구성	← KT_{B1}	BS에 포함되어 있는 B의 서명의 전부 또는 일부분으로 세션키 구성

[그림 16] 키 동의 프로토콜 2

사용자 A		사용자 B
H에서의 랜덤수 r_A생성 $F(r_A, g)$계산 $KT_{A1} = F(r_A, g)\|\|Text1$	KT_{A1} →	H에서의 랜덤수 r_B생성 $F(r_B, g)$계산 $K_{AB} = F(r_B, F(r_A, g))$ $DB_1 = F(r_B, g)\|\|F(r_A, g)\|\|A\|\|Text2$
KT_{B1}에서 B의 서명 검증 (A, $F(r_A, g)$확인) $K_{AB} = F(r_A, F(r_B, g))$ $f_{K_{AB}}(DB_1)$ 확인 $DB_2 = F(r_A, g)\|\|F(r_B, g)\|\|B\|\|Text4$ $KT_{A2} = S_A(DB_2)\|\|f_{K_{AB}}(DB_2)\|\|Text5$	← KT_{B1} KT_{A2} →	$KT_{B1} = S_B(DB_1)\|\|f_{K_{AB}}(DB_1)\|\|Text3$ KT_{A2}에서 A의 서명 검증 (B, $F(r_A, g)$, $F(r_B, g)$확인) $K_{AB} = F(r_A, F(r_B, g))$ $f_{K_{AB}}(DB_2)$ 확인

〔그림 17〕 키 동의 프로토콜 3

4) 사용자 B는 아래와 같은 방법으로 수신된 메시지를 검증한다.

$K_{AB} = F(r_A, F(r_B, g))$
$f_{K_{AB}}(DB_2)$ 확인

키 동의 프로토콜 3에 대한 수행 과정은 〔그림 17〕과 같다.

〔표 18〕 키 동의 프로토콜 3 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key 연산
키 동의 3	3	양방향	양방향	양방향	3,3

④ 키 전송 프로토콜 1

1) 사용자 A는 랜덤수 r_A를 선택하고 아래와 같은 방식으로 KT_{A1}을 생성하여 사용자 B에게 전송해 준다.

$KT_{A1} = r_A\|\|Text1$

2) 사용자 B는 생성한 세션키를 A의 공개키로 암호화하여 BE을 생성한다. 아래와 같이 랜덤수 r_B, 랜덤수 r_A, BE, 사용자 A의 ID를 연접하고 사용자 B의 서명키 S_B로 서명하여 전송해 준다.

$BE = E_A(B\|\|K\|\|Text2)$
랜덤수 r_B선택
$KT_{B1} = S_B(A\|\|r_A\|\|r_B\|\|BE\|\|Text3)\|\|text4$

3) 사용자 A는 아래와 같은 방법으로 수신한 메시지의 서명을 검증하고 암호문을 복호화하여 키를 확인한다.

$V_B(S_B(A\|\|r_A\|\|r_B\|\|BE\|\|Text3))$
(A, r_A확인)
$D_A(BE)$
(B확인)

키 전송 프로토콜 1에 대한 사용자간의 수행과정은 〔그림 18〕과 같으며, 이 프로토콜은 2회의 통신회수와 개체인증, 키 확인, 묵시적 키 인증을 모두 일방향으로 제공한다. 분석결과에 대한 정리는 〔표 19〕와 같다.

사용자 A		사용자 B
랜덤수 r_A생성 $KT_{A1} = r_A\|\|Text1$	KT_{A1} →	
		$BE = E_A(B\|\|K\|\|Text2)$ 랜덤수 r_B생성 $KT_{B1} =$ $V_B(S_B(A \| r_A \| r_B \| BE \| Text3))$
$V_B(S_B(A \| r_A \| r_B \| BE \| Text3))$ (A, r_A확인) $D_A(BE)$ (B확인)	← KT_{B1}	

〔그림 18〕 키 전송 프로토콜 1

〔표 19〕 키 전송 프로토콜 1 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key 연산
키 전송 1	2	일방향	일방향	일방향	4.4

⑤ 키 전송 프로토콜 2

1) 사용자 A는 랜덤수 r_A를 생성하고, 아래와 같은 방법으로 KT_{A1}을 생성하여 사용자 B에게 전송해준다.

$$KT_{A1} = r_A \| Text1$$

2) 사용자 B는 자신의 ID와 생성한 세션키 K_B를 사용자A의 공개키로 암호화한 중간값 BE_1을 생성한다. 아래와 같은 방법으로 생성한 랜덤수 r_B와 랜덤수 r_A, 사용자 A의 ID, 중간값 BE_1을 연접한 후 자신의 비밀키로 서명하여 사용자 A에게 전송해 준다.

$BE_1 = E_A(B\|K_B\|Text2)$
랜덤수 r_B생성
$KT_{B1} = S_B(r_B\|r_A\|A\|BE_1\|Text3)\|Text4$

3) 사용자 A는 아래와 같은 방법으로 수신한 서명문의 서명을 검증하고 복호화하여 메시지를 확인한다. 사용자 A는 자신의 ID와 키 K_A을 연접하고 사용자 B의 공개키로 암호화하여 BE_2를 생성한다. 생성된 BE_2와 랜덤수 r_A, r_B, 사용자 B의 ID를 연접하고 사용자 A의 비밀키로 서명하여 사용자 B에게 전송해 준다.

$V_B(S_B(r_B\|r_A\|A\|BE_1\|Text3))$ (A, r_A확인)
$D_A(BE_1)$ (B확인, K_B)
$BE_2 = E_B(A\|K_A\|Text5)$

$KT_{A2} = S_A(r_A\|r_B\|B\|BE_2\|Text6)\|Text7$

4) 사용자 B는 아래와 같은 방법으로 서명을 검증하고 복호화하여 메시지를 확인한다.

$V_A(S_A(r_A\|r_B\|B\|BE_2\|Text6))$
(B, r_A, r_B확인)
$D_B(BE_2)$ (A확인, K_A)

키 전송 프로토콜 2에 대한 수행 과정은 〔그림 19〕와 같다. 이에 대한 분석결과는 〔표 20〕과 같다.

〔표 20〕 키 전송 프로토콜 2 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	Key 연산
키 전송 2	3	양방향	opt	양방향	4.4

*opt : optional

⑥ 키 전송 프로토콜 3

1) 사용자 A는 랜덤수 r_A를 선택하고 키 K_A를 생성한다. 아래와 같은 방법으로 K_A, r_A, 사용자 A의

사용자 A		사용자 B
랜덤수 r_A생성 $KT_{A1} = r_A\|Text1$	KT_{A1} →	
		$BE_1 = E_A(B\|K_B\|Text2)$ 랜덤수 r_B생성 $KT_{B1} = S_B(r_B\|r_A\|A\|BE_1\|Text3)\|Text4$
$V_B(S_B(r_B\|r_A\|A\|BE_1\|Text3))$ (A, r_A확인) $D_A(BE_1)$ (B확인, accept K_B) $BE_2 = E_B(A\|K_A\|Text5)$ $KT_{A2} = S_A(r_A\|r_B\|B\|BE_2\|Text6)\|Text7$	← KT_{B1} KT_{A2} →	
		$V_A(S_A(r_A\|r_B\|B\|BE_2\|Text6))$ (B, r_A, r_B확인) $D_B(BE_2)$ (A확인, accept K_A)

〔그림 19〕 키 전송 프로토콜 2

ID를 연접하고 사용자 B의 공개키로 암호화하여 BE_1을 생성한다. 아래와 같이 생성한 BE_1를 사용하여 KT_{A1}을 생성하고, 사용자 A에게 전송한다.

$$BE_1 = E_B(A||K_A||r_A||Text1)$$
$$KT_{A1} = BE_1||Text2$$

2) 사용자 B는 수신한 암호문을 복호화하여 정당한 메시지인지를 확인한다. 아래와 같은 방법으로 랜덤수 r_B와 키 값 K_B를 생성하고 사용자 B의 ID, 랜덤수 r_A와 연접하여 사용자 A의 공개키로 암호화하여 BE_2를 생성한다. 생성한 BE_2은 아래와 같은 방법을 사용하여 KT_{B1}을 생성하고, 사용자 B에게 KT_{B1}을 전송한다.

$$D_B(BE_1)\ (A\text{ 확인})$$
$$BE_2 = E_A(B||K_B||r_A||r_B||Text3)$$
$$KT_{B1} = BE_2||Text4$$

3) 사용자 A는 암호문을 복호화하여 메시지를 확인하고, 사용자 B의 랜덤수로 생성한 KT_{A2}를 전송해 주어 메시지 인증을 제공한다.

$$D_A(BE_2)\ (r_A\text{ 확인})$$
$$KT_{A2} = r_B||Text5$$

4) 사용자 B는 자신의 랜덤수를 확인한다.

$$KT_{A2}\text{에서 } r_B \text{ 확인}$$

키 전송 프로토콜 3에 대한 수행 과정은 [그림 20]과 같고, 3회의 통신회수를 갖으며 개체인증, 키 확인, 묵시적 키 인증을 모두 양방향으로 제공한다. 이에 대한 분석결과는 [표 21]에 정리하였다.

[표 21] 키 전송 프로토콜 3 분석

구 분	통신 회수	개체 인증	키 확인	묵시적 키 인증	Key 연산
키 전송 3	3	양방향	양방향	양방향	2,2

[표 22]에서 지금까지 살펴본 ISO/IEC 11770의 공개키 기반 키 분배 프로토콜의 특징을 간략히 정리하였다.

[표 22] ISO/IEC 11770 공개키를 이용한 키 분배 방식

구 분	통신 회수	개체 인증	키 확인	묵시적 키인증	Key 연산
키 동의 1	2	-	opt	양방향	2,2
키 동의 2	2	일방향	opt	일방향	2,2
키 동의 3	3	양방향	양방향	양방향	3,3
키 전송 1	2	일방향	일방향	일방향	4,4
키 전송 2	3	양방향	opt	양방향	4,4
키 전송 3	3	양방향	양방향	양방향	2,2

V. 결 론

현재 키 관리와 관련한 표준 규격은 ISO 11568, 15946, IEEE P1363, ANSI x9.42, x9.63, KS x6318 등의 여러 표준 규격이 제정되어 있다.

본 논문에서는 ISO/IEC 키 관리 표준인 ISO/IEC 11770과 ISO/IEC 15946-3에 정의된 것을 키 생성과 키 분배로 분류하여 분석하였다.

ISO/IEC는 키 생성과 키 분배에 대하여 정의하고 있는데 키 생성은 ISO/IEC 11770에 기술되어 있는 것을 분석하였으며 키 분배는 ISO/IEC 15946-3에

사용자 A		사용자 B
$BE_1 = E_B(A\|\|K_A\|\|r_A\|\|Text1)$ $KT_{A1} = BE_1\|\|Text2$	KT_{A1} →	$D_B(BE_1)$ (A확인) $BE_2 = E_A(B\|\|K_B\|\|r_A\|\|r_B\|\|Text3)$ $KT_{B1} = BE_2\|\|Text4$
$D_A(BE_2)$ (r_A 확인) $KT_{A2} = r_B\|\|Text5$	← KT_{B1} KT_{A2} →	KT_{A2}에서 r_B 확인

[그림 20] 키 전송 프로토콜 3

서 타원 곡선 기반의 키 분배 프로토콜에 대하여 설명하고 있으며 키 교환 방법을 키 동의와 키 전송으로 나누어 기술하고 있다. ISO/IEC 11770에 기술된 키 분배 방식은 대칭키를 이용한 분배방식과 공개키를 이용한 분배 방식에 대하여 기술하고 있으며 이를 통하여 두 객체만이 참여하는 방식과 키 분배 센터(KDC)와 키 전송 센터(KTC) 같은 신뢰기관을 이용한 방식에 대하여 기술되어져 있다.

그러나, ISO/IEC에서 키 생성과 분배 이 외의 다른 요소들에 관한 사항은 설명이 미약하거나 제외되어 있는 부분들이 있다. 따라서 현재 필요한 키 관리 기술을 충분히 설명하지 못 하는 면이 많아 향후 다른 기술 표준들과의 상호 보완을 통하여 키 관리 기술에 대한 포괄적인 내용을 포함하는 표준이 요구된다.

참 고 문 헌

[1] ANSI X9.42, "Agreement of symmetric keys Using discrete Logarithm Cryptography", *ANSI*,1998.

[2] ANSI X9.63, "Public Key Cryptography for the financial Services Industry", *ANSI*, 1999.

[3] ANSI X9.69, "Framework for Key Management Extensions", *ANSI*, 1998.

[4] FIPS-PUB 185, "Escrowed Encryption Standard", *NIST*, 1994.

[5] ISO 10202-7, "Financial transaction cards Security architecture of financial transaction systems using integrated circuit card - part 7: Key management", *ISO*, 1998.

[6] ISO 11568, "Banking- Key management (retail)", *ISO*, 1998

[7] ISO/IEC 15946, "Information technology - Security techniques - Cryptographic techniques based on elliptic curves", *ISO/IEC*, 1994.

[8] ISO/IEC 11770, "Information technology - Security techniques - Key management", *ISO/IEC*, 1996.

[9] IEEE P1363, "Standard Specifications For Public Key Cryptography", *IEEE*, 2001.

[10] PKCS #3, "Diffie-Hellman Key- Agreement Standard", *RSA research*, 1999.

〈著 者 紹 介〉

조 은 성 (Eun-Sung Cho)

2000년 8월 : 성균관대학교 산업공학과 공학사
2002년 8월 : 성균관대학교 대학원 산업공학과 공학석사
2003년 2월~현재 : 성균관대학교 대학원 컴퓨터공학과 박사과정

정 영 석 (Young-Seok Chung)

2002년 2월 : 성균관대학교 정보공학과 졸업(공학사)
2002년 3월~현재 : 성균관대학교 정보통신공학부 석사 과정

오 수 현 (Soo-Hyun Oh)

1998년 2월 : 성균관대학교 정보공학과 공학사
2000년 2월 : 성균관대학교 대학원 전기전자 및 컴퓨터공학부 공학석사
2003년 8월 : 성균관대학교 대학원 전기전자 및 컴퓨터공학부 공학박사

양 형 규 (Hyung-Kyu Yang) 정회원

1983년 2월 : 성균관대학교 전자공학과 학사
1985년 2월 : 성균관대학교 전자공학과 석사
1995년 2월 : 성균관대학교 정보공학과 박사
1984년 12월~1991년 2월 : 삼성전자 컴퓨터부문 선임연구원
1995년 3월~현재 : 강남대학교 컴퓨터미디어공학부 부교수

원 동 호 (Dong-Ho Won) 종신회원

성균관대학교 전자공학과
(학사, 석사, 박사)
한국전자통신연구소 전임 연구원
일본 동경공대 객원연구원
성균관대학교 교학처장, 전기전자 및 컴퓨터공학부장, 정보통신대학원장, 국무총리실 정보화추진위원회 자문위원
한국정보보호학회 이사, 부회장, 수석부회장, 회장
현재 : 성균관대학교 정보통신공학부 교수
성균관대학교 연구지원처장
정통부지정 정보보호인증기술연구센터 센터장