기업비밀유통을 위한 MSEC 기반 그룹 키 관리 프로토콜 설계와 구현 연구☆

The Study on Design and Implementation of MSEC-based Group Key Management Protocol for Corporate Secret Distribution

최 정 현*
Cheong Hyeon Choi

요 약

국내 기업의 우수 기술 관련 기밀 정보량은 늘어났지만 정보 유통의 디지털화로 유출 위험은 커지고 불법유출 경로를 파악하기가 매우 어려워서, 실제 유출 사례가 늘어감에도 막을 대책이 없어 재정적 피해는 점점 커지고 있다. 그러나 현재 문서보호 시스템은 저작물 보호 차원에서 설계되어 있어서, 합법적 사본이 유출될 경우 불법적 사용 방지 및 추적에 무방비이고, 내부자가 공모한 유출에는 대책이 전혀 없다는 것이 문제이다. 기업비밀의 특성상 폐쇄된 멀티캐스트 그룹 내 유통으로 제한하는 것이 적합하지만 그룹 기반 표준 프로토콜을 적용하여 기업비밀보호에 필요한 보호조건을 충분히 만족시키는 설계 연구가 없었다. 본 연구는 원천봉쇄 시스템 구조 설계 연구의 후속 연구로서 MSEC 기반 키-관리 프로토콜을 IGMP, SNMP와 접목하여 완전한 보호기능을 달성하는 비밀정보 유통 프로토콜을 제안하고, 프로토콜 구현의 단계적 시간 순서 차트까지 설계하였다. 본 연구는 비밀정보보호의 강력 보안조건인 암호화 유통원칙, 키와 정보 분리원칙, 반출시 처리와 저장의 분리원칙, 사용자와 호스트의 이중 인증 원칙을 적용한 키-관리 프로토콜 및 비밀정보 유통 프로토콜을 제안한다.

ABSTRACT

Recently competitive Korean companies are suffered from financial loss due to illegal exposure of their own proprietary know-how secrets, since it is difficult to watch hidden illegal channels to leak them due to their digitalization. Today the DRM-based system designed to protect such secrets is insufficient to prevent it, since DRM-based protection system cannot defend the intelligent robbery of secrets, in special, employee's robbery. The MSEC is much appropriate to secure secrets against employee's robbery. Our paper notes that IGMP, MSEC and SNMP can work easily together to realize secure system that satisfy strong security condition for prevention from leaking secrets. Since the previous research was on the architectural design for prevention of illegal exposure, this paper proposes the efficient protocol based on MSEC protocol. Our protocol satisfies the strong security conditions that the principles that the secret should be stored/distributed only in an encrypted shape, and should be separated physically from its encryption key, and should be carried in registered mobile storage separate from its processing device, and should be verified in terms of both user and device. Thus this paper proposes both the protocol for secret document distribution and its group key management.

☞ keyword : Multicast Security(멀티캐스트 보안), Group Key Management(그룹키관리), Network Security(망보안)

1. 서 론

주요 비밀정보의 양적 증가와 통신망을 통한 디지털 형태로의 정보 유통이 급격히 늘어나면서, 최근 기업에서는 디지털 비밀정보 보호에 대한 관심이 커지고 있다. 기업비밀의 잠재적 가치로 볼 때 경쟁 상대 특히 해외(예: 중국)로 불법 유출될 경우, 해당 기업은 물론이고 국가 경쟁력 측면에서도 그 피해가 매우 심각하다.

특히 디지털 기업비밀정보에 대해서는 기업 외부로의 반출은 가능하면 통제하고, 내부 유통까지도 합법성 검사와 처리 과정을 실시간 감시하고,

* 종신회원 : 광운대학교 경영정보학과 교수
chchoi@kw.ac.kr
[2010/11/05 투고 - 2010/11/08 심사 - 2010/12/06 심사완료]
☆ 본 논문은 광운대학교 2006년도 학술지원으로 작성됨

유통 경로를 추적하여, 위험 경로 및 불법유출 가능 지점을 색출하고, 필요하면 봉쇄하고, 합법적 사용자들 모두에게도 정보 노출에 따른 법적 책임을 고지하고, 내부자가 공모할지라도 불법 유출이 불가능한 강력한 시스템의 구축이 필요한 시기가 되었다[1,2].

현재는 기업 비밀정보를 보호 수준이 상대적으로 낮은 지적재산 저작물의 보호차원에서 다루고 있다. 그러나 저작물과 기업비밀정보는 보호 범위와 수준이 전혀 다르다는 점을 주목해야 한다.

지적재산 저작물(예: 음악, 영화)의 보호 수준은 저작물을 구매할 때 이용할 권리를 주고 저작물 재생 시 정당한 구매자인지만 확인하면 그 후 구매자는 이용에 영속적 권리를 가지게 되고 그 후 시간적 공간적 제약이 없이 저작물의 이용이 가능하다. 이런 저작물 보호 수준의 취약점은 이용권을 일회 검사한 후에는 타인에게 양도(불법유출)하는 것을 막거나 추적하는 것이 불가능하다는 점이다.

저작물의 불법유출에 대한 유일한 대비책은 디지털 워터마크(watermark)를 삽입하여 합법적 구매자의 재생장치에서 왜곡을 보정하는 워터마크 복호키로 정상 재생되도록 하는 방식이다[9,10]. 그러나 이 방식은 저작물 재생장치 제조사와의 합의된 표준이 필요하고 더욱이 현재는 이용할 수 없는 방식이다. 디지털 저작물 유통관리 DRM(digital right management)이 제공하는 보호 수준은 통신 과정에서 해킹을 막는 암호화 전송과 이용자 합법성 검증 후 복호키를 전송해주는 수준이 전부이다.

그러나 기업 비밀정보는 저작물과 달리 정보에 대한 접근 권한이 사용자의 이직, 퇴직, 승진, 전보 등으로 시간이 지남에 따라 변화하는 시간적 제약이 있고 유통 그룹 내 장치들에서만 정보 접근이 가능토록 한 공간적 제약, 시간에 따라 호스트의 교체, 이동, 폐기 등으로 그 권한이 변하는 동적 특징을 가진다.

어떤 비밀정보도 디지털 형태나 인쇄물 형태로 유통되어야 할 경우는 관리자의 특별한 허가가 있어야 하며, 허락을 얻은 정보도 장비와 사용자 모두 실시간 감시와 통제를 받아 노출로부터 보호되어야 한다. 이는 내부자 공모 유출을 막기 위한 것이고 이런 정도의 보호 수준을 만족시킬 때 비로소 기업 비밀정보는 보호된다고 정의할 수 있다 [14,15].

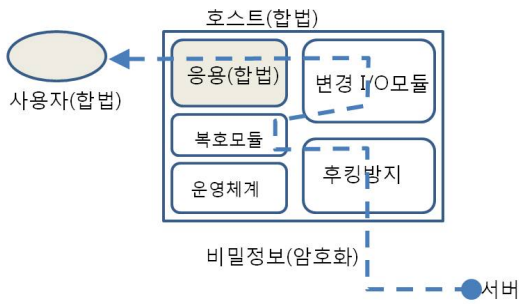
따라서 기업 비밀정보 보호를 위한 보안조건은 저작물 차원과는 비교할 수 없는 강력한 보안조치를 필요로 한다. 특히 기업 비밀정보는 내부자 공모 유출에 매우 취약한 것으로 조사되었다[1,21]. 첫 번째 기본적 보안조건은 허가된 사용자가 제한적으로 허락된 호스트들에서만 정보 접근이 가능하고, 사용자와 호스트는 각각 접근 권한이 따로 설정되어 있어야 한다. 이는 특정 호스트들로 구성된 그룹과 멀티캐스트 보안 전송에 기반 하여 정보가 유통되어야 한다는 것을 의미하며 그 보안 수준은 그룹 키-관리를 통한 멀티캐스트 보안 통신 시스템에 기반 하여야 한다.

따라서 본 논문은 1:M, N:M 멀티캐스트를 허용하는 송신자(sender) 우선(initiated) 보안 그룹 통신 기반 표준 RFC 4046 MSEC (Multicast Security) GKMA (Group Key Management Architecture)을 연구하여 비밀정보 유통 프로토콜과 관련한 그룹 키 관리 프로토콜을 설계하는 것이 목표이다[3,4].

2. 기업 비밀정보 보호〔21〕

기업 비밀정보의 흐름을 표현한 그림 1을 보면 중요 비밀정보는 기밀성이 유지된 상태에서만 전송되고, 사용자와 호스트(장치)에 대한 인증(authentication)과 두 개체의 권한 (authority)이 모두 합당한 경우에만 접근이 가능하다. 이런 이중 인증/권한 검사는 정보의 유출 가능성을 완전히 배제하는 보안 조건이다.

그림 1은 비밀정보의 정상적 흐름에 관여하는 개체는 사용자, 호스트, 응용 프로그램이고 정보의 복호화 과정은 사용자가 관여하지 않고 진행



(그림 1) 비밀정보 유통경로

되는 투명성 (transparency)을 보장하여 내부자 공모 유출 시도를 막는다. 여기서 사용자는 정보의 내용을 응용 프로그램을 통해서만 관찰하거나 편집할 수 있고 사용자가 직접 복사, 인쇄, 저장할 수 있는 방법을 제거한다.

응용 프로그램은 변경된 입출력 모듈을 통해서 비밀정보를 서버에게 요청하고 동시에 복호모듈이 키-서버에게 해당 복호키를 요청하여 복호 모듈이 복호화 후 평문 정보를 응용 프로그램에 전달하여 처리 요구를 수행하게 된다.

이 때 해커는 후킹(hooking)기법을 통해 커널이나 복호모듈 내부로부터 비밀정보 해킹시도가 가능하므로 모든 후킹방지 조치가 요구된다. 특히 운영체계 차원의 커널후킹 및 I/O API 후킹을 포함한 모든 후킹이 통제되어야 안전성이 보장되며, 불법적 화면인쇄도 막을 수 있다. 후킹방지는 본 논문의 연구범위를 벗어나므로 후킹에 의한 해킹은 없다고 가정한다.

그룹 내 유통에서 불법 유출을 방지하기 위한 멀티캐스트 보안 그룹 프로토콜의 설계를 위해 암호화 및 키-관리와 관련된 보안조건을 논의해 보자.

2.1 보안 조건

2.1.1 공간적 시간적 제약

기업 비밀정보는 보호 등급에 따라 사용자, 호스트, 응용 프로그램 등 각 개체 마다 공간적 및 시간적 접근 제한을 둔다. 공간적 제약은 유통망의 구조적 측면에서 보면 그룹 내 폐쇄된 통신을 기반으로 해야 하는 것을 의미이다(그림 2 참조).

이는 호스트의 위치 또는 사용자의 등급에 따라서 접근 권한이 달라진다. 예를 들어 CEO나 CTO 사무실에 있는 호스트는 비밀정보에 대한 모든 권한이 있어야 하지만 일반 사무실에 위치한 호스트에서는 업무와 관련한 권한만을 부여해야 한다. 사용자의 경우도 고위직 임원과 일반 직원의 접근 권한이 당연히 달라야 할 것이다. 응용 프로그램의 경우도 등록 프로그램이면 해당 정보를 처리할 수 있지만 그렇지 않은 사용자 설치 프로그램으로는 비밀정보에 접근은 불가능해야 한다.

또한 이런 개체들은 시간에 따라서 권한이 변화되는 시간적 제약을 둔다. 이는 그룹이라는 통제된 공간이지만 항상 그룹 내 움직임을 관찰, 감시, 추적해야 한다는 뜻이다. 호스트는 교체, 이동, 폐기에 따라서 권한은 변경되고, 사용자는 이직, 퇴직, 보직 변경, 승진에 따라서 권한이 변경된다.

2.1.2 이중 인증/권한 검증

그러므로 사용자와 호스트, 그리고 응용 프로그램 셋 모두 합법적 권한이 있는 지를 기업 비밀정보 접근 요구마다 항상 검증하게 되며 특별히 사용자와 호스트에 대한 검증은 명시적으로 이루어지며 이를 이중 인증/권한 검증 (authentication/authority)이라고 한다. 권한 검증은 정책서버에서 수행하는 것으로 본 논문에서는 상세히 다루지 않는다.

응용 프로그램은 입출력 모듈이 적법하게 변경/설치되어 있어야 정보처리가 가능하므로 묵시적 인증이라고 할 수 있다. 이와 같은 합법적 개체 인증은 사용자-호스트-응용프로그램의 삼중(triple) 인증/권한 검증으로 간주될 수 있다. 이것은 매우 강력한 보안조건으로서 해커가 합법적 권한을 도용하지 않는다면 접근은 현실적으로 불가능하다.

역으로 서버에서 호스트에 보낸 정보는 서버의

합법성을 검증하되, 서버 자체에 대한 검증과 개체가 속한 합법적 그룹인지를 검증하는 것이 필요하다.

2.1.3 암호화 유통 원칙과 키와 정보 분리 원칙

일반적으로 정보가 노출되는 주요 원인은 정보가 평문 형태 존재하거나, 암호문과 복호키가 함께 같은 장소에 저장되어 유출되기 때문이다. 저작물 보호에서 일회 검증 후 항상 평문 형태로 존재하는 것은 정보 노출을 막을 수 없다. 그러나 본 연구의 비밀정보는 오직 암호화 형태로만 존재하고 복호화는 복호모듈에서만 이루어지므로 해당 응용 프로그램이 없다면 정보 내용에 접근할 수 없다.

따라서 본 시스템은 복호키 없이 암호문 형태로만 존재하고, 정보와 키가 동시에 존재하지 않으므로 노출 가능성의 원인을 제거한다.

그러므로 본 논문은 완전한 기밀성을 위한 두 가지 중요한 보안조건으로 이중키 방식에서처럼 개인키의 유통을 금지하고 안전한 곳에 보관하여 기밀성(credentiality)을 높이므로 비밀정보는 보관과 유통에서 암호화 상태로만 두는 '암호화 유통 원칙'과 복호키는 비밀정보와 분리하여 저장한다는 '키와 정보 분리원칙'을 준수한다.

2.1.4 키 및 정보 일회성 원칙

복호키는 오직 합법적 응용 프로그램의 요청에 의해서만 제공되며, 복호키는 사용 후 호스트의 메모리에 기록을 삭제해야 하고, 동시에 응용 프로그램도 처리 도중 일시 저장되는 평문 형태의 어떤 사본도 남지 않도록 삭제하는 '키 및 정보 일회성 원칙'을 복호모듈에 구현한다.

본 논문은 2.1절에서 언급한 강력한 세 가지 기밀성 보안조건을 채택한다. 요약하면 모든 비밀정보의 유출방지를 위해서 정보는 암호화 상태로만 오직 존재할 수 있다는 암호화 유통원칙과 복호키는 응용 프로그램을 통해서만 요청되고 키와 정보는 분리되는 원칙, 해당 처리 후 모든 복호키와 임시 파일은 완전히 삭제하는 일회성 원칙을 적용하여 노출 가능성을 배제한다.

2.5 내부자 공모 방지

위 강력한 보안조건을 준수하면 모든 정보 유출경로를 통제할 수 있지만 마지막 남은 가능한 유출경로는 내부자가 호스트 또는 그 저장매체를 허가 없이 몰래 탈취하여 반출한 후 해외로 불법 유출하는 경로이고 이는 매우 빈번하게 발생하는 불법유출 사례이다[2]. 따라서 비밀정보 유통 그룹 내의 모든 자원들은 네트워크 관리대상 개체(entity)로 등록하고 설정하여 동작 상태 변화를 실시간으로 감시하고 관리하여야 한다.

관리대상 자원은 데스크톱과 같은 고정 호스트와 그 저장매체, 노트북이나 PDA와 같은 이동 호스트와 그 저장매체, 플래시(flash) 메모리와 같은 이동 저장매체 등이다.

위와 같은 자원들의 상태 관리를 위한 네트워크 관리 프로토콜(예: snmp)의 동작을 보면 그룹 내 호스트에 고객 에이전트는 자신의 자원들의 가동 상태 변화를 서버에 있는 관리 에이전트에게 정기적으로 보고하거나, 트랩(trap) 설정을 통해서 특이사항 발생 즉시 경보를 발행하면 관리자는 필요한 조치를 취하는 것이다.

먼저 해당 장치들의 탈퇴를 시행하고, 노출이 의심되는 정보교환(Data) SA(security association)나 키-변경(rekey) SA의 키-정보(그룹 암호키, 암호키, 인증키) 변경 절차를 수행하고 만일 도용 장치에 비밀정보가 있었다면 해당 암호키는 변경하고 다시 암호화하고 저장한다.

업무상 비밀정보를 외부로 반출해야 하는 경우 내부자 유출 시도가 빈번하므로 적법한 응용 프로그램이 설치된 호스트와 이동 저장매체에 비밀정보를 저장하여 반출한다. 이 때 암호키의 노출을 막기 위해서 이동 저장매체에 저장할 비밀정보는 새로운 세션키(일회용)로 다시 암호화하여 인증 정보와 함께 이동 저장매체에 담는다. 이는

비밀정보와 장치를 분리함으로서 동시에 유출될 확률을 낮추려는 노력이다.

2.6 설계 동기와 범위

선행연구를 통하여 유통 시스템은 어떤 구조로 구성되어야 하며, 유출의 원인 분석을 통해서 보안구멍의 위치와 유통망 구조를 만들어 주는 프로토콜의 개략적 동작 구조를 연구하였다[21]. 그러나 본 연구에서는 MSEC을 본 유통 프로토콜의 설계의 모체로 활용하여 Data SA와 Rekey SA를 확립하는 단계를 프로토콜로 확정하였다.

기업 비밀정보 보호 시스템은 위 2.1절의 강력한 보안조건을 준수하면서, 프로토콜의 복잡성을 숨기는 처리 투명성(transparency)과 수행속도가 느려지지 않도록 하는 까다로운 조건들을 모두 만족하는 프로토콜을 설계한다는 것은 상당한 도전이 되었다.

2.6.1 수행속도 고려

비밀정보 유출을 시도하는 외부 해커를 봉쇄하는 것과 동시에 내부자 공모로부터 보호한다는 두 상반된 측면의 보안조건을 모두 만족하는 보안기법은 복잡성으로 인한 당연히 수행속도는 느릴 것으로 생각했지만, 성능 분석은 특이하게 첫 보호조건 암호화 유통원칙이 프로토콜의 수행 부담을 오히려 줄인 것으로 보여준다. 이는 기존 보안기법에서 사용하는 암호기법들 세션키 생성, 전자봉투, 해쉬함수, 전자서명 등을 줄일 수 있었기 때문이었고, 그 결과는 정보유통이 증가할수록 수행속도에서 암호화 유통의 부담 비율은 줄어들었지만 보안성은 여전히 유지되었다.

새로운 입출력 모듈은 복호화 함수 호출을 추가하여 복호화와 인증 및 권한 검증까지 추가 모듈로 내부적으로 투명하게 수행 가능하고, 기존 응용 프로그램을 사용할 때와 비교해서 투명성뿐만 아니라 수행속도까지 향상되는 효과를 보았다. 처리량이 증가할수록 성능이 높아지는 것을 6장에서 보여준다.

2.6.2 저작물 측면 문제

기업 비밀정보 보호를 지적재산 저작물 측면에서 다루는 것은 위험한 보안구멍(security hole)을 방치하는 것과 같다. 저작물 관리 DRM의 보안구멍은 공개된 인터넷에 정보를 유통시키면서 보호책임을 구매자에게 전달하는 것까지로 제한한 것이다. 이용하고 있는 정보에 대한 감시와 보호가 DRM에서는 없다는 것이다.

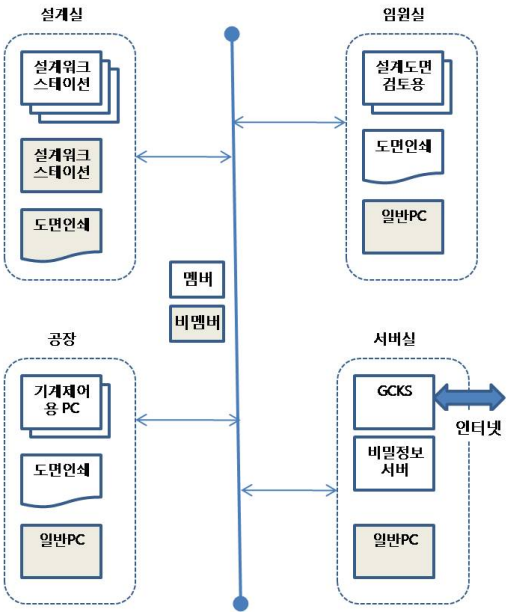
기업 비밀정보 보호의 특징은 모든 접근 자를 잠재적 해커로 간주해야 하고 정보의 생명주기동안 보호 대상으로 관리해야 하며, 내부자 공모 유출 시도까지 색출할 수 있는 강력한 합법성 검사가 필요로 한다[9,10,11,21].

현재 저작물 차원의 DRM 기법을 사용하는 이유는 MS사와 같은 유명기업에서 제공하는 DRM 모듈을 사용하는 것이 상업적으로 안정되기 때문이다. 본 연구는 같은 근거로 검증된 프로토콜들을 채택하여 안정성을 유지하는 것이 필요하다고 판단하고 기존 표준 MSEC에 기반을 두었다.

2.7 그룹 통신 기반 유통망

DRM 기반 저작물은 인터넷 광역망에서 유통되는 것을 전제로 하는 반면 기업 비밀정보 유통은 통제된 사용자 그룹과 통제된 장소에 등록된 호스트의 그룹 사이에서 폐쇄(closed)적 이루어져야 한다. 내부자 공모 유출을 방지하기 위해서 [11,12,13] 본 유통망은 전형적인 그룹 통신망으로 구성되어야 하고, 그룹 멤버관리, 그룹 보안 키-관리, 그리고 네트워크 장비관리가 핵심이다(그림 2 참조).

본 비밀정보 유통망에서 정보 전달 목적지는 그룹 멤버들이므로 전형적인 그룹 통신이고 그룹 멤버관리도 전형적인 프로토콜 IGMP(Internet Group Management Protocol)로 충분하다(3.1절 참조). 단지 기존 IGMP의 보안방식은 보호수준이



(그림 2) 그룹 유통망 사례 [21]

낮으므로 그룹 통신 보안에 필수적 키-관리는 그룹 보안 키-관리 구조 MSEC GKMA (Group Key Management Architecture)에 근거한다(3.2절 참조). 좀 더 구체적인 논의는 3장에서 이루어진다[3,4, 8,20].

장비의 반출 여부 검사는 SNMP(Simple Network Management Protocol) 네트워크 관리 프로토콜의 목적과 완전히 일치함으로 본 시스템에 그대로 적용한다고 가정한다(SNMP 요약은 3.3절 참조)[7].

3. 관련 표준 프로토콜 적용

본 연구의 비밀정보 유통망(그림 2)에서 정보 유통은 등록 호스트만으로 구성된 그룹 안으로 제한하여 노출로부터 보호하므로 멀티캐스트 기반 보안(MSEC) 프로토콜을 적용하여 이를 달성하는 프로토콜 설계가 핵심이다.

구체적으로 프로토콜 설계의 주요 논점은 첫째 그룹으로(부터) 멤버의 정상적 가입과 탈퇴를 목적으로 IGMP join/release 수행할 때 MSEC 기반

(표 1) 각 프로토콜 기능 단위별 역할

개체 / 프로토콜	연구	IGMP	MSEC GKMP	SNMP
GC(Group Control)	GCKS	그룹 멤버 관리		관리자
KC(Key Control)			SA 키 생성/등록/삭제/변경	
비밀정보서버	파일서버			
호스트	정보처리 대상	가입/탈퇴 요청	키-정보 수신/보관/삭제/변경	보고자

등록/해제(registration/de-registration)을 각각 함께 수행하여 Data SA와 Rekey SA 키-정보를 생성하고 분배하며 또 해제 때는 키-변경(rekey)하는 프로토콜의 설계이다. 또 그룹 멤버에게 이상이 발생한 경보를 발행할 때 강제 탈퇴의 IGMP release와 함께 MSEC 해제(De-registration) 과정도 수행하고 키-변경도 하는 프로토콜의 설계이다.

둘째 그룹 내에서 안전한 정보유통을 위한 기본적 네 가지 보안기법, 기밀성 (confidentiality), 무결성(integrity), 인증성 (authentication), 부인봉쇄(non-repudiation)를 위해 필요한 암호기법 선정과 연관된 키-정보의 생성, 분배, 그리고 변경하는 그룹 키-관리 프로토콜 설계이다.

셋째 비밀정보 유통망의 실시간 자원 상태 관리는 SNMP를 통해서 수행한다. SNMP의 기능 변경 없이 적용이 가능하므로 각 적용 프로토콜의 기능 요소들 사이의 매핑만으로 설계를 대신한다 (표 1. 참조).

3.1 IGMP 적용 [6]

핵심 기능을 요약하면 그룹 멤버와 멀티캐스트 라우터 사이에 그룹 멤버관리에 관련 정보 송수신 절차를 규정한 프로토콜이다. 특별히 멀티캐스트 라우터는 특정 멀티캐스트 주소에 가입된 호스트들이 자신의 네트워크에 존재하는지 확인하

고 동일 그룹 내 호스트들의 정보를 담은 호스트 리스트는 상태 정보도 함께 담고 있다. 주기적으로 '일반(general) 질의(query)'를 보내어 호스트 리스트 정보를 갱신할 호스트 상태 정보를 보고 받는다. 각 멤버는 인터페이스의 변화가 발생하면 라우터에 보고하여 호스트 리스트 내의 상태 정보를 갱신한다.

IGMP의 주 프로토콜은 그룹 멤버 호스트의 가입/탈퇴 과정을 수행하는 것이다. 그러나 MSEC의 Data SA와 Rekey SA의 키-정보를 각 호스트에 생성, 분배, 변경 하도록 MSEC의 등록/해제 프로토콜의 수행을 촉발한다. 특별히 가입 과정은 GKEK 생성을 수반하고, 탈퇴 과정은 그룹 전체 호스트에 현재 GTEK 변경과 다음 변경에 사용할 GKEK 변경을 수반한다. 멤버 관리는 IGMP가 수행하고 키-정보의 생성, 분배, 변경은 MSEC 프로토콜의 등록, 해제, 변경 프로토콜이 수행한다[20].

표 1을 보면 그룹 제어 키 관리 서버 GCKS는 IGMP 멀티캐스트 라우터의 역할과 동일하며, 등록된 호스트가 그룹 내에 존재하는지 확인하는 과정을 수행하면서 IGMP 질의 및 보고 메시지를 GTEK 키-정보로 암호화 하고, 인증 과정에서 GCKS 서버는 해당 멤버 호스트를, 호스트는 서버를 인증/검증한다.

GCKS 서버가 호스트의 그룹 가입 메시지를 받으면, 안전통로(secure channel)를 개설하여, 그룹 서버가 모든 키-정보를 생성한 후 서버에 대한 인증 키-정보는 호스트로 전송하고, 호스트 인증 키-정보는 서버에 보관한다. 또 그룹 탈퇴 메시지를 받으면 호스트의 모든 그룹 키-정보를 삭제하고, 새로운 키-정보를 생성하여 모든 호스트를 변경한다.

결론적으로 IGMP는 GKMP와 아무런 충돌 없이 상호 연동이 가능하고 기존 IGMP의 그룹키를 사용하지 않고 GKMP의 키-정보를 그대로 접목시킬 수 있다.

3.2 MSEC 적용 〔3,4,20〕

MSEC 그룹 키-관리 프로토콜이란 보안 그룹 안에 가입된 멤버들 사이 통신을 보호하기 위한 그룹 관련 키-정보의 관리가 목적이다[20].

MSEC은 그룹 관련 키-정보는 오직 그룹 가입 멤버들만이 접근할 수 있도록 제한한다. 합법적 그룹 멤버에게 기밀성과 인증성 보장을 위한 암호기법에 관련한 키-정보를 생성하고 분배하고 변경하는 프로토콜이다.

MSEC(Multicast Security) 프로토콜은 키-정보의 등록(registration)과 해제(de-registration) 그리고 변경(rekey) 프로토콜과 키-정보의 이용(data) 프로토콜로 구성된다.

키-정보의 등록에서는 Data SA(security association)의 GTEK와 Rekey SA의 GKEK를 서버가 생성하여 호스트에 전달한다. 키-정보 해제는 모든 SA의 키-정보를 제거하고 다른 키-정보 모두 변경하는 프로토콜이다.

키-변경 프로토콜은 이전 Rekey SA의 키-정보를 이용해서 새로 생성한 GTEK와 GKEK를 분배한다.

본 정보 유통 프로토콜 설계에서 불필요한 복잡성을 줄이기 위해 다음과 같이 가정한다.

MSEC는 DoS 공격 및 재생(replay) 공격으로부터 안전하다.

키-변경은 호스트 독립 키-정보와 종속 키-정보를 따로 처리한다. GCKS와 멤버 사이에 호스트 독립 키-정보는 유니캐스트(unicast)가 아닌 멀티캐스트로 전송하고 유통 그룹 내 확장성(scalability) 문제는 고려하지 않는다.

MSEC에서 초기 안전채널로 IPSec AH, ESP 등을 사용할 수 있고 안전채널은 인증기법, 권한 검사, 전송방식에 따라 달라지지 않는다.

탈퇴 멤버 호스트와 멤버 아닌 호스트에게 키-정보는 노출되지 않는다. 그 보관 장소는 안전하고 특별히 새로 생성된 Rekey SA는 전방기밀성(Forward Secrecy)와 후방기밀성(Backward Secrecy) 모두 만족한다.

MSEC은 기존 PKI (public key infrastructure) 인증과 호환되며 NAT (network address translation) 기법에 영향을 받지 않는다.

이상에서 언급한 MSEC GKMP의 단순화 가정은 IGMP, MSEC GKMP, SNMP의 기능 단위 요소가 명확하게 일치하여 기능 요소를 통합하는 것이 그림 2의 구조 변경 없이도 상호 연동이 가능하다.

3.3 SNMP 적용 [7]

네트워크 관리 프로토콜 SNMP는 관리 대상으로 호스트와 그 자원들, 네트워크 장치 등 네트워크 내 장치에 에이전트를 설치하여 관리자(manager)와 보고자(client)로 나누고, 보고자는 장치의 상태 정보와 이상 발생을 관리자에게 보고하고, 관리자는 관리 설정정보를 보고자에게 응답하는 프로토콜이다.

네트워크 관리 대상 장치의 하드웨어와 소프트웨어, 프로토콜, 인터페이스, 각종 컴퓨터 부속품 등 모든 관리 대상을 표준 식별자 OID(Object ID)로 구분하여 인터넷 표준 관리정보로 데이터베이스화 한 MIB(관리정보 베이스)에 관리 대상의 ID, 장치 관련 정보와 관리 특성을 기록할 수 있고, 관리자와 보고자 에이전트는 주기적 또는 이상 발생 시에 관리 정보를 주고받을 수 있다. 그 우수한 기능성에 비해 부담이 작고 단순하여 네트워크 관리를 쉽게 해주므로 본 비밀정보 유통 시스템에는 성능 부담이 없이 효과적이다.

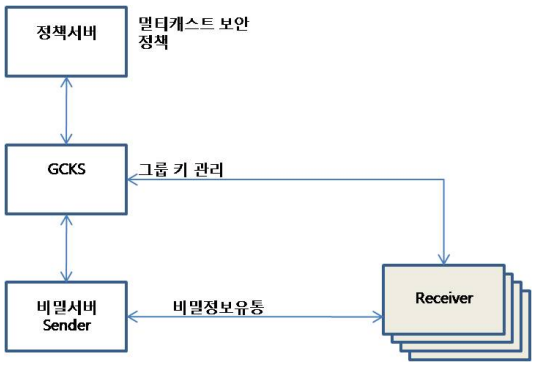
또한 관리정보 데이터베이스(MIB)에는 관리 대상에서 문제가 발생하면 경보를 발행하도록 하는 트랩(trap)을 정의하는 OID 식별자도 있어서, 비밀정보 유통 프로토콜의 실시간 키-변경에 방아쇠 역할까지 수행하고, SNMP 뿐 아니라 IGMP 질의도 그룹 내 호스트의 존재 여부를 알려주고, 각 호스트의 내부 자원으로 CPU, 하드디스크, 네트워크 카드, 인터페이스, 포트, 응용 프로그램 등 모든 자원의 작동 상태를 실시간으로 감시할 수 있으므로 자원의 유출된 후 감지될 때까지 무방비의 키-정보 노출의 가능 시간을 제로(0)로 만들어주므로 전방/후방 기밀성 보장에 도움이 된다.

SNMP 적용의 가장 큰 혜택은 비밀정보의 유출 위험이 큰 자원, 호스트, 하드디스크 그리고 이동 저장매체의 절취 행위를 실시간으로 탐색할 수 있다는 것이다.

4. 비밀정보 유통 프로토콜

비밀정보 유통과정은 크게 세(3) 과정으로 구분할 수 있다. 첫째 호스트가 비밀정보파일 생성하고 파일서버에 등록하는 과정, 둘째 사용자가 호스트에서 응용 프로그램을 통해 비밀정보파일을 요청하면 파일서버가 전송하고 응용 프로그램이 처리하는 과정, 셋째 비밀정보파일을 외부로 반출하려고 새로운 세션키 생성하여 재-암호화 하고 이동 저장매체로 복사하고 그 매체를 인증 등록하는 반출 처리 과정이다. 이 세 과정 프로토콜은 4.2절에서 기술하며 비밀정보의 생명주기 관점에 주목한다.

그러나 비밀정보 유통 프로토콜의 보안성 보장은 키-관리에 있고 크게 네 단계로 구분한다. 첫째 비밀정보 유통 그룹 초기 서버 설정 단계로서 서버와 통신에 사용할 Data SA와 Rekey SA 의 키-정보를 생성하고 호스트로 분배할 준비를 하는 단계이다. 둘째 등록 호스트의 그룹 멤버 가입 단계로서 준비된 Data SA와 Rekey SA의 키-정보를



(그림 3) 비밀유통 기능 구조도

호스트로 분배하여 호스트가 통신할 준비를 갖춘다. 셋째 비밀정보 유통 보호 단계로서 비밀정보 파일을 암호화 상태로 전송하고, 복호키를 그룹 키-정보로 암호화하고 인증 정보를 붙여서 안전한 통신을 수행한다. 넷째 단계는 호스트의 가입 또는 탈퇴, 자원 이상 발생 때에 키-변경 과정을 수행하는 단계이다. 이 단계의 구체적 프로토콜 시간 순서는 5장에서 기술하였다.

4.1 암호기법 용어 정의

본 절에서는 프로토콜 기술에 사용될 수학적 기호를 설명한다. 해당 절에서 그 의미와 역할에 대해서 더 자세히 설명할 것이다.

4.1.1 암호함수

fid: 정보 식별자로 비밀정보파일을 식별하는 ID이다.

(표 2) 키-정보

구분	키 종류	역할
GKCS 인증	S_{ks}, V_{ks}	인증, 암호
파일서버 인증	S_{fs}, V_{fs}	인증, 암호
그룹 인증	S_g, V_g	그룹 인증, 서버가 해당 그룹 인지 여부
사용자 인증	S_{uid}, V_{uid}	사용자 인증, 암호
호스트 인증	S_{did}, V_{did}	호스트 인증, 암호
저장매체 인증	S_{sid}, V_{sid}	매체 인증, 암호
키-변경 암호키	K_g	키-변경 정보 암호
비밀정보 암호키	K_{fid}	비밀정보 암호
비밀정보 세션키	Σ_{fid}	임시 암호키
호스트-서버 인증	$E_{h\to s}()$	$V_{fs}V_gS_{did}S_{uid}()$
서버-호스트 인증	$E_{s\to h}()$	$V_{did}V_{uid}S_{fs}S_g()$
파일-키서버 인증	$E_{ks\to fs}()$	$V_{fs}V_gS_{ks}()$
키서버-파일 인증	$E_{fs\to ks}()$	$V_{ks}V_gS_{fs}()$
호스트-서버 검증	$D_{h\to s}()$	$V_{fs}V_gS_{did}S_{uid}()$
서버-호스트 검증	$D_{s\to h}()$	$V_{did}V_{uid}S_{fs}S_g()$
파일-키서버 검증	$D_{ks\to fs}()$	$V_{ks}S_gS_{fs}()$
키서버-파일 검증	$D_{fs\to ks}()$	$V_{fs}S_gS_{ks}()$

$K_{fid}()$: 식별자 fid 파일의 암호키와 알고리즘을 포함한 대칭 암호기법을 의미한다. 대칭 암호/복호를 기술할 때 알고리즘과 키(key)를 분리하여 표현하는 기존 방식은 암호기법의 특징을 표현하지 못하므로 두 측면을 모두 포함하는 함수형식을 사용한다. 메시지 M의 암호화(키 중심표현 $E_k(M)$)는 $K_{fid}(M)$으로, 복호화(키 중심표현 $D_k(M)$)는 역함수 형식 $K_{fid}^{-1}(M)$으로 표현되고 관계 $K_{fid}^{-1}K_{fid}(M)\to I_{fid}(M)$이 성립하고, 여기서 $I_{fid}()$ $identity$ 함수이고 $I_{fid}(M)\to M$이다.

인증 $S_{id}()$, 검증 $V_{id}()$: 개체 식별자를 id라고 하면, 이중키 암호기법에서 개인키 함수 [19]는 $S_{id}()$로, 공개키 함수는 $V_{id}()$로 표현하고, 관계 $V_{id}S_{id}=S_{id}V_{id}=I$ 성립한다. 이 방식은 암호키와 알고리즘을 모두 포함한 암호함수는 암호기법 DSA(digital signature) 관점을 사용하여 표현된다 [19]. 각 개체(식별자=id)는 (S_{id}, V_{id}) 이중키 쌍을 가진다. $S_{id}()$는 인증성에서는 사인(sign)으로, 기밀성에서는 복호화를 수행하고, $V_{id}()$는 인증성에서는 검증(verify)으로, 기밀성에서는 암호화를 수행한다. 이중키의 수행시간이 매우 크므로 크기가 큰 비밀정보파일 암호화에는 사용하지 않는다.

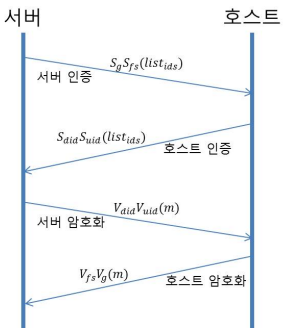
4.1.2 식별자 리스트와 암호기법

식별자 리스트 $list_{ids}$는 그룹 내 여러 개체들이 합력하여 하나의 처리를 가능하게 하는 적합성을 가진다. 예를 들면 적법한 사용자, 호스트, 응용프로그램, 비밀정보파일이 하나로 모이면 처리에 적법한 권한을 가진다고 정의한다(2장 참조). 따라서 식별자 리스트에 포함된 정보는 비밀정보(식별자=fid)와 그 암호함수 $K_{fid}()$, 이동 저장매체(식별자=sid)와 그 검증함수 $V_{sid}()$, 호스트(식별자=did)와 그 검증함수 $V_{did}()$, 그리고 사용자(식별자=uid)와 그 검증함수 $V_{uid}()$으로 구성된다.

인증성에서 이중 인증/검증 원칙은 서버는 호스트와 사용자를 동시 검증하고 호스트는 서버와

그룹을 동시 검증하는 것을 의미한다. 여기서 그룹은 타 그룹에 속한 서버로 해킹하는 것을 방지하기 위한 서버의 이중 인증이다.

사실 공개키로서 검증함수 $V_{id}()$은 공개되어도 무관하므로 식별자 리스트가 해킹으로 노출되어도 문제가 되지 않지만 식별자 정보를 도용할 가능성이 있으므로 이를 방지하기 위해서 검증함수로 암호화하여 기밀성을 유지한다. 즉 기밀성은 호스트에서 서버로 보낼 때는 서버와 그룹 공개키 함수로 암호화 하고, 서버에서 호스트로 보낼 때는 호스트와 사용자 공개키 함수로 암호화 한다.



(그림 4) 인증과 암호화

여기서 모든 식별자 리스트는 호스트에서 파일서버로 전송되는 경우 $V_{fs} V_g S_{did} S_{uid}(list_{ids}^{0011_{(2)}})$을, 파일서버에서 호스트로 전송되는 경우는 $V_{did} V_{uid} S_{fs} S_g(list_{ids}^{0011_{(2)}})$로 이중 인증성과 기밀성을 보장한다. 여기서 $E_{h\to s}()$는 $V_{fs} V_g S_{did} S_{uid}()$을 의미하고 $E_{s\to h}()$는 그 역과정인 $V_{did} V_{uid} S_{fs} S_g()$로 표기할 것이다. 또 서버 간 통신에서 키 서버에서 파일서버로 전송되는 경우 $E_{ks\to fs}()$은 $V_{fs} V_g S_{ks}()$로, 파일서버에서 키 서버로 전송은 $E_{fs\to ks}()$은 $V_{ks} V_g S_{fs}()$로 인증과 암호화를 한다.

따라서 복호와 검증 $V_{fs} V_g S_{did} S_{uid}()$은 $D_{h\to s}()$로, $V_{did} V_{uid} S_{fs} S_g()$은 $D_{s\to h}()$로, $V_{ks} S_g S_{fs}()$은 $D_{ks\to fs}()$로, $V_{fs} S_g S_{ks}()$는 $D_{fs\to ks}()$로 표기한다.

비대칭 공개키 함수를 식별자와 함께 전송하는 이유는 파일서버 이외의 호스트가 파일을 분산 보관할 경우 비밀정보파일은 호스트 간 요청과 전송이 될 때 요청 호스트는 자신의 인증서를 주는 것이 일반적이지만 본 시스템에서는 식별자 리스트 안에 공개키 함수를 두면 이중키 방식의 최대 단점인 많은 키-쌍 유지 부담을 줄이려는 것이다.

그러나 본 논문에서는 프로토콜의 복잡도를 줄이기 위해서 모든 정보파일은 파일서버에만 있는 것으로 가정하여 구조를 설계하였다. 따라서 호스트 사이에 정보교환은 불필요하므로 식별자 리스트의 암호함수는 일부만 활용된다.

식별자 리스트의 길이는 암호화 수행시간과 관련이 있어서 사용되지 않을 정보는 포함시키지 않고 리스트 만든다. 따라서 $list_{ids}^{i_{(2)}}$의 상위 첨자 이진수 $i_{(2)}$는 암호함수 정보 포함된 여부를 지칭한다. 예를 들면 $i_{(2)} = 1101_{(2)}$ 이면 포함된 암호함수는 (K_{fid} V_{sid} ϕ V_{uid}) 으로서 포함된 것은 파일 암호기법, 저장매체 검증함수, 마지막 사용자 검증함수이고 호스트 검증함수는 생략했다는 것을 의미한다(ϕ 는 생략을 의미). 리스트 내에 특정 식별자(id)도 필요 없는 경우 $id \leftarrow none$으로 특정 식별자 생성을 요청하는 경우 $id \leftarrow req$ 이다.

4.1.3 키-변경 및 요청 메시지

키-변경은 시간에 따라 GTEK와 GKEK이 변한다는 의미이며 이러한 키-정보의 교환은 GKEK로 암호화되어 전송되므로 따라서 사용가능 시간을 의미하는 t 라는 시간 구분자가 포함되어야 하고, 예를 들면 $V_{uid}^{t}()$는 시간 t 때에 사용 가능한 사용자(식별자=uid)의 공개키 암호함수를 의미한다. 시간 정보는 상위 첨자로 표시하며 $V_{uid}^{t+1}()$은 $V_{uid}^{t}()$의 다음에 사용될 키-정보를 의미한다. 그러나 본 논문에서는 시간 첨자가 없으면 현재 키-정보를 의미한다.

그룹 내에서 정보 및 키-관리 요구는 메시지 전달방식으로 이루어진다. 여기서 메시지 구성은

$M_{요청구분자}$ = (요청구분코드, 식별자리스트) 형태이고 식별자 리스트는 인증과 암호화 된 형태로 보낸다. 요청구분코드는 다음 표와 같다.

(표 3) 처리요청 코드

요청구분코드	요청 내용
create	파일 식별자 및 암호키 생성 요청
encrypt	정보 암호화 요청
store	파일 저장 요청 (파일서버에게 보냄)
data-ren	정보/복호키 요청 (파일서버에게 보냄)
register	사용자 등록 요청
user	사용자 등록 후 식별자 통보
carry	정보 반출 요청 (키 서버에게 보냄)
data-pass	정보 전달 요청 (파일서버에게 보냄)
data	정보 전송
key-create	키 생성 요청 (키 서버에게 보냄)
key-pass	키 전달 (키 서버에게 보냄)
key	키 전송 (키 서버가 보냄)
key-req	키 검색 발췌 전송을 요청
rekey	키 변경 인증서 리스트 전송
private	인증 개인키 전달 요청

4.2 비밀정보 유통과정 3단계〔21〕

기존 인증방식은 메시지에 해쉬(hash) 함수를 적용하여 메시지 다이제스트(digest)를 생성하여 인증 함수로 다이제스트를 암호화 한 것이 사인(sign)이고, 검증 함수로 사인(sign)된 다이제스트 복호화가 가능하면 검증이 된 것이다.

그러나 비밀정보 유통과정에서 그룹 내 개체들의 인증은 다이제스트(digest)를 식별자 리스트로 대체한다(참조 4.1.3). 식별자 리스트는 의미상 '누가(who) 어디서(where) 무슨(how) 매체로 무엇(what)을 처리'라는 것으로 식별자 리스트의 각 요소는 (what, how, where, who) 벡터로 구성되고 모든 요소는 이 벡터 위에 인증 정보를 붙이고 모두 검증을 통과해야 요청이 처리된다.

비밀정보 처리과정은 하나의 요청에 대해서 한 번의 암호기법만을 적용되는 일회성 원칙 때문에 각 요청마다 응용 프로그램은 키-정보를 다시 요청해야 하고 매번 사용자와 호스트의 인증/권한 검증을 요구하므로, 어떤 요청에 대한 비밀정보 : 호스트 : 사용자 관계는 1:1:1 관계를 가지며, 반출 과정을 제외하면 호스트와 저장매체는 한 개체로 간주한다. 반출 요청은 이동 저장매체와 특정 호스트를 하나로 간주하는 인증 정보를 저장매체에 등록하고 그 호스트가 검증하는 방식을 적용하므로 1:1:1 관계는 여전히 성립한다. 그러므로 본 과정에서 검증은 이중 검증 원칙으로 충분하다.

식별자 리스트는 언급한 것처럼 ((비밀정보와 암호기법), (저장매체와 검증기법), (호스트와 검증기법), (사용자와 검증기법))와 같이 개체 정보를 담고 있다(자세한 것은 4.1.2를 참조할 것)

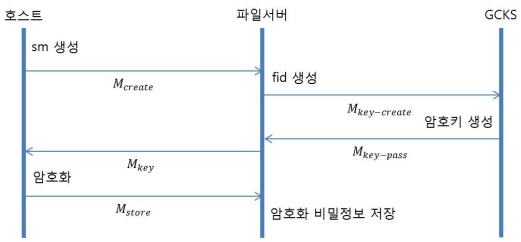
4.2.1 비밀정보 생성과 등록 과정

서버를 제외하고 비밀정보 생성이 가능한 호스트는 그 위치와 사용자 권한과 관련된다. 사용자는 호스트에 로그인하여 응용 프로그램을 통해서 비밀정보파일을 검색하여 특정 정보의 처리를 파일서버에게 요청한다. 생성 과정은, 예를 들면, 어떤 기업 설계실에서 중요한 도면설계가 완료되면 사용자가 비밀정보로 분류 요청하고, 비밀정보 파일서버는 식별자를 할당하고, 비밀정보에 대한 권한과 이용 정책을 결정하고, 비밀정보를 파일서버로 전송하면, 암호화 유통원칙으로 키-서버에게 암호키 생성을 요청하고 파일서버는 암호화하여 비밀정보파일을 식별자와 함께 저장하고, 암호키와 정책은 키-서버에 등록한다.

다음은 비밀정보의 생성/등록 과정 프로토콜의 시간흐름 차트 (time-sequence chart)이다.

① 호스트
- 비밀정보 sm 생성

② 호스트→파일서버
- 정보 식별자 생성을 요청하는 식별자 리스트 $list_{ids}^{0011_{(2)}}$ = $((req, req), (none, \varnothing), (did, V_{did}), (uid, V_{uid}))$ 생성.



(그림 5) 정보생성 등록

- $E_{h\to s}(list_{ids}^{0011_{(2)}})$ 생성
- 식별자와 암호키 생성 요청 메시지 M_{create} = $(create, E_{h\to s}(list_{ids}^{0011_{(2)}}))$ 생성. 파일서버로 전송.

③ 파일서버
- 복호와 검증 $D_{h\to s}(M_{create})$ 에서 (req, req) 발췌
- 정보 식별자 fid 생성 후 (fid, req) 구성.
- $list_{ids}^{0011_{(2)}}$ = ((fid, req) , $(none, \varnothing)$, (did, V_{did}) , (uid, V_{uid}))로 변경.

④ 파일서버→GCKS
- 암호키 생성 요청 메시지 $M_{key-create}$ = $(key-create, V_{ks}S_{fs}(list_{ids}^{0011_{(2)}}))$ 생성. GCKS로 전송.

⑤ GCKS
- 암호키 K_{fid} 생성. (fid, K_{fid}) 생성. 저장.
- $list_{ids}^{0011_{(2)}}$ 의 $(fid, req)\leftrightarrow(fid, K_{fid})$ 변경.
- 인증과 복호로 $V_{fs}S_{ks}(list_{ids}^{0011_{(2)}})$ 생성.

⑥ GCKS→파일서버
- 키 전달 메시지 $M_{key-pass}$ = $(key-pass, V_{fs}S_{ks}(list_{ids}^{0011_{(2)}}))$ 를 파일서버로 전송.

⑦ 파일서버→호스트
- $M_{key-pass}$ 에서 $V_{ks}S_{fs}()$ 로 $list_{ids}^{0011_{(2)}}$ 발췌
- 키 전송 메시지 M_{key} = $(key, E_{s\to h}(list_{ids}^{0011_{(2)}}))$ 생성. 호스트로 전송.

⑧ 호스트
- M_{key} 에서 $D_{s\to h}()$ 로 $list_{ids}^{0011_{(2)}}$ 발췌.
- (fid, K_{fid}) 에서 $K_{fid}()$ 로 비밀정보 sm_{fid} 을 암호화 $K_{fid}(sm_{fid})\to om_{fid}$ 생성.

⑨ 호스트→파일서버
- om_{fid} 과 $list_{ids}^{0011_{(2)}}$ 을 함께 파일서버 fs 에게 비밀정보 저장 요청 메시지 M_{store} =$(store, om_{fid}, E_{s\to h}(list_{ids}^{0011_{(2)}}))$ 를 파일서버로 전송.

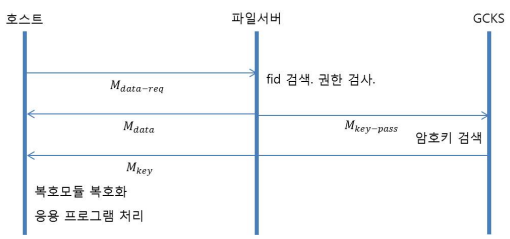
⑩ 파일서버
- (fid, om_{fid}) 를 저장. 등록.

4.2.2 비밀정보 요청과 처리 과정

위의 예를 사용하면 비밀 설계도면을 생산제어 컴퓨터에 프로그램으로 입력하기 위해 공장의 도면 캐드 컴퓨터에서 도면 검색기를 이용하여 설계도면의 다운로드를 요청하는 경우 다음과 같은 시간흐름 차트를 수행한다.

① 호스트→파일서버
- 식별자 리스트 $list_{ids}^{0011_{(2)}}$ = $((fid, \varnothing), (none, \varnothing), (did, V_{did}), (uid, V_{uid}))$ 생성.
- 이중(double) 사인(sign)과 서버 공개키로 암호화 $E_{h\to s}(list_{ids}^{0011_{(2)}})$ 생성
- 비밀정보 요청 메시지 $M_{data-req}$ = $(data-req, E_{h\to s}(list_{ids}^{0011_{(2)}}))$을 생성. 파일서버로 전송.

② 파일서버→GCKS, 호스트
- 복호와 검증 $D_{h\to s}()$ 으로 $list_{ids}^{0011_{(2)}}$ 발췌.
- fid 로 정보파일을 검색. did 와 uid 로 호스트와 사용자의 권한과 이용 정책 부합 검사.



(그림 6) 정보 요청 처리

om_{fid} 검색.

· 정보 전송 메시지 M_{data}=($data$, om_{fid}, $E_{s\to h}(list_{ids}^{0011_{(2)}})$) 생성. 호스트로 전송

· 키 전달 요청 메시지 $M_{key-pass}$ = ($key-pass$, $V_{gcks}S_{fs}(list_{ids}^{0011_{(2)}})$)를 GCKS로 전송.

③ GCKS→호스트

· $M_{key-pass}$ 에서 fid 를 발췌. 암호키 K_{fid}를 검색

· $list_{ids}^{0011_{(2)}}$에서 (fid, K_{fid}) 을 변경. $list_{ids}^{1011_{(2)}}$ 생성

· 암호와 인증으로 $E_{s\to h}(list_{ids}^{1011_{(2)}})$ 를 생성.

· 키 전송 메시지 M_{key} = (key, $E_{s\to h}(list_{ids}^{1011_{(2)}})$) 을 호스트에 전송.

④ 호스트

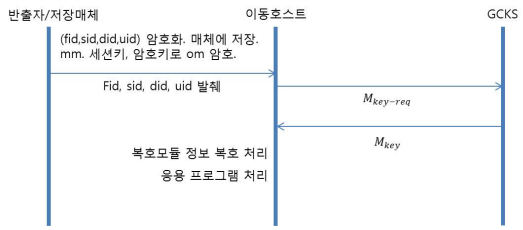
· M_{data}에서 비밀정보 om_{fid}을, M_{key}에서 K_{fid}을 얻어서 응용 프로그램의 복호모듈이 $K_{fid}^{-1}(om_{fid})$을 수행 처리.

4.2.3 비밀정보 반출 과정

비밀정보의 반출은 외부에서 이용할 목적으로 관리자의 허락을 받아 진행하는 과정이다. 노출로 인한 위험을 낮추는 조치로서 키와 정보 분리 원칙이 요구되며, 암호키 노출은 파일서버 비밀정보들의 키-정보 크래킹 위험을 야기하므로 반출용 암호키는 새로운 세션키로 바꾸어 일회성 암호화 상태로 키-정보 분리원칙에 따라서 분리된 이동 저장매체에 인증 키-정보와 함께 저장되어 반출한다. 허락된 외부장소에서 호스트는 응용 프로그램으로 GCKS에 세션키를 요청하여 복호한다. 이 때 반출된 자원들은 SNMP 관리 밖에 있어서 자원 도용으로 인한 정보유출위험이 커지므로 강력한 키-관리를 수행하고 구체적인 프로토콜은 5.1.4절을 참조하고 다음은 정보 유통 관점에서 기술한다.

1) 반출자

· 반출 요청한 사용자가 허락된 이동 호스트에 로그인 후 이동 저장매체를 연결한 상태에서 5.1.4절 프로토콜을 수행.



(그림 7) 반출 처리

· 이동 호스트(식별자=sid), 비밀정보(식별자=fid), 반출 사용자(식별자=uid)와 이동 저장매체(식별자=sid)는 반출 정보 처리에 대한 의미상 벡터 (what, how, where, who) = (fid, sid, did, uid)를 생성. 권한 및 이용 정책으로 등록.

· 이 벡터를 두 개체 사용자(uid)와 이동 호스트(did)의 공개키 V_{uid} , V_{did}로 암호화 $V_{did}V_{uid}(fid, sid, did, uid)$ 형태로 이동 저장매체에 인증 정보로 저장.

· 세션키 Σ_{fid}로 비밀정보 om_{fid}를 재-암호화 $\Sigma_{fid}K_{fid}^{-1}(om_{fid})\to mm_{fid}$을 수행. 비밀정보와 함께 이동 저장매체에 저장.

2) 이동 호스트

· 비밀정보 mm_{fid}를 처리하기 위해서 이동 저장매체를 검증. fid, sid, did, uid 를 발췌.

· 응용 프로그램이 비밀정보 mm_{fid}를 저장.

· $list_{ids}^{0011_{(2)}}$= ((fid, req) , $(sid, \varnothing)$, (did, V_{did}) , (uid, V_{uid})) 생성

· 키 요청 메시지 $M_{key-req}$생성. GCKS에 전송.

3) GCKS

· $M_{key-req}$의 fid로 세션키 Σ_{fid} 검색.

· (fid, Σ_{fid}) 을 삽입하여 $list_{ids}^{1011_{(2)}}$로 변경.

· 키 전달 메시지 M_{key} 생성. 세션키를 이동 호스트에 전송.

4) 이동 호스트

· M_{key}에서 세션키 Σ_{fid} 발췌.

· 복호모듈은 $\Sigma_{fid}^{-1}(mm_{fid})\to sm_{fid}$을 수행. 응용 프로그램에서 처리.

4.3 유통 키-관리 기본기능

MSEC 그룹 키-관리에서는 프로토콜의 공통 기능을 기술하여 프로토콜을 단순화한다[8,20]. 본 논문에서는 이 공통 기능 처리는 효율적이고 전체 성능에 영향을 미치지 않는다고 가정한다.

본 시스템이 주목하는 공통기능은 다음과 같다.

① 멤버 식별과 인증/검증
② 그룹 멤버십 확인
③ 키-정보 교환을 위한 안전채널 설정
④ 키-정보 불법이용 탐지

공통기능은 프로토콜 보안방식이나 운영 시나리오에 따라서 처리방법이 변한다. DRM 기반 보호는 상업적 안정성에만 초점을 맞추었고 정보보호에 취약성을 보이는 것은 공통기능을 고려하지 않기 때문이다.

본 비밀정보 유통 프로토콜에서는 2.1절의 강력한 보안조건을 만족시키기 위해 키-관리 방법에도 변화가 필요하지만 공통기능 처리방법을 살펴보자.

4.3.1 식별자와 인증 방법

그룹 멤버 식별자는 유일해야 하므로 고유하게 서버에서 그룹 내 식별자를 생성하는 방법과 기존 주소 IP 또는 MAC 주소를 이용하는 방법이 있다. 본 프로토콜은 유통 시스템에서만 고유한 독립적 식별자를 사용한다고 가정한다. 모든 자원은 다음과 같은 식별자로 구분한다.

① 정보파일 식별자
② 이동 저장매체 식별자
③ 호스트 식별자: 고정 호스트 식별자, 서버 식별자, 이동 호스트 식별자
④ 사용자 식별자

네 개의 식별자로 구성된 리스트를 인증 사인(sign)의 다이제스트로 사용한다. 이중 인증 사인(sign)은 호스트와 사용자의 개인키를 사용한다.

4.3.2 그룹 멤버십 확인

그룹 멤버십은 주기적 IGMP 질의 및 보고 메시지 교환으로 확인되고 멀티캐스트 라우터의 멤버 리스트가 갱신되고 해당 멤버는 탈퇴된다. 다른 한 가지는 SNMP에서 클라이언트 에이전트가 주기적 상태 보고와 트랩의 이상 발생 보고로서 해당 멤버를 그룹에서 강제 탈퇴 절차를 수행한다.

4.3.3 키-정보 안전 채널 설정

키-정보의 보안 연관성 (Security Association)은 메시지 전송과 키-정보 교환 암호기법에 관한 정보를 담고 있다. MSEC에서는 다음과 같은 키-관리 과정을 설명하고 있다.

초기 멤버 등록 단계에서 키-정보를 안전하게 교환하기 위해서는 안전채널을 확보하여 등록(registration) SA를 확립하고 최초 정보교환(Data) SA와 초기 키-변경(Rekey) SA를 확립한다. 그 다음 정보교환 SA와 키-변경 SA의 키-정보 변경 과정은 현 키-변경 SA를 활용하여 안전하게 수행한다.

4.3.4 키-정보 노출 위험 탐지

그룹 통신 키-정보 노출이 의심된다면 이미 노출된 것으로 간주하는 적극적 보호를 취하는 것이 비밀정보 유출방지의 전제 조건이다. 노출 위험이 높아지는 경우를 분석하면 다음과 같다.

1) SNMP 자원 관리에서 이상 발생 경보 발행: 저장매체 도난 위험 발생, 응용 프로그램 재-설치로 해킹 위험, 운영체계 변경으로 후킹 위험
2) IGMP 질의 보고에서 호스트 미동작 감지: 호스트 도난 위험
3) 반출시 호스트의 이동 저장매체 검증 실패: 이동 매체 도난 위험
4) 사용자와 호스트의 권한과 이용 정책이 불합치: 내부자 유출시도 위험
5) 반출 이동 호스트 동작시간과 복귀시간 미준수 탐지: 내부자 유출시도 위험

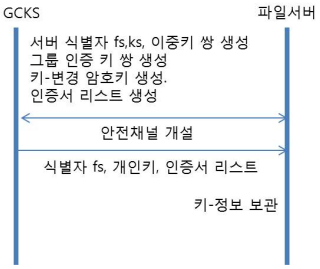
이런 위험이 판단되면 키-변경 과정을 수행한다.

5. 비밀정보 유통 그룹 키-관리 프로토콜

3,4장에서 비밀정보 유통 과정, 키-정보 이용, 그리고 키-관리 공통기능을 대한 논의를 했다. 본 장에서는 MSEC GKMP의 키-관리 프로토콜을 기반으로 비밀 정보 유통 프로토콜을 기술한다.

5.1 등록(registration) 프로토콜: 키-정보 생성 및 분배

5.1.1 서버 키-정보 설정



(그림 8) 서버 키-정보 설정

비밀정보 유통 시스템은 최초에 그룹 내 서버, 파일서버와 GCKS가 안전한 정보교환을 위한 키-정보(표 4)를 설정한다. 파일서버는 정보 식별자와 비밀정보 쌍(pair)을 보관하고, GCKS는 서버 인증/검증 키 쌍, 그룹 인증/검증 키 쌍, 비밀정보 암호키(세션키 포함), 호스트와 사용자 인증/검증 키 쌍, 이동 저장매체 인증/검증 키 쌍, 키-변경(rekey) 암호키 등 모든 키-정보의 생성/분배/보관하고, 멀티캐스트 라우터로서는 그룹 멤버 리스트를 유지하고, SNMP 관리자 에이전트가 실행된다. 그리고 정책서버로서는 비밀정보 유통 로그 정보, 권한과 이용정책을 기록한다. 그러나 본 논문은 정책서버의 구체적 기능은 기술하지 않는다.

다음은 유통 그룹 초기 서버 설정 단계에 관한 프로토콜이다.

① GCKS

- 자신 식별자와 비밀정보 파일서버 식별자로 ks와 fs를 생성
- 자신 및 비밀정보 파일서버의 인증/암호 이중키 쌍 (S_{ks}, V_{ks})와 (S_{fs}, V_{fs})을 생성.
- 그룹(g) 인증과 암호화 이중키 쌍 (S_g, V_g)을 생성
- 키-변경 암호키 K_g 생성
- 멤버 등록 때 호스트에 보내는 Rekey SA 키-정보와 개체 인증 키-정보 즉 그룹과 서버의 공개키와 키-변경 암호키를 담은 인증서의 리스트 $LIST_{cert}$ = Cert(K_g, V_g, V_{ks}, V_{fs})를 생성. 보관

② GCKS ↔ 파일서버

- 안전채널 개설

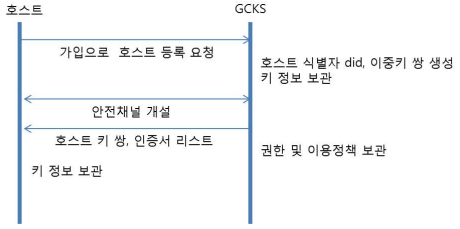
③ GCKS → 파일서버

- 파일서버 식별자, 개인키, 그룹 인증 키 쌍, 서버 공개키 인증서 리스트를 담은 ((fs, S_{fs}, S_g), $LIST_{cert}$)를 전송.

④ 파일서버

- 모든 키-정보는 보관

5.1.2 호스트 등록



(그림 9) 호스트 등록

비밀정보 유통 그룹에 호스트가 가입 요청을 보내면 다음 프로토콜 절차가 수행된다.

① 호스트 → GKCS

- IGMP 질의응답으로 가입 요청 메시지 보냄.

② GCKS

- 호스트 식별자 did 생성. 호스트 인증/암호 이중키 쌍 (S_{did}, V_{did}) 생성. (did, (S_{did}, V_{did}))

보관.

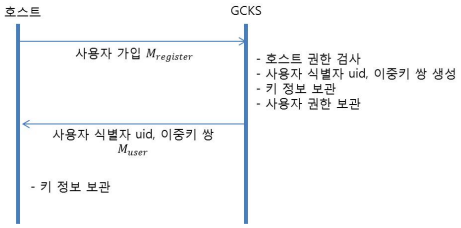
③ GCKS ↔ 호스트
- 안전채널 개설

④ GCKS → 호스트
- $((did, (S_{did}, V_{did})), LIST_{cert})$ 를 호스트로 전송
- 호스트(did)의 권한 $Auth_{did}$ 결정. 정책서버에 보관.

⑤ 호스트
- 호스트는 보관

5.1.3 사용자 등록

비밀정보 취급 사용자가 등록될 때 사용자 인증 정보를 보관할 개인 OTP(one time password) 플래시에 다음 절차를 따라서 생성된 키-정보를 보관한다.



(그림 10) 사용자 등록

① 사용자
- 등록된 호스트(did)에 로그인 후 사용자 가입 신청

② 호스트 → GCKS
- 호스트(did)는 사용자 식별자 생성 요청을 담은 $list_{ids}^{0010_{(2)}} = ((none, \varnothing), (none, \varnothing), (did, V_{did}), (req, \varnothing))$ 생성.
- 사용자 등록 요청 메시지 $M_{register} = (register, E_{h\to s}(list_{ids}^{0010_{(2)}}))$를 전송

③ GCKS
- $M_{register}$에서 $D_{h\to s}()$으로 $list_{ids}^{0010_{(2)}}$ 발췌.
- 호스트(did) 권한 $Auth_{did}$ 검사.
- 사용자 식별자 uid 생성, 사용자 인증/암호 이중키 쌍 (S_{uid}, V_{uid}) 생성. $(uid, (S_{uid}, V_{uid}))$ 보관.

④ GCKS → 호스트
- $(uid, (S_{uid}, V_{uid}))$를 그룹 및 서버의 인증 사인(sign) 후 호스트 공개키 암호화 하여 사용자 등록 통보 메시지 $M_{user} = (user, E_{s\to h}(uid, (S_{uid}, V_{uid})))$ 을 전송.

⑤ GCKS
- 사용자(uid)의 권한 $Auth_{uid}$ 결정. 정책서버에 보관

⑥ 호스트
- M_{user} 복호와 검증 후 OTP 플래시에 $(uid, (S_{uid}, V_{uid}))$ 보관.

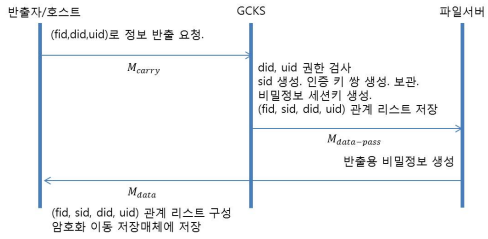
5.1.4 비밀정보 반출

비밀정보를 반출할 때 비밀정보를 저장할 이동 저장매체, 취급 사용자, 처리 이동 호스트를 등록한다. 이동 호스트는 서버에 등록되어 있어야 한다.

① 사용자
- 이동 호스트에 로그인
- 비밀정보 fid 검색. 처리 이동 호스트(did)와 이동 저장매체의 반출 요청

② 이동 호스트 → GCKS
- 이동 저장매체 식별자 요청을 담은 $list_{ids}^{0011_{(2)}} = ((fid, \varnothing), (req, \varnothing), (did, V_{did}), (uid, V_{uid}))$ 생성.
- 반출 요청 메시지 $M_{carry} = (carry, E_{h\to s}(list_{ids}^{0011_{(2)}}))$을 전송

③ GCKS
- M_{carry}를 $D_{h\to s}()$로 $list_{ids}^{0011_{(2)}}$을 발췌. 사용자(uid)와 이동 호스트(did)의 권한 검사. 정보 식별자 fid을 발췌.
- 이동 저장매체 식별자 sid 생성. 인증 이중키 (S_{sid}, V_{sid}) 생성. $(sid, (S_{sid}, V_{sid}))$ 보관.



(그림 11) 정보 반출

- 비밀정보 세션키 Σ_{fid} 생성.
- 비밀정보 이용 개체의 관계 리스트 (fid, sid, did, uid)를 저장

④ GCKS → 파일서버

- $list_{ids}^{1111_{(2)}}$=$((fid, \Sigma_{fid}), (sid, V_{sid})(did, V_{did}), (uid, V_{uid}))$을 생성
- 비밀정보 전달요청 메시지 $M_{data-pass}$ = $(data-pass\ E_{ks\to fs}(list_{ids}^{1111_{(2)}}))$를 전송.

⑤ 파일서버

- $M_{data-pass}$를 $D_{ks\to fs}()$로 $list_{ids}^{1111_{(2)}}$ 발췌. 비밀정보 fid로 검색. om_{fid} 받음
- 재-암호화 $mm_{fid}=\Sigma_{fid}(K_{fid}^{-1}(om_{fid}))$ 후 반출용 비밀정보 mm_{fid} 보관.

⑥ 파일서버 → 이동 호스트

- $list_{ids}^{0111_{(2)}}$ = $((fid, \varnothing), (sid, V_{sid}), (did, V_{did}), (uid, V_{uid}))$를 생성.
- 비밀정보 메시지 M_{data} = $(mm_{fid}, E_{s\to h}(list_{ids}^{0111_{(2)}}))$를 보냄

⑦ 이동 호스트

- M_{data}에서 비밀정보 mm_{fid}을 발췌. $D_{s\to h}()$로 $list_{ids}^{0111_{(2)}}$ 에서 개체 관계 리스트 (fid, sid, did, uid) 을 생성.
- 비밀정보 mm_{fid}과 개체 관계 리스트를 암호화하여 $V_{did}V_{uid}((fid, sid, did, uid))$ 을 이동 저장매체에 저장.

5.4.2 키 이용

1) 일반 비밀정보 이용

호스트가 비밀정보와 암호키를 요청하는 프로토콜이다.

① 사용자

- 호스트에서 응용프로그램을 사용하여 비밀정보(fid) 요청

② 호스트 → GCKS

- 호스트(did)가 비밀정보 요청 메시지 $M_{req}=(req, E_{h\to s}(list_{ids}^{0011_{(2)}}))$ 을 전송

③ GCKS

- M_{req}를 $D_{h\to s}()$로 $list_{ids}^{0011_{(2)}}$을 발췌. 사용자(식별자=uid)와 호스트(식별자=did)의 권한 검사.
- 비밀정보(fid) 암호키 K_{fid} 검색.

④ GCKS → 파일서버

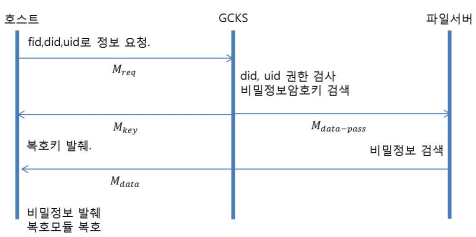
- $list_{ids}^{0011_{(2)}}$ = $((fid, \varnothing), (none, \varnothing), (did, V_{did}), (uid, V_{uid}))$ 생성.
- 정보전달 요청 메시지 $M_{data-pass}$ = $(data-pass\ ,\ E_{ks\to fs}(list_{ids}^{0011_{(2)}}))$를 전송.

⑤ 파일서버 → 호스트

- $M_{data-pass}$를 $D_{ks\to fs}()$으로 $list_{ids}^{0011_{(2)}}$ 발췌. 암호화 비밀정보(fid) om_{fid}를 검색.
- 비밀정보 전달 메시지 M_{data} = $(data, om_{fid}, E_{s\to h}(list_{ids}^{0011_{(2)}}))$를 전송

⑥ GCKS → 호스트

- $list_{ids}^{1011_{(2)}}$ = $((fid, K_{fid}), (none, \varnothing), (did, V_{did}), (uid, V_{uid}))$ 생성.



(그림 12) 일반 정보 이용

· 키-전송 메시지 M_{key} = (key, $E_{s\to h}(list_{ids}^{1011_{(2)}})$)을 전송

⑦ 호스트

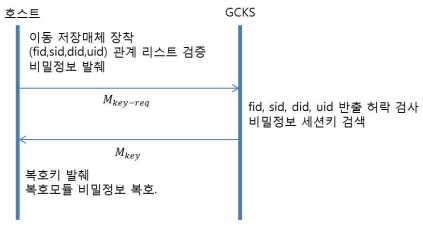
· M_{key}를 $D_{s\to h}()$로 $list_{ids}^{1011_{(2)}}$ 발췌. K_{fid} 발췌.

· M_{data}에서 om_{fid} 발췌

· 복호모듈은 $K_{fid}(om_{fid})$로 sm_{fid} 얻어 처리.

2) 반출 비밀정보 이용

반출한 비밀정보를 처리하기 위해 세션키의 요청 프로토콜이다.



(그림 13) 반출 정보 복호키

① 사용자

· 이동 호스트(did)에 로그인 후 이동 저장매체(sid)를 장착 후 처리 요청.

② 이동 호스트

· 이동 저장매체(sid)에서 개체 관계 리스트 $V_{did}V_{uid}((fid, sid,did,uid))$ 와 비밀정보 mm_{fid}을 읽고 리스트 복호 후 (fid, sid,did,uid) 얻음

③ 이동 호스트 → GCKS

· 키 요청 메시지 $M_{key-req}$ = ($key-req$, $E_{h\to s}(list_{ids}^{0011_{(2)}})$)을 보냄

④ GCKS

· $M_{key-req}$를 $D_{h\to s}()$로 $list_{ids}^{0011_{(2)}}$ 발췌. 반출 허락 여부 검사.

· 사용자(uid)와 호스트(did)의 권한 검사. 비밀정보(fid) 세션키 Σ_{fid}를 검색

⑤ GCKS → 이동 호스트

· $list_{ids}^{1111_{(2)}}$ = ((fid, Σ_{fid}), (sid, V_{sid}), (did, V_{did}), (uid, V_{uid})) 생성.

· 세션키 전송 메시지 M_{key} = (key, $E_{s\to h}(list_{ids}^{1111_{(2)}})$)을 전송

⑥ 이동 호스트

· M_{key}를 $D_{s\to h}()$로 $list_{ids}^{1111_{(2)}}$ 발췌. 복호모듈은 $\Sigma_{fid}(mm_{fid})\to sm_{fid}$ 수행 비밀정보 처리

5.3 키-변경 (rekey)

4.4.3절에서 노출 위험이 의심되는 상황에서 키-변경 프로토콜을 수행하여 불법 유출의 위험을 원천 봉쇄한다. 보통 키-변경은 주기적으로 수행하거나 호스트와 사용자의 인증/암호 이중키 쌍의 개인키의 노출이 의심되는 경우에 수행한다. 그러나 본 시스템에서 위험한 상황은 반출에서 발생할 가능성이 높다. 주로 이동 호스트와 이동 저장매체의 분실이나 동작 중지 상황에서 정보 불법 복제로 인한 노출에 주의해야 한다. 이 경우 모든 Data SA, Rekey SA의 키-정보를 모두 변경해야 한다. 키-변경에 사용할 암호키는 K_g^t 이다.

5.3.1 호스트 독립 키-정보 변경

그룹 키-정보 중 특정 호스트에 종속되지 않은 키-정보는 Data SA로 GCKS 공개키, 파일서버 공개키, 그룹 공개키이고, Rekey SA로 그룹 키-변경 암호키이다.

이런 모든 키-정보는 반드시 합법적 서버에서 생성되고 분배된다는 것을 증명해야 하므로 인증서 형태로 서버의 싸인(sign)이 포함된 인증서 리스트 $LIST_{sert} = Cert(K_g, V_g, V_{fs}, V_{ks})$ 로 구성된다.

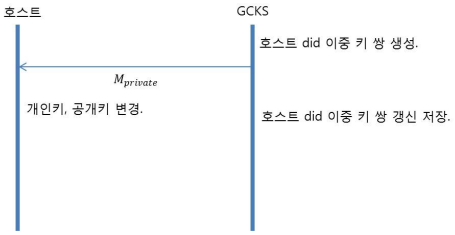
여기서 키-정보 변경은 현재 t, Rekey SA K_g^t를 사용해서 수행된다. 예를 들어 시간에 따른 변경은 $V_g^t \to V_g^{t+1}$ 과 같이 V_g^t는 현재 t 그룹 공개키이고 V_g^{t+1}는 다음 $t+1$의 그룹 공개키를 표현한다.

① GCKS

· 변경할 Data SA 정보는 서버 이중키 쌍들,

GCKS 인증 키 쌍 $(S_{ks}^{t+1}, V_{ks}^{t+1})$ 와 파일서버 인증 키 쌍 $(S_{fs}^{t+1}, V_{fs}^{t+1})$,그룹 인증 키 쌍 (S_g^{t+1}, V_g^{t+1}) 를 생성. Rekey SA 암호키 K_g^{t+1}를 생성

· 변경할 키-정보를 담은 인증서 리스트



(그림 14) Rekey - 종속 키정보

$LIST_{cert}^{t+1} = Cert(K_g^{t+1}, V_g^{t+1}, V_{fs}^{t+1}, V_{ks}^{t+1})$ 를 생성.

② GCKS → 전체 호스트

· 호스트($i = 1..n$)에 키-변경 메시지 M_{rekey_i} 를 생성. M_{rekey_i} = $(rekey_i(E_{s\to h}^t K_g^t(LIST_{cert}^{t+1}))$ 를 전송.

③ GCKS → 파일서버

· 서버 설정 단계처럼 $E_{ks\to fs}^t K_g^t(fs, S_{fs}^{t+1}, S_g^{t+1}, LIST_{cert}^{t+1})$ 을 파일서버에게 보냄

④ 호스트

· M_{rekey_i}을 $(K^{-1})_g^t D_{s\to h}^t()$로 $LIST_{cert}^{t+1}$ 을 발췌. 보관, 과거 GKEK (K_g^t)는 폐기

⑤ 파일서버

· 복호 후 $(fs, S_{fs}^{t+1}, S_g^{t+1}, LIST_{cert}^{t+1})$ 를 보관. 과거 GKEK 및 GTEK 키-정보는 폐기

5.3.2 호스트 종속 키-정보 변경

호스트 종속 키-정보는 호스트의 인증키이다. 노출 위험이 예상되는 모든 호스트의 $(did, (S_{did}, V_{did}))$ 는 변경되어야 한다. 이 호스트 종속 키-정보의 변경은 다음과 같다.

① GCKS

· 호스트 $i(i = 1..n)$를 위한 미래 $t+1$ Data SA로서 이중키 쌍 (S_i^{t+1}, V_i^{t+1}) 생성.

② GCKS → 호스트 $i(i = 1..n)$:

· 개인키 전달 메시지 $M_{private}$ = $(private$, $E_{s\to h}^t K_g^t(i, (S_i^{t+1}, V_i^{t+1}))$)을 해당 호스트(i)에 전송

③ GCKS

· 키-정보 테이블에 호스트(i) 관련 새로운 이중키 쌍 $(i, (S_i^{t+1}, V_i^{t+1}))$ 로 갱신

④ 호스트

· 복호 후 $(i, (S_i^{t+1}, V_i^{t+1}))$ 로 대체 보관

6. 성능분석

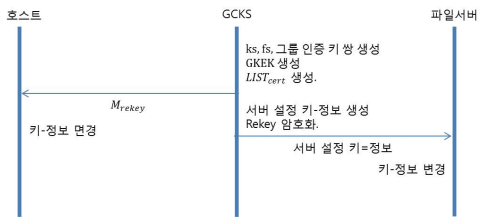
본 성능분석은 첫째 안전성 성능과 둘째 수행시간 성능으로 나누어 살펴본다.

먼저 안전성 성능은 기밀성, 무결성, 인증성, 부인봉쇄에 대한 안전성과 서버 GCKS에 대한 서비스거부(DoS) 공격과 재생(Replay) 공격에 대한 안전성이다.

· 기밀성

본 시스템의 암호화 유통원칙은 복호기법만 노출되지 않으면 비밀정보에 대한 기밀성은 보장된다.

기밀성의 문제는 암호키의 노출이다. 그러나 본 시스템에서 암호키는 $list_{ids}^{1011_{(2)}}$의 첫 항목에 담겨지고 공중키로 두 번 암호화 되고 개인키로 이



(그림 15) Rekey - 독립 키-정보

중 인증 싸인(sign)이 되므로 비대칭키의 동일한 안전성을 유지할 수 있고(그림 4, 표 3 참조)따라서 크래킹이 불가능하다.

두 번째 문제는 암호키와 비밀정보가 함께 저장된 호스트를 반출하여 비밀정보를 복호화 하려는 것이다. 이것은 키와 정보 분리원칙으로 불가능한 상황이고, 혹 일시 저장 키-정보와 평문화된 비밀정보를 복원하여 복호화를 시도하는 경우는 키-정보 일회 사용원칙으로 불가능한 경우이다.

IGMP 그룹 관리 메시지를 스니핑(sniffing) 하여 식별자를 알아내서 도용할 가능성이다. 이런 그룹 관리 정보와 키 정보는 $list_{ids}^{1011_{(2)}}$에 담아서 보호하므로 불가능하다.

마지막으로 그룹 내 등록 자원을 탈취하여 그 안에 담긴 정보를 이용하여 해킹하려는 경우이다. 그러나 네트워크 자원 관리로 인해서 자원의 탈취는 이상 경보를 발행하고 그 즉시 모든 키-정보는 변경되므로 과거 정보는 이용할 수가 없다.

기밀성을 크래킹 하려는 모든 경우를 방지하는 원칙을 본 시스템은 적용하고 있어서 기밀성 보장은 매우 높다고 할 수 있다.

· 무결성:

비밀정보는 암호화 상태로만 유통이 가능하므로 사실상 기밀성 크래킹이 없이는 정보내용의 변경은 불가능하다. 그러면 해커가 조작한 비밀정보를 등록하여 시스템을 공격하는 것이 가능한가? 이는 해커가 최소한 사용자와 호스트의 인증키 쌍 중 개인키를 알아야 하지만 이는 호스트와 사용자 OTP를 동시에 탈취해야 가능하다. 생일공격과 같은 공격이 가능한가? 실제 내용의 추측이 가능해도 인증키 해킹이 불가능하므로 공격은 성립되지 않는다. 암호화 유통 원칙은 가장 안전한 메시지 인증 코드라고 할 수 있다.

· 인증성

초기 배달된 그룹 관련 키-정보는, 호스트에서 서버로 전송 사용하는 호스트 인증 개인키와 사용자 인증 개인키로 이중 인증 사인(sign)을 하므로 관련된 개체 검증이 가능하고, 서버에서 호스트로 전송에서 서버 인증 개인키와 그룹 제어를 증명하는 그룹 인증 개인키로 이중 인증 사인(sign)을 하므로 역시 검증이 가능하다(그림 4, 표 2 참조). 개체의 합법성을 의미하는 인증과 검증은 완벽하게 이루어진다.

· 부인봉쇄:

이중 인증 원칙으로 그룹 내 사용자와 호스트, 서버, 그룹 개인키 사인(sign)은 교환을 부인할 수가 없다.

· 서비스거부(DoS) 공격:

서비스거부 공격이 가능하려면, 외부에 있도록 허락된 합법적 호스트와 사용자를 가장한 그룹 관리 메시지를 서버에 보내는 것이 가능해야 한다. 그러나 이것은 반출 허락된 기록과 권한 정책에서 네 가지 식별자의 의미상 연결이 전부 검사되어야 하므로 성립될 수 없다.

특별히 허가가 없는 그룹 관리 메시지를 외부에서 다량 보내는 행위는 암호키 요청 메시지를 통해서 가능하지만 이것을 만들어 내는 일은 응용 프로그램의 복호모듈을 모방해야 하므로 불가능하다.

· 재생(Replay) 공격

모든 그룹 메시지는 자신의 개인키와 상대의 공개키, 그룹 공개키로 암호화 되어 있어서 재생공격을 위한 정보를 한 조각도 얻을 수가 없다. 성립할 수 없는 공격이다.

이런 안전성과 수행시간 성능과 반비례(tradeoff) 관계에 있다는 것이 일반적이지만, 본 비밀정보 유통 시스템은 암호화 유통원칙으로 비밀정보의 생성 단계에서 암호화 되어, 유통 단계에서는 암복호화 부담은 전혀 없으므로 수행속도

(표 4) 시뮬레이션 인자조건 (p=1, B는 byte)

구 분	값
메시지 크기	s 바이트
식별자 길이	32 비트 (4 바이트)
대칭 암호기법	3DES
인증 기법	DSA
암호키 길이	128 비트 (16 바이트)
다이제스트 길이	64 비트 (8 바이트)
개인키 수행시간	100p
공개키 수행시간	1000[1]p
대칭암호 수행시간	s
식별자 리스트 평균	a (16B < a < 54B)
반출 횟수	c
이용 횟수	n

(표 5) 기존 방식과 연구의 암호기법 횟수 비교

비교	작업구분	순서	장소	인증/검증	이중키 암호/복호
연구 시스템	생성	1	호스트	2 / 0	2 / 0
		2	키서버	2 / 2	2 / 2
		3	호스트	2 / 2	2 / 2
		4	파일서버	0 / 2	0 / 2
	이용	1	호스트	2 / 0	2 / 0
		2	키서버	1 / 2	2 / 2
		3	파일서버	2 / 1	2 / 2
		4	호스트	0 / 2	0 / 2
	반출이용	1	호스트	2 / 1	2 / 0
		2	키서버	2 / 2	2 / 2
		3	호스트	0 / 2	0 / 2
	반출요청	1	호스트	2 / 0	2 / 0
		2	키서버	2 / 2	1 / 2
		3	파일서버	2 / 2	2 / 1
		4	호스트	0 / 2	0 / 2
기존	생성	1	호스트	1 / 0	1 / 0
		2	파일서버	0 / 1	0 / 1
	이용	1	파일서버	1 / 0	1 / 0
		2	호스트	0 / 1	0 / 1

가 빨라질 것으로 판단된다. 대부분의 수행될 암호기법의 종류와 횟수는 다음 표 5, 6과 같고 암호기법 수행을 평가할 때 사용할 인수(파라미터)도 표 4에 기술되어 있다.

크기 s인 비밀정보의 생성부터 n번 이용, 그 중 c 번 반출 후 폐기되는 동안 수행하는 암호화 횟수로 수행 성능을 평가해 보자. 이를 반영하는 것이 표 7, 8에 있다.

(표 5-c) 기존 방식과 연구의 암호기법 횟수 비교

비교	작업구분	순서	장소	대칭키 암호/복호	해쉬	키생성
연구시스템	생성	1	호스트	0 / 0	0	0
		2	키서버	0 / 0	0	1
		3	호스트	1 / 0	0	0
		4	파일서버	0 / 0	0	0
	이용	1	호스트	0 / 0	0	0
		2	키서버	0 / 0	0	0
		3	파일서버	0 / 0	0	0
		4	호스트	0 / 1	0	0
	반출이용	1	호스트	0 / 0	0	0
		2	키서버	0 / 0	0	0
		3	호스트	0 / 1	0	0
	반출요청	1	호스트	0 / 0	0	0
		2	키서버	0 / 0	0	0
		3	파일서버	1 / 1	0	1
		4	호스트	0 / 0	0	0
기존	생성	1	호스트	1 / 0	1	1
		2	파일서버	0 / 1	1	0
	이용	1	파일서버	1 / 0	1	1
		2	호스트	0 / 1	1	1

본 비밀정보 유통의 총 수행시간이 $1\times t_{생성}+n\times t_{이용}+c\times(t_{반출요청}+t_{반출이용})$ 이다. 비밀정보 유통 수행시간을 계산하면 다음과 같다.

- $a = Average(list_{ids}^{(i)})$ for $i = 0..7$
- $t_{생성} = 12\times 1000\times p\times a + 12\times 100\times p\times a + s$ $= 13200pa + s$
- $t_{이용} = 12100pa + s$, $t_{반출요청} = 12100pa + 2s$, $t_{반출이용} = 9800pa + s$
- $total_{연구}(n,c,s)$ $= 13200pa + s$ $+ n\times(12100pa + s)$ $+ c\times(21900pa + 3s)$, $total_{연구}(n,c,s)$ = $(12100n + 21900c + 13200)pa$ + $(n + 3c + 1)s$

1) 이중키 암호화 수행시간은 대칭키의 100~1000배 정도 느리고, 공개키 암호화가 개인키 보다 느리다.

(표 6) 기존 방식과 연구의 암호기법 횟수 종합

비교	작업구분	순서	장소	공개키/개인키/대칭키
연구 시스템	생성	1	호스트	12 / 12 / 1
		2	키서버	
		3	호스트	
		4	파일서버	
	이용	1	호스트	11 / 11 / 1
		2	키서버	
		3	파일서버	
		4	호스트	
	반출이용	1	호스트	9 / 8 / 1
		2	키서버	
		3	호스트	
	반출요청	1	호스트	11 / 11 / 2
		2	키서버	
		3	파일서버	
		4	호스트	
기존 시스템	생성	1	호스트	2 / 2 / 4
		2	파일서버	
	이용	1	파일서버	2 / 2 / 4
		2	호스트	

반면에 기존 방식은 총 수행시간을 계산하면 다음과 같다.

- $t_{생성} = 2 \times 1000 \times 8 \times p + 2 \times 100 \times 8 \times p + 4s$
  $= 17600p + 4s$
- $t_{이용} = 17600p + 4s = t_{반출요청} = t_{반출이용}$
- $total_{기존}(n,c,s) = 17600(n+2c+1)p + 4(n+2c+1)s$

(표 7) 비밀유통($total_{연구}()$) 암호기법과 기존($total_{기존}()$) 암호기법의 수행시간 비교 (n을 따라서)

n	연구 시간	기존 시간	조건
2	1,315	1,308	s=61KB c=n/2회 a=16B unit=1K
3	1,836	1,831	
4	2,357	2,354	
5	**2,879**	**2,878**	
6	3,400	3,401	
7	3,921	3,924	
8	4,443	4,447	
9	4,964	4,970	
10	5,485	5,494	
11	6,007	6,017	
12	6,528	6,540	
13	7,049	7,063	
14	7,570	7,586	
15	8,092	8,110	
16	8,613	8,633	
17	9,134	9,156	
18	9,656	9,679	
19	10,177	10,202	
20	10,698	10,726	
21	11,220	11,249	

(표 8) 비밀유통 암호기법과 기존 암호기법의 수행시간 비교 (s을 따라서)

s	연구 시간	기존 시간	조건
1	37,342	4,342	n=100회 c=50회 a=16B unit=1K
11	39,852	12,382	
21	42,362	20,422	
31	44,872	28,462	
41	47,382	36,502	
51	49,892	44,542	
61	**52,402**	**52,582**	
71	54,912	60,622	
81	57,422	68,662	
91	59,932	76,702	
101	62,442	84,742	
111	64,952	92,782	
121	67,462	100,822	
131	69,972	108,862	
141	72,482	116,902	
151	74,992	124,942	
161	77,502	132,982	
171	80,012	141,022	
181	82,522	149,062	
191	85,032	157,102	

수행 횟수 즉 시간이 지남에 따른 전체 성능 관점에서 위 두 암호기법 수행시간을 비교해 보면 다음 표 7, 8와 같다. 표 7, 8에서 보여주는 위 결

과는 원 자료의 1/100로 조정했다. 메시지의 크기가 클수록 본 연구의 시스템의 수행 성능은 우수해지고, 또 이용횟수가 증가할수록 본 시스템의 성능은 우수해진다는 것을 알 수 있다.

7. 결 론

아주 복잡하여 많은 것을 상호 연결되도록 구조화하는 작업이 필요한 시스템이 멀티캐스트 그룹 키-관리 프로토콜이다. 그 프로토콜의 복잡성에 비해서 본 논문이 제시한 보안조건의 원칙은 성능분석 결과는 매우 우수하게 변화시킨 것으로 나타났다. 이전에 이미 기업정보 유출 원천봉쇄 시스템 구조에 관한 선행 연구로 비밀정보 유통 보안조건과 보안 조건이 만족되도록 키-관리 프로토콜을 설계할 수 있었다.

그러나 무엇보다도 도전이 되었던 DRM 기반 문서보호의 많은 취약점은 본 시스템으로 해소되었다고 생각한다. 그러나 여전히 추후 연구가 많이 남아 있다. 무엇보다도 프로토콜 사양을 작성하는 것이고 두 번째 애드혹(adhoc) 단말로만 이루어진 무선 환경에서 빠른 인증과 키 전달방식을 연구하는 것이다. 본 시스템에서는 단순화를 위해서 반출 장비가 적다는 가정에서 무선 환경을 다루고 있다. 그러나 기업 환경은 무선장치를 통한 기업비밀유통이 필요한 시나리오가 많이 있으므로 새로운 도전이 될 것이다.

참고문헌

[1] 홍보실 손세원, “국내기업의 기밀유출 대응 실태 조사”, 대한상공회의소 보도자료, 2007. 11.20., http://km.korcham.net/Common/FileDown.jsp?fname=20071121012_K.pdf&f_adr=/File/Dataroom/200801/, 2010.04.11 참조

[2] 정주미, “중기 3년간 산업기밀 유출 피해 4조원 대”, 재경일보 온라인, 2010.01.11, http://news.jknews.co.kr/article/news/20100111/6007293.htm, 2010.04.11 참조

[3] Harney, H., Muckenhirn, C. and T. Rivers, "Group Key Management Protocol Architecture", RFC 2094, September 1994.

[4] Harney, H., Muckenhirn, C. and T. Rivers, "Group Key Management Protocol Specification", RFC 2093, September 1994.

[5] Maughan, D., Schertler, M. Schneider, M. and J. Turner, "Internet Security Association and Key Management Protocol, Version 7", February 1997.

[6] B. Cain, I. Kouvelas, B. Fenner, A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002

[7] J. Case, M. Fedor, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)", RFC 1157, May 1990

[8] T. Hardjono, B. Weis, "The Multicast Group Security Architecture", RFC 3740, March 2004

[9] 김종우, 양원일, 한승조, “네트워크 상에서 디지털 콘텐츠 보호를 위한 DRM 프레임 설계”, 정보보호학회논문지 Vol.16 No.3, 2006. 6, pp. 101~113

[10] 홍강한, 김인한, “IFC 표준모델기반 건설 도면정보의 DRM 적용에 관한 기초연구”, 대한건축학회 창립60주년기념 학술발표대회논문집 제25권 제1호(건축계획), 2005. 10, pp. 315~318

[11] 문진규, “내부 정보 유출 발지를 위한 DRM 적용 방법 설계”, 한국정보과학회 2007, 한국컴퓨터종합학술대회 논문집 제34권 제1호 (D), 2007. 6, pp.7~10

[12] 이광우, 김승주, “기업 비밀정보 유출 방지 및 보호 관점에서의 디지털 복합기 보안 기술 동향 분석”, 정보보호학회지 제20권 제1호, 2010. 2, pp. 47~55

[13] 성경상, 오동열, 김정재, 나원식, 오해석, “전자문서 보관 서비스의 정보유출 최소화를 위

한 효율적 시스템 설계에 관한 연구", 한국통신학회논문지 제33권 제10호(통신산업응용), 2008. 10, pp. 350~358

[14] 김용, "온라인 환경에서의 전자문서 안전배포 및 이용을 위한 인증방법 설계 및 구현", 정보관리학회지 제25권 제1호, 2008. 3, pp.75 ~ 98

[15] 성경상, 오동열, 김정재, 나원식, 오해석, "전자문서 보관 서비스의 정보유출 최소화를 위한 효율적 시스템 설계에 관한 연구", 한국통신학회논문지 제33권 제10호(통신산업응용), 2008. 10, pp. 350 ~ 358

[16] 박경철, 정경석, 박우석, 염수열, 김세원, "PKI 기반의 저작권 보호 기술 및 장치 개발", 정보통신산업진흥원 연구보고서, 2002. 4.30

[17] R Anderson, "Two remarks on public key cryptology", 1997 Advances in Cryptology, Asiacrypt 96, Springer LNCS vol 1163 pp,26 - 35

[18] Michel Dupuy, Pierre Paradinas, "Trusted information: the new decade challenge : IFIP TC11 16th International", Kurwel Academic Publisher, MA, 2001

[19] SIMON BLAKE-WILSON, "Information Security, Mathematics, and Public-Key Cryptography", 2000 Kluwer Academic Publishers, Boston. 2000

[20] M. Baugher, R. Canetti, L. Dondeti, F. Lindholm, "The Multicast Security (MSEC) Group Key Management Architecture", RFC 4046, April 2005

[21] 최정현, "전자기밀문서 유출봉쇄 유통시스템 구조 연구", 인터넷정보학회논문지 제11권 제4호, 2010. 8., pp143 ~ 158

◐ 저 자 소 개 ◑

최 정 현

1984년 서울대학교 컴퓨터공학과(공학사)
1988년 미국 조지아공과대학교(GIT) 대학원 컴퓨터학과(이학석사)
1992년 미국 알라바마(Alabama) 주립 어번(Auburn)대학교 대학원 컴퓨터공학과(공학박사)
1994~현재 광운대학교 경영대학 경영정보학과 교수
관심분야 : 인터넷 프로토콜, 정보보안, 인공지능, 서비스기반기술 etc.
E-mail : chchoi@kw.ac.kr