AGR-VII-2019-②-84

금융부문 암호기술 활용 가이드

2019. 1.

금융보안원
FINANCIAL SECURITY INSTITUTE

금융부문 암호기술
활용 가이드

2019. 1.

제·개정 이력

버전	제·개정일	내용
1.0	2014. 12.	- 금융부문 암호기술 활용 가이드 제정
2.0	2019. 1.	- 사용 제한된 암호 알고리즘 관련 내용 삭제 및 축소 - 가이드 개정 사유 및 항목 추가 - 키 길이에 따른 안전성 유지기간 관련 내용 2030년 기준으로 수정 - TLS 1.0/1.1의 사용 제한 및 TLS 1.2 이상의 프로토콜 권장 내용 추가 - 블록암호 및 해시 알고리즘의 변경 예시 추가 - '암호 관련 금융IT 보안 컴플라이언스 통제사항' 수정 - 포스트 양자암호 및 동형암호에 대한 내용 추가

< 목 차 >

<그림 목차>

<표 목차>

제1장 개요

제1장 개요

제1절 개정 사유 및 항목

최근 TDES 암호 알고리즘 사용 제한 및 포스트 양자 암호, 동형 암호와 같은 새로운 분야의 암호 알고리즘 등장 등 암호기술 시장에 변화가 있었다. 이러한 암호기술 시장 현황을 반영하고 가이드 내용을 현재 기준으로 업데이트하기 위하여 본 가이드를 개정하였다.

❍ TDES 및 일부 암호 알고리즘 사용 제한

TDES 암호 알고리즘과 관련된 내용을 모두 삭제하고, SHA-1 등 용도에 따라 권고하지 않는 알고리즘에 관한 내용을 삭제하거나 축소하였다. 또한, TLS 1.0/1.1의 취약성에 관한 내용을 추가하였다.

❍ 암호 알고리즘 추가

최근 새로운 분야의 암호 알고리즘이 등장함에 따라, 본문 및 부록에 포스트 양자 암호와 동형 암호에 대한 내용을 추가하였다.

제2절 배경 및 목적

인터넷뱅킹, 자동화 기기, IC카드, 모바일 결제 등 많은 분야에서 사용되고 있는 암호기술은 IT 기술 발전과 금융서비스 환경이 변화됨에 따라 모든 전자금융환경에 필수 고려요소가 되고 있다. 국내의 경우, 개인정보 유출에 관한 전자금융사고가 사회적 이슈로 붉어지면서 관련 법률의 개정과 더불어 암호기술사용이 의무화되고 있으며 암호기술의 필요성 또한 더욱 중시 되었다.

국내외 암호기술 전문기관에서는 고도화되는 컴퓨팅 성능과 해킹기술에 주목하면서 암호기술의 안전성 및 신뢰성 증대를 위해 지속적으로 변경 지침을 제시하고 있다. 하지만, 이론적으로 어떤 암호기술도 해킹을 완벽히 막을 수 있다고 보장할 수 없고 실제 암호기술을 적용할 때에도 시스템 변경에 소요되는 기간과 비용문제 등으로 사고원인이 밝혀진 후에도 단기간에 문제를 해결하여 보안성을 확보하는 것이 쉽지 않다.

국내 금융권도 시스템에 적용된 암호 중 잠재적 취약성을 내재한 알고리즘이 적시에 변경되지 못하고 그대로 사용되고 있는 것이 현실이다. 따라서 안전성이 저하된 암호기술로 인해 기밀성, 무결성 등이 깨지게 되는 위협과 그 영향을 줄이기 위해서 각 분야에서는 안전한 보안수준을 유지할 수 있도록 꾸준히 규격에 이를 반영하고 권고하고 있다.

본 가이드는 이러한 노력의 일환으로 국내 금융부문 암호기술 적용현황을 살펴보고 향후 안전성이 저하될 암호기술을 변경할 때 고려해야하는 주요 보안 고려사항을 기술하여, 담당자들의 암호기술 활용 및 이해에 도움을 주고자 한다.

제3절 구성

본 가이드는 배경 및 목적, 용어정의 등의 가이드라인 개요를 시작으로 2장에서 금융부문 암호기술 적용대상 및 현황을 기술하고 3장에서 활용 시 고려사항을 기술하였다.

제4절 용어 정의

본 가이드에서 사용되는 용어 정의는 다음과 같다.

- 해시함수: 임의 길이의 메시지를 항상 고정된 길이의 해시값으로 변환하는 일방향 함수

- 일방향 함수: 계산하기는 쉽지만 역을 구하는 것은 어려운 함수로, 대표적인 일방향 함수는 해시함수가 있음

- 블록암호: 평문을 일정한 블록 크기로 나누어, 각 블록을 송신자, 수신자 간에 공유한 비밀키를 사용하여 암호화 하는 방식

- ECC(Elliptic Curve Cryptography): 공개키 암호 알고리즘 종류 중에 하나로서, 타원곡선 암호체계에서의 공개키 암호

- 생체 정보: 지문, 얼굴, 홍채, 정맥, 음성, 필적 등 개인을 식별할 수 있는 신체적 또는 행동적 특징에 관한 정보로서 이를 통해 가공되거나 생성된 정보를 의미함

- SSL(Secure Socket Layer): 데이터를 송수신하는 두 컴퓨터 종단의 전송

계층과 어플리케이션 계층(HTTP, TELNET, FTP 등) 사이에 위치하여 인증, 암호화, 무결성을 보장하는 표준 프로토콜로, 이후 TLS(Transport Layer Security)라는 이름으로 표준화됨

- 보안 서버: 인터넷상에서 전송되는 자료를 암호화하여 송수신하는 기능을 제공하는 웹 서버를 지칭

- MAC(Message Authentication Code): 메시지의 인증에 쓰이는 작은 크기의 정보로서, 전자서명과 유사하게 데이터 인증과 무결성 기능을 제공하지만 부인방지는 제공하지 못함

- MS카드: 마그네틱 방식(MS)의 카드로서 자성체에 개인 고유의 데이터를 기록하며, 정보 입출력이 쉽기 때문에 카드 위변조가 쉬움

- EMV: 국제적 호환을 위해 Europay, Master, Visa가 공동으로 사용하는 IC카드의 표준규격

- PKCS(Public-Key Cryptography Standards): 미국의 RSA Security가 제정한 공개키 암호 알고리즘 표준

제2장 금융부문 암호기술 적용대상 및 현황

제2장 금융부문 암호기술 적용대상 및 현황

제1절 금융부문 암호화 대상

전자금융은 고객에게 본인 확인 및 부인 방지 등의 수단을 제공함으로써 금융회사 창구와 직접 대면하지 않고 업무를 처리할 수 있는 편리성을 제공한다. 하지만 편리성 이면에 사칭 및 정보오류로 인한 사고발생시 개인과 기업에 직접적인 금전적 피해를 가져 올 수 있다. 따라서 금융 부문에서는 보다 더 안전한 암호기술에 대한 올바른 이해 및 적용이 필요하다.

제도적으로 적용되는 암호화 대상과 그 범위는 관련법에서 살펴봐야한다. 국내 전자금융 거래의 안전성 강화를 위한 대표적인 법률은 전자금융거래법이다. 암호화 적용대상은 크게 금융회사가 저장하는 금융 관련 저장 정보와 전자금융 거래 시 전송되는 전송 정보로 구분할 수 있으나, 이를 법과 규정에서 명확히 구분 짓는 것은 쉽지 않다. 전자금융 감독규정은 전자금융업의 안전성 기준 강화를 위한 내용을 중점적으로 다루고 있다. 이에 대한 주요 내용은 표 1에 간략히 기술하였다.

[표 1] 전자금융거래법과 감독규정 중 암호화 관련 내용

종류	구분 및 내용	참 고
전자금융거래 법 및 시행령	거래의 안전성과 신뢰성 확보 관련 기준 준수, 전자서명법 내 필요 기준 설정 등	제21조 (안전성 확보의무)
전자금융감독 규정	무선통신망을 통한 불법접속 방지, 사용자 인증, 암호화 등 보안대책 수립	제15조 (해킹 등 방지대책)

	암호 및 인증시스템에 적용되는 키 관리(주입·운용·갱신·폐기) 관련 절차 및 방법 마련	제31조 (암호프로그램 및 키 관리 통제)
	전자자금 이체 시 보안카드를 포함한 일회용 비밀번호를 적용, 암호화 통신, 이용자 전자적 장치에 보안 프로그램 설치 등 보안대책 적용, 전자금융거래프로그램의 위·변조 여부 등 무결성 검증 방법 제공 등	제34조 (전자금융거래 시 준수사항)
	전자서명법에 의한 공인인증서 또는 이와 동등한 수준의 안전성이 인정되는 인증방법 사용	제37조 (공인인증서 사용 기준)
전자금융감독규정 시행령	이용자 인증, 정보처리시스템 인증, 통신채널 암호화, 전자금융 거래내역의 무결성, 전자금융 거래내역의 부인방지	제7조 (인증방법의 안전성 평가 등)

이 외에도 금융회사는 저장해야하는 정보에 대해, 필요시 개인정보보호법, 정보통신망 이용촉진 및 정보보호 등에 관한 법률(약칭: 정보통신망법), 신용정보업감독규정 등의 법규를 준수하여 개인식별정보 및 주요 금융정보 등을 암호화 전송 또는 저장해야 한다. 전자금융 관련 법 이외의 법규에서 암호화 관련한 주요 내용은 표 2에 간략히 기술하였다.

[표 2] 전자금융 관련 법 이외의 법규에서 암호화 관련 내용

종류	구분 및 내용	참 고
개인정보보호법	개인정보처리자가 고유식별정보를 처리하는 경우 고유식별정보가 분실·도난·유출·변조 또는 훼손되지 아니하도록 암호화 등 안전성 확보에 필요한 조치를 할 것	제24조 (고유식별정보 처리의 제한)
개인정보보호법	개인정보처리자는 개인정보가 분실·도난·유출·변조 또는 훼손되지 아니하도록 내부	제29조 (안전조치의무)

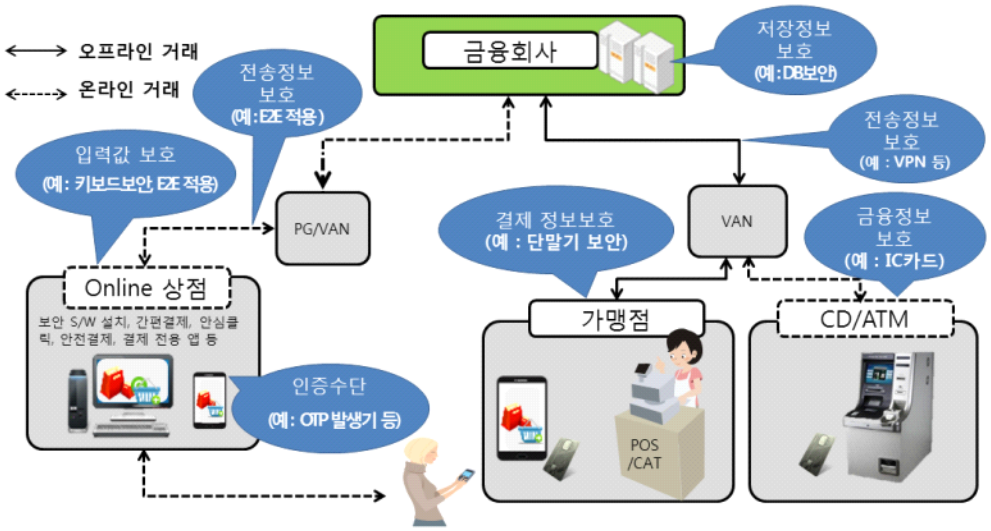
	관리계획 수립, 접속기록 보관 등 안전성 확보에 필요한 기술적·관리적 및 물리적 조치를 할 것	
신용정보업감독 규정	신용정보회사등은 개인신용정보의 암호화 등 기술적·물리적·관리적 보안대책의 구체적인 기준 마련	제20조 (기술적물리적관리적 보안대책)

금융부문의 안전한 암호기술 관련한 보안 규제 사항들을 준수하기 위해서 전자금융거래법 및 시행령, 전자금융감독 규정 등 금융 IT 보안 규제 사항과 타 분야 적용 법규, 지침 등을 관리적, 기술적, 물리적 체계의 통제항목들을 기반으로 컴플라이언스 활동이 이루어질 수도 있다. 통제항목에 대한 요약 및 관련 정보(법률 등)는 부록 2에서 참고 할 수 있도록 수록하였다.

제2절 금융부문 암호기술 적용현황

전자금융 서비스는 이용자에 의해 조작되는 스마트폰, PC, TV, 자동화기기(CD, ATM) 등 전자적 장치를 통해 금융회사가 제공하는 서비스를 의미한다. 전자금융 서비스 유형으로는 계좌 조회, 자금 이체, 주식 매매, 현금 출금, 현금 서비스, 전자결제 등 다양하다.

온라인 형태의 전자금융 거래는 모든 거래 과정이 서로 대면하지 않고 이루어지는 거래로 정의 했을 때, 자동화기기(CD, ATM)를 이용한 거래 또한 온라인 거래로 분류된다. 반면, 오프라인 형태의 금융거래는 금융회사 직원과 직접 대면하여 거래하는 형태로 볼 수 있다. 전자금융 거래는 비대면성의 특성에 의해 여러 공격 위협이 존재할 수 있으므로, 각 단계마다 거래에 대한 안전성을 보장해야 한다.



<그림 1> 금융서비스 암호기술 적용 현황

이를 위해 적용되는 암호기술을 그림 1에 간략히 나타내었다. 대표적인 적용기술을 구분해보면, 온라인상에서 이용자의 입력정보 보호(키보드 보안 등), 거래 시 추가인증 수단(OTP, 인증서 등), 전송정보 보호(종단간 암호 적용)등이 있으며 오프라인에서는 IC카드 적용(이용자 및 카드인증, 거래정보 확인 메커니즘), 온오프라인 공통적인 전송정보 보호(VPN, 통신 암호화 등), 금융회사 내부 저장정보 보호(DB보안 등)등 다양한 분야에서 암호기술이 이용되고 있다. 본 절에서는 금융부문에서 암호기술이 어떠한 목적으로 이용되며, 보안성을 어떻게 확보하고 있는지 사례별로 살펴보도록 한다.

2.1 입력정보의 암호화

이용자를 확인하기 위해 사전에 정의한 특정 정보를 입력하여 거래를 처리하는 절차는 온라인 뱅킹과 결제뿐 아니라, 오프라인 거래(PIN

입력 등)에서도 존재한다. 하지만 PC를 포함한 거래 단말기는 범용 기기이므로 정보가 쉽게 해킹에 노출될 위협이 있고 해커가 이용자로 위장할 위험이 존재한다. 따라서 입력값의 보호는 금융권 암호기술 적용에서 그 의미가 크다.

대표적인 입력값 보호 수단으로는 키보드 암호화 프로그램이 있는데, 이는 키보드를 통해 입력되는 중요 정보(아이디, 비밀번호, 계좌번호, 카드번호 등)를 암호화함으로써 키보드 해킹 프로그램 등에 의해 중요 정보가 유출되는 것을 차단하는 것이다.

국내의 경우 2013년, 정부에서 메모리 해킹사고의 재발방지를 위해 기존 종단 간(End to End: 이하 E2E) 암호화 기반 키보드 보안프로그램에 확장 E2E 암호화 기능을 추가하도록 권고[1]하였으며, 현재 금융권에서는 확장 E2E 암호화, 가상 키패드 등을 적용하고 있다.

[표 3] 초기 E2E 및 확장 E2E 보호영역

구분	E2E[2]	확장 E2E
보호영역	계좌비밀번호, 보안카드번호, 인증서 비밀번호	계좌비밀번호, 보안카드번호, 인증서 비밀번호, 거래 정보(이체금액, 입금정보)

표 3은 두 가지 E2E 암호 기술에서의 암호화 적용 보호영역을 나타낸다. 확장 E2E 암호 기술의 주 목적은 계좌 비밀번호, 보안카드번호, 인증서 비밀번호뿐만 아니라 거래 정보까지 중요 정보를 메모리해킹으로부터 보호하기 위해 원본데이터와 암호문을 함께 전송하는 것이다. 확장 E2E 암

1) 금융위원회, 메모리 해킹 관련 보도 참고 자료
http://www.fsc.go.kr/info/ntc_news_view.jsp?menu=7210100&bbsid=BBS0030&no=29612
2) 금보원, 종단 간(End-to-End) 암호화 적용 가이드, 2007.10

호 기술은 기존의 E2E의 암호화 필드에서 거래정보에 대한 암호화를 추가하여 평문의 거래정보(이체금액, 입금정보)와 암호문 형태의 거래정보를 동시에 금융회사에 전송함으로써 거래정보에 대한 변조 방지 및 무결성 검증을 위한 암호기술을 적용하고 있다.

인터넷 뱅킹에서 이용자는 이체를 하기 위해 로그인, 예비거래[3], 본거래[4], 전자서명, 거래정보 전송의 순으로 진행한다. 각 순서에서 적용되는 암호기술은 표4에 나타내었다. 로그인 순서에서 공인인증서를 이용한 로그인 시 개인키를 추출할 때 쓰는 암호기술이 적용되고, 예비거래와 본거래 순서에서는 키보드 보안I과 키보드 보안II라는 암호가 적용된다. 전자서명 순서에서는 공인인증서 개인키 추출 암호기술과 전자서명이 이뤄지고 전송 순서에서는 통신구간 암호기술이 적용된다.

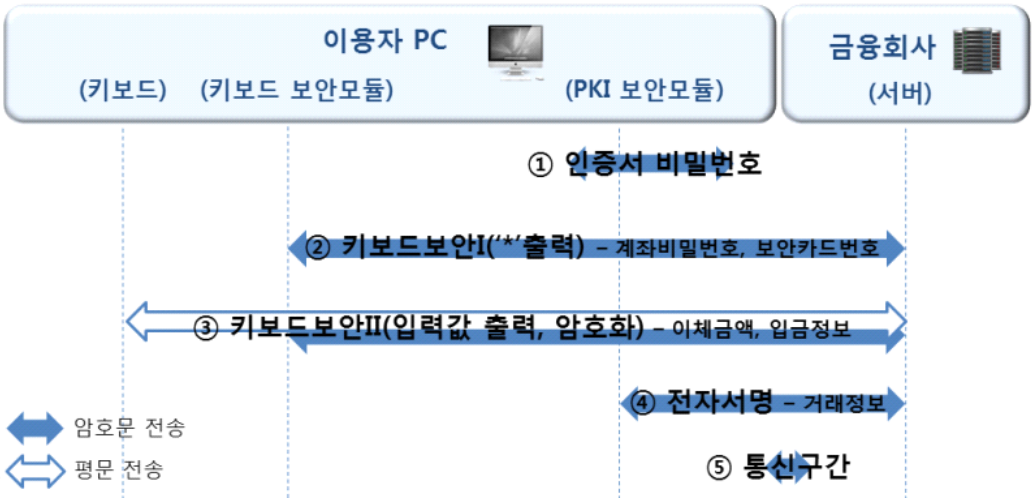
[표 4] 온라인 뱅킹 거래 시 진행 순서 및 암호적용 기술

진행 순서		로그인	예비거래	본거래	전자서명	전송
암호적용 단계	입력값					
① 공인인증서 개인키 추출	인증서 비밀번호	O			O	
② 키보드보안I ('*'출력)	계좌비밀번호 보안카드번호		O	O		
③ 키보드보안II (평문출력, 암호화)	이체금액, 입금정보		O	O		
④ 전자서명	거래내역				O	
⑤ 통신구간	통신 데이터					O

3) 온라인 뱅킹의 이체서비스는 '예비거래'와 '본거래'라고 불리는 두 가지 거래단계로 구분할 수 있는데, 두 단계 모두 금융회사 서버까지 암호화 됨, 이 때 '예비거래'는 입력된 정보를 통해 금융거래가 처리되기 전에 진행되는 거래로서, 수취인의 계좌정보가 유효한지, 예금주는 누구인지 등을 확인하는 단계임

4) '본거래'는 '예비거래'단계에서 확인된 정보로 이용자가 거래금액과 개인정보를 입력하여 거래를 진행하는 단계임

그림 2는 표 4에 구분한 5가지 암호기술 적용단계가 각각 이용자PC와 금융회사 서버 사이에서 어떤 구간에 대한 정보를 보호하는지 나타낸다.



<그림 2> 온라인 뱅킹 시 입력값 종류 및 암호화 구간

인증서 비밀번호를 입력하는 단계(그림 2, ①)는 공인인증서의 서명용 개인키를 추출 하는 단계로서, 로그인 시 이용자 인증과 전자서명에 쓰기 위한 개인키를 암호문으로부터 추출하는 단계이다. 이 때 개인키는 대칭 암호 알고리즘으로 암호화 되어 암호문 형태로 저장된다.

키보드보안I 부분(그림 2, ②)은 예비 거래, 본 거래에서 모두 동작하는데, 이용자의 계좌비밀번호나 보안카드번호(또는 OTP)와 같이 이용자 화면에서도 입력값 대신 다른 문자('*' 등)로 정보를 보호하여 금융회사 서버까지 전달된다.

키보드보안II 부분(그림 2, ③)도 키보드보안I과 마찬가지로 예비 거래와 본거래에서 모두 동작하는데, I과 II의 차이는 확장 E2E와 초기

E2E의 차이와 같다. 즉 이용자가 웹브라우저 상에서 입력하는 정보를 확인해야 하는 경우를 감안하여 키보드보안II에서 이체금액과 입금정보에 대해 평문 출력과 암호화를 모두 진행하는 것이 차이점이다.

키보드보안I, II는 입력된 값을 암호화하기 위해서는 두 종류의 암호화 방식(치환방식과 암호화 방식으로 구분)을 주로 이용한다.[5)]

전자서명 부분(그림 2, ④)은 이용자가 거래정보를 확인 후, 거래확정을 하기 위해 이용자의 개인키로 전자서명을 진행하는 단계로서 공인인증서를 이용하여 전자서명을 수행하고, 추후 이용자의 거래내용에 대해 부인방지를 제공한다.

마지막 단계는 통신구간 암호화 부분(그림 2, ⑤)으로 네트워크 단에서 수행하는 암호 프로토콜이 적용되는 부분이다. 이 부분에서는 SSL 보안 프로토콜을 이용한 통신구간 암호가 적용된다.

2.2 신용IC카드 거래

IC카드는 MS카드와 달리 카드 내에 연산이 가능한 CPU와 메모리 등으로 구성되고, 칩 고유의 일련번호 등을 가지고 있어서 카드 복제가 어렵다. 또한 거래 시 카드가 단말기 또는 발급자에 대한 자체적인 확인 절차를 진행 하여 카드의 정당성 확인이 용이하다는 특징이 있다.

금융의 목적으로 사용하는 IC카드는 그 용도에 따라 현금카드나 신용카드의 기능을 담은 응용프로그램을 칩에 탑재해야하는데, 신용카드 기능을 탑재할 경우에는, 신용카드 거래 수행이 가능하도록 EMV의 인증을 받아야 한다. EMV에서는 IC카드와 거래 단말기 간

5) 키보드보안 솔루션에서 칭하는 치환방식과 암호화 방식은 암호학에서의 현대암호(SEED 등)와 고전적인 치환형 암호를 구분하기 위함이며, 암호학적으로 볼 때에는 둘 다 암호화 방식에 속함

통신 과정 중 인증 및 중요 데이터 보호를 위해 여러 단계에서 암호 알고리즘을 사용하고 있다. 본 절에서는 IC카드에서 가장 주목 할 부분인 카드인증 절차에 이용된 암호기술에 대해 살펴본다.

❍ 카드인증(Offline Data Authentication)

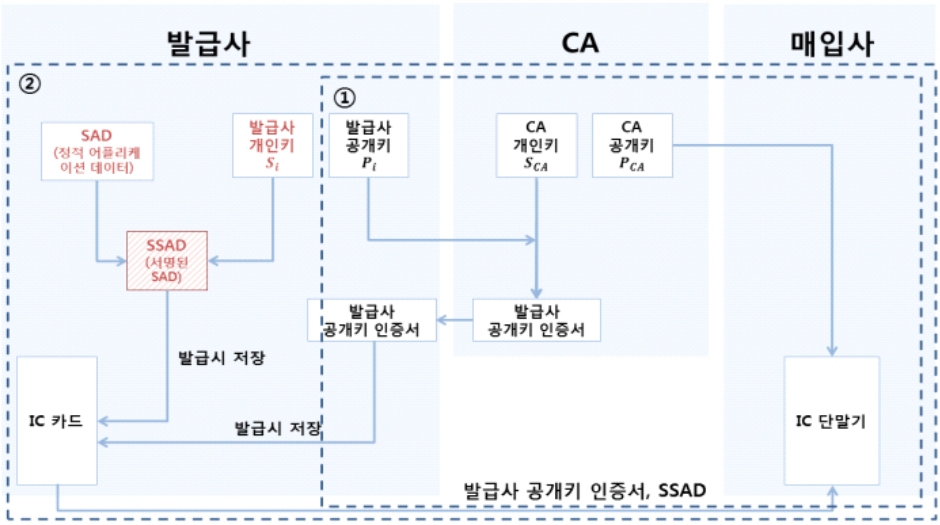
카드인증이란 거래를 진행하려는 IC카드가 정상 발급 카드인지, 발급 후 카드 내 데이터에 불법적인 변경이 있었는지를 체크함으로써 카드의 정당성을 확인하는 과정이다. EMV에서는 RSA 서명 알고리즘을 이용하여 단말기가 카드를 인증하도록 하고 있다. 카드인증 단계에서 나오는 결과는 정당한 거래 승인 확인을 위한 것으로 카드와 단말기 간의 거래 승인 및 결제절차에 직접적인 영향을 미친다.

카드인증 방식은 사용하는 암호키의 종류와 고정된 데이터의 이용 유무에 따라 SDA(Static Data Authentication)과 DDA(Dynamic Data Authentication)로 분류한다. 일반적으로 SDA에 비해 DDA의 보안 레벨이 높다.

❍ SDA(Static Data Authentication, 정적 데이터 인증)

IC카드에는 발급 시 카드인증을 위해 두 가지 정보를 저장한다. 이는 카드 내 고정된 데이터인 정적 어플리케이션 데이터(SAD, Static Application Data)를 발급사의 개인키로 서명한 SSAD값과 발급사의 공개키를 CA의 개인키로 서명한 발급사 공개키 인증서이다.

카드 인증 시에는 카드에 발급되어 있는 데이터 중에서 항상 지정된 데이터만을 사용하여 인증한다. 그림 3은 SDA 수행과정을 나타내며 IC 단말기에서 수행하는 SDA 상세 과정은 아래 박스와 같다.

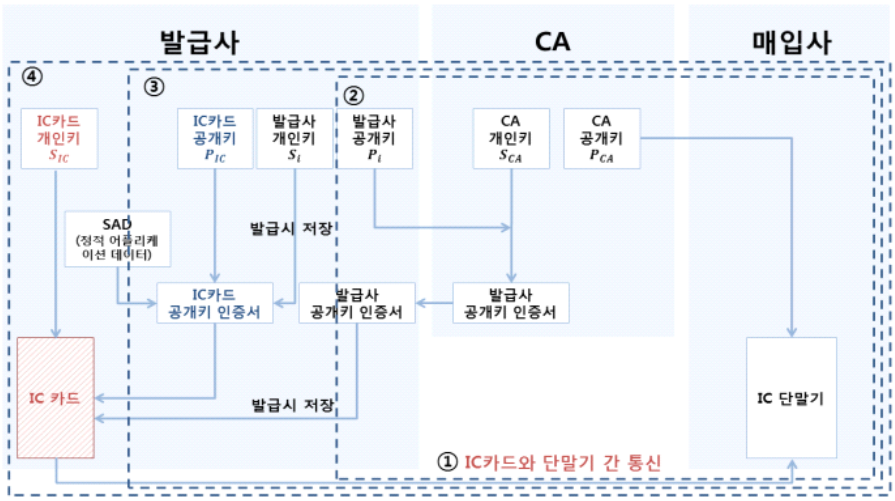


<그림 3> SDA 수행과정

① IC 단말기는 CA 공개키(P_{CA})를 통해, IC카드에 저장되어있는 발급사 공개키 인증서를 검증하여 발급사의 공개키(P_i)가 정당한지 확인한다.
② 정당할 경우, 발급사 공개키 인증서에서 발급사 공개키(P_i)를 추출하여 IC카드에 저장된 SSAD값을 검증한다. 즉, 카드 내 고정된 정적 어플리케이션 데이터(SAD)가 맞는지 확인한다.

❍ DDA(Dynamic Data Authentication, 동적 데이터 인증)

DDA는 SDA절차를 포함하고 있으며, 매 인증 시 IC카드 내부에서 동적으로 생성한 동적 어플리케이션 데이터(DAD, Dynamic Application Data)를 추가로 사용한다. DDA의 실행을 위해서는 IC카드에 암호 계산을 위한 보조 프로세서가 탑재되어 있어야한다. 그림 4는 DDA 수행과정을 나타낸다.



<그림 4> DDA 수행과정

IC카드에는 발급 시 두 개의 인증서가 저장된다. 이는 정적 어플리케이션 데이터와 IC카드 공개키 데이터를 발급사 개인키로 서명한 IC카드 공개키 인증서와 발급사의 공개키를 CA 개인키로 서명한 발급사 공개키 인증서이다. IC 단말기에서 수행하는 DDA 상세과정은 아래와 같다.

① IC 단말기는 IC 카드에 임의로 생성한 동적 난수를 전송하고 IC 카드는 이 값을 IC카드 개인키(S_{IC})를 통해 서명하여 카드단말에 전달한다.
② IC 단말기는 CA 공개키(P_{CA})를 통해, IC카드에 저장되어있는 발급사 공개키 인증서를 검증하여 발급사의 공개키(P_i)가 정당한지 확인한다.
③ 정당할 경우, 발급사 공개키 인증서에서 추출된 공개키(P_i)로 IC카드 공개키 인증서를 검증하여 IC카드 공개키(P_{IC})가 정당한지 확인한다.
④ 정당할 경우, 검증된 IC카드 공개키 인증서에서 추출된 IC카드 공개키(P_{IC})로 카드에서 전송한 서명된 SAD와 ①에서 전송된 서명된 DAD값(S_{IC}로 서명)을 검증한다. 즉, 카드에서 전송한 값의 진위 여부를 확인한다.

2.3 현금IC카드 및 금융자동화 기기 거래

국내 금융자동화기기는 2014년 2월 이후 마그네틱 카드로 이용할 수 없고 카드 앞면에 IC칩을 가진 IC카드로 서비스를 이용해야 한다. 이는 마그네틱 카드 불법 복제로 인한 사고를 줄이기 위함이며, 현금IC카드는 은행 간 금융자동화기기(CD, ATM) 공동 사용을 위해 한국은행이 개발한 「금융IC표준」을 준수하고 있다.

❍ 현금IC카드 발급 시 적용되는 암호기법

현금IC카드는 카드키 주입, 발급단계와 카드 인증, 키 갱신 단계를 거쳐 조회, 거래에 이용된다. 카드키 주입, 발급 단계에서 현금카드에 사용되는 키는 금융공동망 키를 통해 발행기관이 관리하며, 이 키는 카드 제조 후 배포 시 탑재 되어 있다. 상세 절차는 아래와 같다.

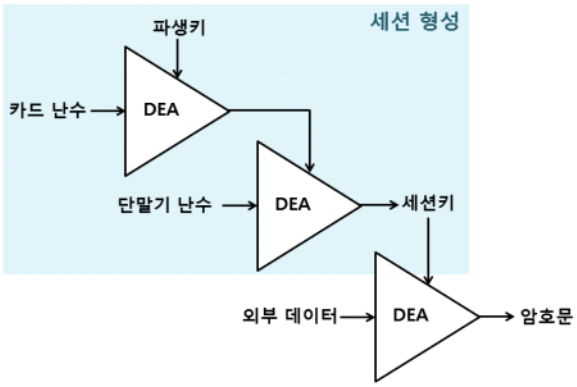
① 금융IC카드 발행 시 금융공동망 애플릿을 카드에 다운받고, 이용자 요구에 따라 계좌 개수 등을 설정한다.
② 카드일련번호(CSN)을 카드에 설정한다.
③ CSN과 금융공동망의 유도키(DK)를 금융공동망키로 암호화하고 메시지 인증코드(MAC) 값과 함께 IC카드에 저장한다.
④ 1차 카드정보 설정 완료를 위해 카드를 리셋한다.
⑤ 카드의 사용처 등을 설정하고 이용자의 계좌정보 등을 입력한다.

카드 인증 및 키 갱신 단계에서는 서명 인증서와 키 분배용 인증서를 통해 IC카드의 발급처 등을 인증하는데, 이 때 공개키 기반 인증서를 이용하여 처리하고 IC카드 내에 키와 PIN을 갱신하는 경우에는 아래와 같이 암호화 하여 저장한다.

[표 5] IC카드에 PIN, 키 갱신 시 데이터 암호화 방법

구분	표기	설명
키 암호화	Ciphered Data = CBC(Key, DK)	유도키(DK)를 가지고 갱신 대상 키값을 CBC모드로 암호화
PIN 암호화	Ciphered Data = CBC(PIN ‖ Padding DATA, DK)	PIN값에 패딩처리 후 유도키(DK)를 가지고 패딩 처리된 PIN값을 CBC모드로 암호화

카드 발급 후 현금IC카드로 CD, ATM기에서 서비스를 이용할 때 조회, 거래 처리절차 중 암호기술이 적용되는 부분은 통신 세션을 맺는 부분이다.



<그림 5> 외부 데이터(비밀번호 등) 암호화

세션은 IC카드에서 중요 데이터(비밀번호 등)를 전송할 때 맺으며 그림 5의 방식으로 세션키를 생성하여 외부 데이터를 암호화 한다. DEA(Data Encryption Algorithm)로는 SEED 암호 알고리즘을 사용하며 세션을 형성한 후에는 전송되는 정보의 보호를 위해 MAC값으로 인증하는 절차를 수행한다.

❍ 금융자동화 기기에서 적용되는 암호기법

CD, ATM 기기에서 안전한 IC카드 이용 및 거래를 위해 적용된 암호기법은 인증, 조회, 거래 업무 서비스에서 서로 유사하다. 현금IC카드를 이용한 금융자동화기기 조회 및 거래 절차는 아래와 같다.

① 거래를 위한 초기화 과정: 카드 리셋, 거래 관련 셋팅, 칩 일련번호 읽기
② 계좌정보 파일을 읽음
  A. 계좌 개수가 1일 경우 CD, ATM 화면에서 서비스 선택(현금지급, 계좌이체 등), 금액 및 비밀번호를 입력할 수 있게 함
  B. 계좌 개수가 2 이상일 경우 PIN을 요구하고 PIN을 검증(PIN 실패 시 오류처리, PIN 재시도 횟수는 은행이 선택)한 후, CD, ATM화면에 계좌를 나열하고, 서비스 선택(현금지급, 계좌이체 등)과 금액 및 비밀번호를 입력할 수 있게 함
③ 계좌번호 암호화 명령어와 터미널 난수, 비밀번호를 카드로 전송함
④ IC카드는 IC카드 내 저장된 암호화 알고리즘과 명령어를 통해 데이터를 암호화 하고, 암호화된 데이터와 카드에서 발생된 난수를 CD, ATM으로 전송함
⑤ CD, ATM기는 카드난수, 터미널난수, 암호화된 데이터 등으로 구성된 공동망 전문을 취급은행으로 전송, 취급 은행은 동 전문을 금융공동망센터로 전송, 금융공동망센터는 동 전문을 개설 은행으로 전송
⑥ 개설 은행은 암호화된 데이터를 복호화함
⑦ 원장 조회 및 승인 여부를 확인하여 승인, 거절코드를 전송

2.4 전자서명 및 인증수단

전자금융 거래법에서는 거래내용의 진실성과 정확성을 확보하기 위해 전자서명을 수행하고 이 때 서명자를 인증하기 위한 접근매체로 공인인증서를 이용하고 있다. 표 6은 접근매체와 인증수단에 대한 설명이다.

[표 6] 전자금융거래법에서의 접근매체와 인증수단

구분	접근매체(전자금융거래법)	인증수단
설명	전자금융거래에 있어서 거래지시를 하거나 이용자 및 거래내용의 진실성과 정확성을 확보하기 위하여 사용되는 수단 또는 정보	전자금융거래의 당사자 인증에 사용되는 유형, 무형의 수단 및 인증 절차를 처리하기 위한 기반 시설
예시	이용자정보, 생체정보, 공인인증서, 신용카드 등	생체정보 입력장치, HSM, 보안카드, OTP발생기 등

❍ 전자서명

금융권에서는 전자금융거래에 대한 본인인증 및 부인방지의 목적으로 전자서명 기술을 많이 사용한다. 계약을 처리 할 때 인감도장을 동사무소 등과 같은 공공기관에 등록하여 공증을 받아 계약서 등의 날인에 사용하듯이 전자서명도 인감도장과 동일한 역할을 한다.

전자서명은 대부분 원본 문서의 해시를 계산하여, 전자서명을 하고 원본 문서의 진위여부를 판단한다. 이 때 원본 문서는 암호화 되는 것이 아니므로 누구나 읽을 수 있다. 즉, 전자서명을 통해 원본 문서에 대한 기밀성을 보장할 수는 없다. 단, 전자서명을 통해 위조불가, 인증, 재사용 불가, 변경 불가, 부인방지 등의 기능을 제공한다. 이에 대한 설명은 표 7과 같다.

[표 7] 전자서명이 제공하는 기능 및 설명

제공 기능	설 명
위조불가	서명자만이 서명문을 생성할 수 있다.
인증	서명문의 서명자를 확인할 수 있다.

재사용 불가	서명문의 해시값을 전자서명에 이용하므로 한 번 생성된 서명을 다른 문서의 서명으로 사용할 수 없다.
변경불가	서명된 문서는 내용을 변경할 수 없기 때문에 데이터가 변조되지 않았음을 보장하는 무결성을 만족한다.
부인방지	서명자가 나중에 서명한 사실을 부인할 수 없다.

전자서명의 생성과 검증의 과정은 그림 6과 같다.

<그림 6> 전자서명 생성과정 및 검증과정

전자서명은 서명키(서명자의 개인키)로 암호화하는 전자서명 생성과정과 서명값을 서명자의 공개키로 복호화하는 검증 과정으로 나뉜다.

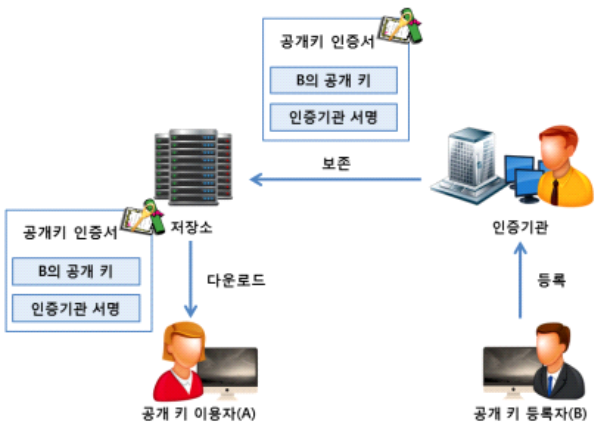
[표 8] 전자서명 생성 및 검증 절차

	생성과정	검증과정
설명	원본 메시지의 해시값에 대한 전자서명을 계산한다.	원본 메시지의 해시계산 후, 전달받은 서명값을 검증한다.
절차	① 원본 메시지의 해시를 계산한다. ② 계산한 해시값을 서명자의 개인키로 암호화 하여 전자서명을 계산한다.	① 전달받은 원본 메시지를 서명값을 생성할 때와 동일한 해시함수로 해시를 계산한다. ② 전달받은 전자서명값에 대해 서명

		자의 공개키로 복호화 하여 검증 값의 원본 데이터를 계산한다. ③ 계산을 통해 얻은 해시값과 검증으로 얻은 원본 데이터의 값이 동일한지 확인한다.

전자서명된 문서에 서명 시 사용한 공개키는 인증기관에서 등록 및 관리하기 때문에, 누구나 사용자의 공개키에 접근이 가능하고 서명값 검증 시에 이를 사용할 수 있다.

한편, 전자서명을 수행하기 위해서는 누구나 사용자의 공개키에 접근이 가능하고 공개키 값을 확인 할 수 있도록 하기 위한 공개키 공증 문서(인증서)와 이를 인증하는 인증기관(CA, Certificate Authorities)이 필요하다. 공개키 공증 문서인 인증서는 온라인 뱅킹이나 일정 금액 이상의 온라인 결제, IC카드 인증 등 다양한 분야에서 이용되고 있고 이에 대한 인증기관은 각 관련 금융회사에서 관여한다. CA와 함께 필요한 지원 매커니즘으로 형성되는 전체 시스템을 공개키 기반 구조(PKI, Public-Key Infrastructure)라 한다.



<그림 7> PKI 구성 요소

PKI를 설정하고 운영하는 것은 복잡하다. 인증서 발급을 위한 사용자의 신원을 확인하는 것과 신뢰할 수 있는 CA의 키 분배와 같은 이슈가 해결되어야 한다. 국내의 경우 최상위 인증기관은 한국인터넷진흥원에서 담당하고 있으며, 개인이 금융권에서 이용하는 공인인증서는 등록대행기관인 은행, 상호금융회사, 우체국 또는 증권회사 등에서 온라인 계좌 개설 후 무료로 발급받는다.

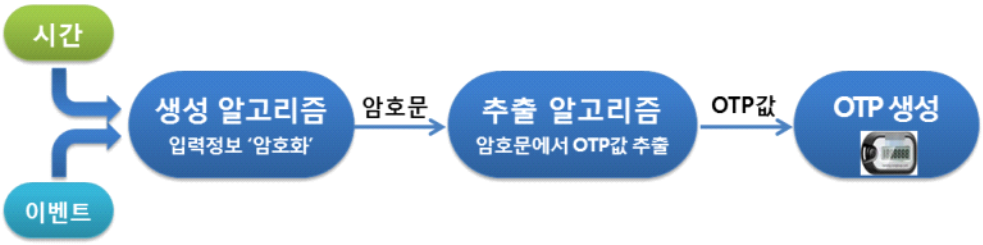
<그림 8> 공인인증서 관리 체계

전자서명을 검증할 때에 이용할 공개키에 대한 공증 문서(인증서)와 공증기관(CA)에 대해 간략히 설명한다. 금융권에서 공인인증서를 이용하여 전자서명을 진행할 경우, 그림 8과 같이 관리된다.

❍ 인증수단(OTP 발생기)

인증수단의 대표적인 것은 OTP(One Time Password)발생기이다. OTP 발생기는 고정된 비밀번호 대신 매번 일회용 비밀번호를 생성하는 기기로서, 인터넷, 휴대폰, 전화 등의 다양한 매체를 이용하여 은행, 증권, 선물사의 전자금융거래 시에 인증수단으로 이용되고 있다. OTP는 재사용이 불가능하여 사용자의 비밀정보를 유추할 수 없는 특징을 가진다.

OTP 값은 그림 9와 같이 생성된다.



<그림 9> OTP 생성단계

입력값은 시각 정보, 이벤트 정보 등으로부터 수집 된 난수를 의미한다. 이 입력값은 서버와 OTP기기 간 동기화 방식을 구분하며 크게 4가지(질의응답, 시간 동기화, 이벤트 동기화, 조합) 방식이 있다. 이 때 입력된 정보는 난수 그대로 사용하거나 해시, 암호화 등의 방법으로 추가적인 난수 생성 과정을 거쳐 변환된 난수를 사용할 수 있다.

생성 알고리즘은 일방향 해시함수와 대칭키 블록 암호 알고리즘 기반으로 구성되어 있으며, 암호문의 유추 가능성에 대응할 수 있어야 한다. 현재 이용되는 생성 알고리즘은 AES_CBC(128비트 키), AES_CBC(256비트 키), SEED_CBC(128비트 키), HMAC_SHA256(256비트 키)이다.

추출 알고리즘은 생성알고리즘을 통해 나온 연계정보를 암호화한 데이터 값을 일회용 비밀번호 추출 알고리즘에 입력하여 특정 규칙으로 일회용 비밀번호를 추출한다.

제3장 금융부문 암호기술 활용 시 고려사항

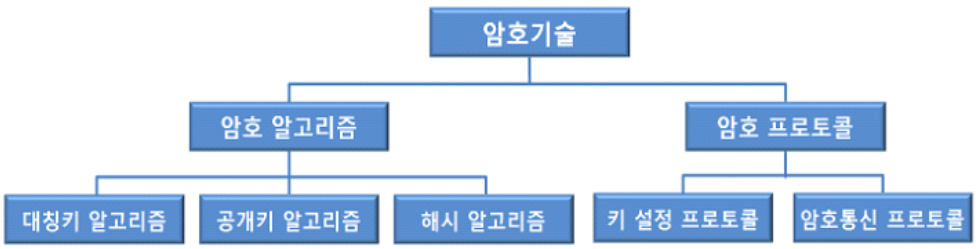
제3장 금융부문 암호기술 활용 시 고려사항

제1절 암호기술 운영 시 고려사항

금융거래 시 불가피하게 저장해야하는 거래정보, 개인정보 등은 사이버 공간에서 이용자를 대신하거나 실제 금전적인 가치를 대신할 수 있기 때문에 정보 처리 및 저장 시 데이터 보호가 중요하다. 데이터 보호기술의 대표적인 것은 암호기술이며, 금융권에서는 암호기술을 도입하여 안전한 서비스가 이루어 질 수 있도록 시스템을 구축하고 있다.

1.1 암호기술 분류 및 용도

정보를 안전하게 처리하기 위해 암호기술의 보안 강도를 최고로 높여 구축하는 것은 현실적이지 않다. 보안과 비용(암호 처리에 필요한 메모리, 처리 속도 및 성능향상을 위한 고성능 장비 도입 등)을 함께 고려해야하기 때문이다. 따라서 암호기술의 구성 시 암호화 대상과 목적을 정확히 파악하고 이를 고려하여 최적의 시스템을 구축해야 한다.



<그림 10> 암호기술 분류

암호기술은 암호 알고리즘과 암호 프로토콜[6]로 구분한다. 세부적으로는 암호 알고리즘은 대칭키, 공개키, 해시 알고리즘으로 분류하며, 암호 프로토콜은 키 구축과 암호 프로토콜로 분류한다.

금융권에서는 도청, 메시지 변경, 위장, 부인과 같은 위협에 대응하기 위해 암호기술을 이용하고 있으며, 이는 표 9에 나타내었다.

[표 9] 암호기술의 주요 보안 특성과 금융권 적용 예시

주요 보안 특성 (보안 위협)	**암호기술**	**금융권 적용 예시**
기밀성(도청)	암호화	종단 간 암호화, 입력값 보호 등
무결성 (메시지 변경)	일방향 해시함수	전자서명
	메시지 인증코드	거래전문 확인
	전자서명	인증서
인증(위장)	메시지 인증코드	거래전문 확인
	전자서명	본인인증
부인방지(부인)	전자서명	본인인증
기타	키 설정 프로토콜	세션키 생성
	암호통신 프로토콜	TLS, IPSec

본 절에서는 암호기술의 목적인 데이터 기밀성, 무결성, 사용자 인증 및 부인방지의 특성과 이와 관련된 금융권 적용 예시를 살펴본다.

❍ 기밀성을 위한 암호기술

기밀성이란, 비인가 대상이 데이터에 접근하여 내용을 파악할 수

6) 프로토콜이란 양자 간 암호화 환경을 운영하기 위해 주고받는 정보에 대한 상호 규칙을 뜻함

없음을 보장하는 것이다. 대표적으로 기밀성을 유지하기 위해 적용된 암호기술로는 대칭키, 공개키 등의 암호기술이 있다. 적용 대상을 기준으로 분류하면 접속 구간의 기밀성, 내용 기밀성, 메시지 흐름 기밀성 등이 있다.

- 금융권 적용 예시: 종단 간 암호화, 입력값 보호 등

○ 무결성을 위한 암호기술

무결성이란 원래의 정보 또는 신호가 전송, 저장, 변환 시 또는 그 후에도 동일함을 유지하는 것이다. 정보보호 분야에서는 메시지가 제3자 등에 의해 임의로 변경되는 것을 방지하기 위해 데이터 전송 에러 확인 또는 해시함수 등의 일방향 함수를 활용한다. 무결성이 깨지는 경우는 사고나 의도적인 변경 또는 삭제, 컴퓨터 바이러스 등에 의한 손상 등이 있으며, 이를 해결하기 위해 이용되는 암호기술로는 일방향 해시함수, 메시지 인증코드, 전자서명이 있다.

- 금융권 적용 예시: 거래 전문 확인 등

○ 인증을 위한 암호기술

인증이란 정보 및 시스템의 자원을 사용하는 주체가 정당한 사용자임을 확인할 수 있도록 보호하는 서비스로서, 사용자가 자신의 신원정보를 밝히고 확인하는 식별과 신원정보 또는 신원확인의 유효성을 확립하는 인증의 단계로 구분할 수 있다. 대상을 기준으로 구분할 경우에는 연결된 송·수신자를 확인하기 위한 사용자 인증과 교환되는 정보마다 진위를 확인하기 위한 메시지 인증이 있다.

- 금융권 적용 예시: 전자서명(본인 인증) 등

❍ 부인방지를 위한 암호기술

부인방지란, 송신자나 수신자 양측이 메시지를 전송한 사실 자체를 부인하지 못하도록 막는 것을 말한다. 따라서 어떤 메시지가 송신되었을 때 수신자는 그 메시지가 실제로 송신자라고 주장하는 주체로부터 송신되었음을 확신 할 수 있다. 마찬가지로 메시지가 수신되었을 때 송신자는 그 메시지가 실제로 수신자라고 주장하는 주체에 의해서 수신되었음을 확신할 수 있다.

- 금융권 적용 예시: 전자서명(거래에 대한 서명) 등

1.2 암호기술 운영 시 고려사항

보안 시스템은 구성 요소들 중 보안에 가장 취약한 부분에 의해 전체 시스템의 보안 강도가 결정된다. 따라서 안전한 암호기술 운영 시 보안 강도가 높은 알고리즘의 선택과 더불어 구현 시의 유의사항 및 인적, 물리적 관리 정책 등을 수립하고 준수해야 한다.

❍ 암호정책 수립

기업에서 암호기술을 운영하기 위해서는 사전에 암호정책을 수립하고 역할을 분담하여 철저히 관리되어야 한다. 표 10은 암호정책 수립 시 관련 분야 업무 관련자들과 책임사항[7]을 나타낸다.

7) 조직의 정보보호 정책 수립 가이드, TTAK.KO-12.0093, 2008.12, TTA.

[표 10] 암호정책 수립 시 책임사항

업무 관련자	책임사항
정보보호관리자	- 사내의 전반적인 보안 계획을 수립 관리하는 역할 - 사내에서 한명을 선정하여, 암호화 기술 및 프로그램 등 암호와 관련된 모든 사항들에 대해서 최종 승인과 총괄적인 관리를 담당함 - 기술 환경의 발달에 따라 암호기술 및 프로그램, 암호키 등의 기준을 검토하고 개정할 의무를 가짐
정보보호담당자	- 각 부서의 정보보호를 담당하는 역할 - 부서별로 한명씩 선정하여, 암호기술 및 프로그램의 도입 또는 개발과 암호키 관리 등을 담당함
사용자	- 암호기술 및 프로그램을 실제로 사용하는 자 - 정보를 다루는 사내의 모든 직원이 이에 해당되며, 정보보호 관리자가 승인한 암호기술 및 프로그램만을 사용함

❍ 다른 암호기술 조합

서로 다른 암호기술을 조합하여 적용할 경우 서로 보안 강도의 차이가 큰 알고리즘을 조합하는 것은 권장하지 않는다. 다만 성능이나 상호 운용성 등의 이유로 상이한 보안 강도의 알고리즘과 키 크기를 사용하게 될 경우에는 가장 약한 보안 강도의 알고리즘이 전체 보안 강도 결정한다는 것에 유의하여야 한다. 특히 키 분배와 데이터 암호를 위해 대칭키 암호 알고리즘과 공개키 암호 알고리즘을 함께 사용할 경우, 보안 강도 고려하여 적절하게 조합해서 사용 할 수 있도록 하여야한다.

❍ 시스템 예상 수명

관리자는 시스템의 예상 수명을 고려하여, 전체 운영 기간 동안

안전할 것으로 예상되는 알고리즘을 선택하거나 환경에 따라 언제든지 해당 알고리즘과 키 크기를 갱신 할 수 있는지 확인해야 한다.

○ 기타

암호 시스템을 운영할 때에는 암호기술이 제 역할을 하기 위해 안전한 난수 생성기의 이용, 안전한 키 관리, 안전한 프로토콜 설계, 안전한 운영모드 선택 등을 함께 고려하여야 하는데, 이에 대한 세부적인 내용은 3장의 각 해당 절에서 상세히 다룬다.

제2절 암호 알고리즘 선택 시 고려사항

암호기술 적용의 기본 가정은 공개된 암호 알고리즘과 비공개된 암호 키로 보호하고자 하는 정보를 안전하게 지키는 것이다. 이 때 공개된 암호 알고리즘은 국내외 권고 표준 암호 알고리즘을 선택해야 한다.

본 절에서는 암호기술 이용 시 선택할 수 있는 표준 암호 알고리즘을 구분하고 그 특징과 보안 강도 및 안전성 유지 기간에 대해 살펴보도록 한다.

2.1 암호 알고리즘 종류 및 특징

암호 알고리즘은 대칭키, 공개키, 해시함수로 구분할 수 있다. 표 11은 각 알고리즘의 종류와 특징, 응용분야에 대해 설명한다.

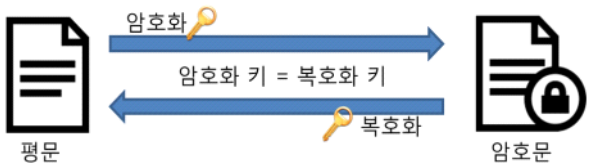
[표 11] 알고리즘의 종류 및 특징

종류	특징	대표적인 알고리즘	응용분야	구체적 예시
대칭키 암호 알고리즘	- 암복호 동일키 사용 - 기밀성 유지 통신 가능 - 키 분배 문제 해결이 필요	AES, SEED(블록)	대부분의 개인정보 대상 암호화 적용	인터넷 뱅킹 조회, 공인인증서 개인키 저장 등
		RC4(스트림)	무선 통신	음성, 데이터 정보 암호
공개키 암호 알고리즘	- 암복호 서로 다른 키 사용 - 키 분배 문제해결 가능 - 대칭키 암호 대비 속도 느림 - 중간자 공격에 약함	RSA (인수분해)	공인인증서, 키 교환	인터넷 뱅킹, SSL
		DSA, DH* (이산대수)	키 교환	SSL 보안프로토콜 키 설정 알고리즘
		ECDSA, ECDH* (타원곡선 이산대수)		
해시 함수	- 키 없음 - 일방향 함수 - 해시함수 충돌	SHA256	비밀번호 검사	아이디, 비밀번호 기반 로그인 인증 수행
기타 (조합)	- 대칭키 암호와 공개키 암호 함께 사용 - 해시함수와 다른 암호 함께 사용 등	HMAC-SHA 256, HMAC_DRBG 등	메시지 축약값을 만들기 위한 MAC, OTP 생성용 의사난수 생성기 등	보안API 및 SSL 프로토콜 전송

* 안전한 키 교환을 위한 공개키 암호 방식의 키 교환 알고리즘

❍ 대칭키 암호기술

송수신자가 서로 동일한 키를 이용하는 암호기술로서 다량의 데이터를 암호화 할 때 주로 이용된다. 암호화 데이터 단위를 기준으로 블록암호 알고리즘과 스트림 암호 알고리즘으로 구분할 수 있다. 블록암호 알고리즘은 데이터를 블록단위로 암호화하여 고정된 크기로 블록을 나누어 처리하고, 스트림 암호 알고리즘은 의사난수를 연속적(스트림)으로 생성하여 비트 또는 바이트 단위로 처리하는 방식이다.

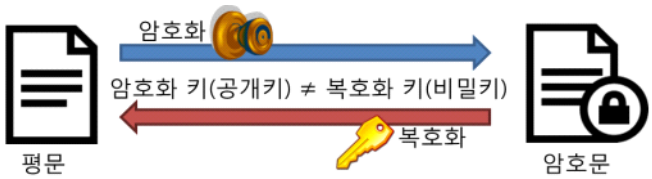


<그림 11> 대칭키 암호화 방식

금융권에서 사용되는 대표적인 대칭키 암호 알고리즘으로는 블록암호 알고리즘인 SEED가 있으며 그 외에 AES, ARIA 등이 있다.

❍ 공개키 암호기술

공개키 암호기술은 대칭키와 달리 각 사람마다 한 쌍의 키(공개키, 개인키)를 가진다. 이 때 공개키는 암호화하는데 사용되며 모든 사람에게 공개되고, 개인키는 복호화하는데 사용되며 공개되지 않는 비밀키이다. 공개키 암호 개념은 비밀키의 사전 분배 문제를 해결하고, 전자서명 개념을 가능하게 했다.



<그림 12> 공개키 암호화 방식

그러나 공개키 암호는 암호화에 많은 시간이 소요되고 계산량이 많기 때문에 일반적으로 대용량의 데이터를 암호화하는 데에는 사용되지 않고 길이가 짧은 데이터(신용카드번호나 개인식별번호 등)를 암호화하는데 사용된다. 공개키 암호 방식 알고리즘과 그 용도는 표 12와 같다.

[표 12] 공개키 암호 방식 알고리즘과 그 용도

공개키 암호 방식 알고리즘	용도		
	암복호	전자서명	키 교환
RSA	O	O	O
DH(디피헬만)	X	X	O
DSA (전자서명알고리즘)	X	O	X
ECC (타원곡선암호체계)	O	O	O

일반적인 공개키 암호 알고리즘은 인수분해 문제를 빠른 시간 내에 풀 수 없다는 것에 기반한다. 하지만 양자 컴퓨터가 상용화 될 경우, 양자 컴퓨팅으로 빠른 시간 내에 인수분해 문제를 풀 수 있기 때문에 공개키 암호의 비밀키가 유출 될 수 있다. 이를 대비하기 위해서는 양자컴퓨터의 암호해독 시도에도 안전한 공개키 암호 알고리즘이 필요하며 이러한 암호를

‘포스트 양자암호’라고 한다. ‘포스트 양자암호’에 대한 자세한 내용은 ‘[부록 4] 새로운 암호 알고리즘’에서 참조할 수 있다.

○ 해시함수

해시함수는 임의의 길이를 가진 메시지를 고정된 크기의 메시지로 출력하는 일방향 함수로서, 데이터를 압축시키는 함수이다. 해시함수는 어떠한 크기의 메시지를 입력 받아도 고정된 길이의 출력을 생성하기 때문에 파일이나 메시지, 데이터의 무결성을 검증할 수 있는 지문(fingerprint)을 생성할 수 있다. 또한 해시값으로부터 이전 평문을 복원할 수 없는 일방향 함수의 성질을 이용하여 메시지 인증이나 전자서명에 사용할 수 있다.

<그림 13> 해시함수(일방향 함수)의 성질

○ 기타(암호 알고리즘 조합)

암호기술을 적용할 때에는 대칭키, 공개키, 해시함수 등의 알고리즘 특징을 고려하여 용도를 구분하고 이를 조합하여 이용하는 경우가 대부분이다. 대표적인 조합으로는 대칭키와 공개키 암호의 조합으로 키 정보 공유를 위해 공개키 암호를 쓰고 그 후 키 갱신이나 전문 전송 시에는 속도가 빠른 대칭키 암호를 쓰는 방법이 있다. 그 외에 전자서명, 인증서, 메시지 인증코드, 의사난수 생성기에서도 암호 알고리즘을 조합하여 이용되고 있다.

[표 13] 대표적인 암호 알고리즘의 조합

구분	조합된 암호 알고리즘	금융권 이용분야
하이브리드 암호 시스템	대칭키 암호, 공개키 암호	온라인 뱅킹 통신 구간 암호
		스마트뱅킹 가상키패드 및 앱 무결성
전자서명	일방향 해시함수, 공개키 암호	거래내용 전자서명
인증서	일방향 해시함수, 대칭키 암호	본인인증, 부인방지
메시지 인증코드 (MAC)	해시함수와 키를 조합, 대칭키 암호로 생성	로그 위변조방지
		증권 트레이딩 시스템
의사난수 생성기	대칭키 암호, 해시함수	구간 암호화용 대칭키 생성
		암호키 생성

2.2 보안 강도

암호학에서 암호 알고리즘의 취약성을 찾아내는데 필요한 계산량을 의미한다. 예를 들어, 보안 강도가 80비트라는 것은 2^{80}번의 계산을 해야 암호 알고리즘의 취약성을 알아낼 수 있다는 것을 뜻한다.

각 보안 강도에 해당하는 대표적인 알고리즘은 ‘[부록1] 금융권 암호기술 적용 시 준수규격’에서 다룬다.

❍ 키 길이와 보안 강도의 관계

키 길이와 보안 강도의 관계는 표 14의 대칭키 암호 알고리즘의 키 길이를 기준으로 키를 찾기 위한 전수조사를 수행할 때 소요되는 시간[8]을 통해 알 수 있다.

[표 14] 대칭키 암호기준, 키 길이에 따른 해독 시간 (단위: 비트)

키 길이	키 공간	해독 시간
56~64	2^{56} ~ 2^{64}	단기: 수 시간 또는 수십 일
112~128	2^{112} ~ 2^{128}	장기: 양자 컴퓨터가 없는 경우 수십 년
256	2^{256}	장기: 현재 알려진 양자 컴퓨팅 알고리즘을 운영하는 양자 컴퓨터가 있어도 수십 년

예를 들어 키 길이가 56비트 크기로 표현 할 수 있는 키의 개수는 2^{56} 개이고 112비트로 표현되는 키의 개수는 2^{112} 개다. 키 길이가 56비트에서 112비트로 2배 늘었을 때 대입해야하는 키의 경우의 수는 2^{56} 배 늘어나며, 해독 시간도 전수 조사에 소요되는 시간이므로 2^{56} 배 늘어난다. 즉, 키 길이가 길수록 해독 시간이 늘어나므로 보안 강도가 높아지게 된다.

❍ NIST 권고 암호 알고리즘

국내·외 암호 알고리즘 권고 기관에서는 보안 강도별로 권고하는 알고리즘을 선정하여 제시하는데, 우리나라는 NIST의 권고 기준을 준수한다. NIST에서 권고하는 암호 알고리즘의 보안 강도는 대칭키 암호 알고리즘의 키 길이를 기준으로 5가지(80[9], 112, 128, 192, 256비트)

8) 암호기술의 이해, 암호분석, 2013.3, 도서출판 그린
9) 80비트의 보안 강도란(=1.2 x 1024)번의 계산을 해야 암호키 또는 암호 알고리즘의 취약성을 찾아낼 수 있음을 의미

단계로 나누어져 있으며, 5가지 단계에 따른 해시함수와 공개키 알고리즘 키 길이를 함께 권고한다. 표 15는 NIST가 권고하는 알고리즘별 보안 강도[10]이다.

[표 15] 암호 알고리즘별 보안 강도 비교 (단위: 비트)

보안 강도	대칭키 암호 알고리즘 (최소 키 길이)	공개키 암호 알고리즘				해시 함수[11] (보안 강도)
		인수분해 (최소 키 길이)	이산대수		타원곡선 암호 (최소 키 길이)	
			공개키 (최소 키 길이)	개인키 (최소 키 길이)		
80	80	1024	1024	160	160	80
112	112	2048	2048	224	224	112
128	128	3072	3072	256	256	128
192	192	7680	7680	384	384	192
256	256	15360	15360	512	512	256

표 15에서 첫 번째 열은 NIST 기준의 보안 강도를 나타내고 두 번째 열은 첫 번째 열에 표시된 보안 수준을 만족하는 대칭키 암호 알고리즘의 보안 강도를 나타낸다. 세 번째부터 여섯 번째 열까지는 보안 강도를 공개키 암호 알고리즘의 최소 키 길이를 나타내며, 일곱 번째 열에서는 해시함수의 보안 강도를 확인할 수 있는데, 해시함수는 해시함수의 취약성[12]을 찾아내기 위해 필요한 연산량을 기준으로 보안 강도를 정한다.

10) NIST SP 800-57 (Recommendation for Key Management-Part1: General), Rev.4, 2016.1
11) 해시함수의 보안 강도는 출력비트 길이가 아니며 해시함수의 보안 강도는 부록1 참고
12) 해시함수의 안정성을 위한 세 가지 조건(충돌 저항, 역상 저항, 제2역상 저항성) 중 하나라도 만족하지 않는 값을 찾는 것

2.3 알고리즘의 안전성 유지 기간

알고리즘에 대한 적절한 키 크기를 선택할 때에는 알고리즘의 보안 강도와 해당 알고리즘의 안전성 유지 기간 및 보호하고자 하는 데이터의 예상 보안 수명 기간을 고려하는 것이 중요하다.

○ 키 길이에 따른 안전성 유지 기간

NIST기준[13]에 의하면 2030년 까지 보안 강도가 최소 112비트 이상으로 제공되어야 하고 그 이후에는 최소 128비트의 보안 강도가 제공되어야 한다. 표 16은 알고리즘과 키 선택에 따라 안전성을 보증할 수 있는 기간(안전성 유지 기간)을 보여주는 NIST의 권고안이다.

[표 16] 키 길이에 따른 안전성 유지 기간에 대한 NIST 권고 (단위: 비트)

암호 기술 / 기간	대칭키 암호 알고리즘 (키 길이)	공개키 암호 알고리즘(키 길이)			해시함수[14] (보안 강도)
		인수분해 기반	이산대수 기반	타원곡선 암호기반	
2030년 까지	112 이상	2048 이상	2048(공개키), 224(개인키) 이상	224 이상	112 이상
2031년 이후	128 이상	3072 이상	3072(공개키), 256(개인키) 이상	256 이상	128 이상

첫 번째 열은 특정 알고리즘으로 보호되는 데이터가 안전할 것으로 예상되는 기간을 나타내고 있고 두 번째와 세 번째 열은 각각 첫 번째

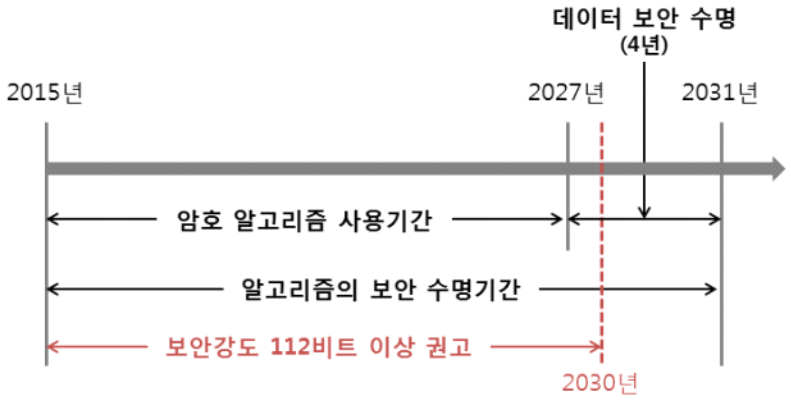
13) NIST SP 800-57 (Recommendation for Key Management-Part1: General), Rev.4, 2016.1
14) 해시함수의 보안 강도는 출력비트 크기가 아니며 해시함수의 보안 강도는 부록1 참고

열의 안전성 유지 기간을 만족할 수 있는 대칭키 암호, 공개키 암호 알고리즘의 키 길이를 나타낸다. 마지막 열은 첫 번째 열의 안전성 유지 기간을 만족할 수 있는 해시함수의 보안 강도를 의미한다.

표 16에 표시된 알고리즘과 키 길이는 표에 명시된 기간 동안 데이터를 보호하는 데 적합한 보안강도이므로 이보다 낮은 보안 강도를 가지는 알고리즘과 키 길이는 해당 기간 동안 데이터 보호에 사용되지 않도록 해야 한다.

❍ 알고리즘의 보안 수명 기간

알고리즘의 보안 수명 기간[15]이란 암호 알고리즘 사용 기간과 알고리즘을 적용할 데이터의 보안 수명 기간을 합친 것이다. 암호 알고리즘의 안전성 유지 기간 이후에도 데이터의 안전성을 유지해야 한다면 데이터 암호화에 해당 알고리즘을 적용하지 말아야 한다.



<그림 14> 알고리즘의 보안 수명기간 개념 및 예시

15) NIST SP 800-57 (Recommendation for Key Management-Part1: General), Rev.4, 2016.1

예를 들어, 정보가 2015년에 암호화 되었고 해당 데이터를 보호해야할 최대 기간이 15년이라면 표 16에 나타난 보안 강도 112 이상을 만족하는 모든 알고리즘과 키 길이를 사용할 수 있다. 그러나 최대 보호 기간이 16년이라면 128 이상의 보안강도를 제공하는 알고리즘과 키 길이를 사용해야 한다.

❍ 주기적인 안전성 검토

NIST에서 권고하는 안전성 유지 기간은 해당 문서 배포 시점까지 알려진 공격 방법을 기준으로 평가한 것으로 인수분해 알고리즘의 발전, 일반 이산대수 공격의 발전, 타원 곡선 이산대수 공격과 양자 컴퓨팅 등의 영향으로 변경될 수 있다. 양자 컴퓨팅이 상용화 된다면 대칭키 암호 및 해시 함수는 안전성을 위해 키와 출력 길이를 증가시켜야 하며, 공개키 암호는 기존의 구조 대신 새로운 구조에 기반한 암호 시스템을 개발해야 한다.

이와 같이 암호 알고리즘의 안전성이 계속해서 바뀔 수 있기 때문에 암호기술 환경의 변화에 따라 암호 알고리즘의 안전성을 주기적으로 검토해야 한다.

제3절 암호 운영모드 선택 시 고려사항

안전한 암호 알고리즘을 사용하더라도 암호 운영모드를 적절히 사용하지 않을 경우 기존의 보안 강도를 유지하지 못 할 수 있다. 따라서 알고리즘 운영 시 적합한 패딩기법과 운영모드를 선택해야 한다.

○ 패딩기법

운영모드를 적용할 때에는 평문과 암호문의 관계성을 낮추거나 블록 암호에서 블록 크기를 맞추기 위해 암호화 전에 특정 데이터를 사전 공유된 규칙으로 채우는 패딩을 적용하게 된다. 블록암호의 경우는 빈자리만큼 '0' 비트를 붙이거나 '1'을 추가한 후 '0' 비트를 덧붙이는 등의 방법[16]이 있고 공개키 암호에서는 PKCS#1에 나타난 내용으로 패딩처리를 하는 것이 일반적이다.

○ 대칭키 블록 암호 알고리즘 운영모드

블록 암호 알고리즘은 일정한 크기의 평문을 암호화하므로 실제 단일 8바이트나 16바이트 보다 더 긴 평문(이메일 또는 파일)을 암호화하기 위한 운영모드가 필요하다. 블록암호 운영모드의 목적은 암호 알고리즘에 의해 생성된 블록 간의 연관성을 주는 것이며, 초기벡터를 추가하여 출력값으로부터 입력값을 유추하기 어렵게 만들 수 있다. NIST에서는 대칭키 블록 운영모드를 평가하여 표 17과 같이 11개의 승인된 블록암호 운영모드를 권고[17]한다.

16) TTAS_KO-12_0025 부록의 패딩기법(덧붙이기 방법) 참고
17) NIST, BLOCK CIPHER MODES, 2014.3., http://csrc.nist.gov/groups/ST/toolkit/BCM/index.html

[표 17] NIST에서 승인된 대칭키 블록 운영모드

용도	대칭키 블록 운영모드	금융권 이용분야	구체적 예시
기밀성	ECB, CBC, CFB, OFB, CTR, XTS-AES	거래내용 암호화 등	SEED128-CBC
인증	CMAC	신용카드 결제 등	AES, SEED128-CMAC
기밀성/인증	CCM, GCM, KW, KWP	교통카드 지불 등	AES, SEED128-CCM

표 17의 첫 줄에 있는 6개 운영모드(ECB[18], CBC[19], CFB[20], OFB[21], CTR[22], XTS-AES[23])는 기밀성을 보장하기 위한 용도로 이용되고, CMAC[24]은 인증용도로 이용되며, 기밀성과 인증을 위한 4개의 운영모드(CCM[25], GCM, KW, KWP)가 있다. 여러 종류의 운영모드가 구조적 특징으로 인한 장단점이 있기 때문에 국가별로 권고하는 운영모드가 상이한 경우가 있다. 가장 많은 국가에서 권고하고 다양한 분야에 사용되는 운영모드는 CBC와 CTR방식이다.

18) ECB(Electronic Codebook): 가장 단순한 구조의 운영모드로 평문 메시지를 블록 크기 단위로 나누어 각 블록을 독립적으로 암호화함
19) CBC(Cipher Block Chaining): 초기벡터 이용, 각 블록은 암호화되기 전에 이전 블록의 암호화 결과와 XOR됨, 암호화 시 병렬 처리 불가능, 복호화 시 병렬 처리 가능
20) CFB(Cipher Feedback): 초기벡터 이용, CBC모드의 변형으로 블록암호를 자기 동기 스트림 암호로 변환할 수 있음, 암호화 시 병렬 처리 불가능, 복호화 시 병렬 처리 가능
21) OFB(Output Feedback): 초기벡터 이용, 블록 암호를 동기식 스트림 암호로 변환할 수 있고, 영상이나 음성 신호에 많이 이용, 병렬처리 불가능
22) CTR(Counter): 블록 암호를 스트림 암호로 변환하는 구조, 블록마다 1씩 증가하는 카운터를 반영하여 암호문을 생성, 병렬처리 가능
23) XTS-AES: NIST SP800-38E문서 참고, 저장기기의 기밀성 보호용도
24) CMAC(Cipher-based MAC): NIST SP800-38B문서 참고, 암호 기반 메시지 인증
25) CCM(Counter with CBC-MAC): NIST SP800-38C문서 참고, 기밀성 목적의 CTR과 인증용도의 CBC 기술을 이용

금융권에서 주로 이용되는 운영모드 방식은 CBC방식이다. CBC 운영모드는 초기벡터를 제2의 키로 사용할 수 있다는 점에서 보안성이 높고 다른 운영모드에 비해 안전한 운영모드로 알려져 있다. 하지만 암호화 시 블록을 병렬로 처리하는 것이 불가능하고 암호문이 블록의 배수가 되기 때문에 복호화 후 평문을 얻기 위해 패딩을 해야 한다는 단점이 있다[26].

CBC외에 ECB와 OFB는 운영모드가 갖는 구조적 특징이 보안성에 영향을 주기 때문에 선택 시 유의해야 한다. ECB의 고려사항은 빠른 연산처리가 가능한 반면 블록마다 연관성이 부족하여 보안 관점에서 다른 모드에 비해 취약하다는 점이고 OFB는 암·복호화 방법이 동일하여 두 번 암호화를 수행하는 경우 평문이 유출된다는 점이다. 또한, OFB, CTR, CFB는 스트림 암호와 유사한 구조를 가지기 때문에 암호키를 재사용할 경우 공격자에게 평문이 쉽게 노출 될 수 있다. 따라서 해당 운영 모드 사용 시에는 매번 다른 키로 암호화해야 한다.

한편, 암호포럼과 학계에서는 CBC모드를 사용할 경우 평문이 일부 해독될 수 있는 취약점이 발견되어 CTR모드의 사용을 권하는 움직임이 있고[27], 통신 프로토콜에서는 CBC모드가 암호블록의 무결성을 보장하지 않는다는 점이 알려지면서 CBC 대신 GCM운영모드의 사용이 권장[28] 되고 있다.

❍ 공개키 암호 알고리즘 용도별 운영모드

현 금융권에서는 공개키 암호 알고리즘으로 주로 RSA를 사용한다.

---

26) 금융보안연구원, 금융부문 암호기술 관리 가이드, 2010.1
27) The Attacks are Efficient enough to be practical, Padding Oracle Attack, CRTYPTO 2012.8
28) Guidelines for the Selection, Configuration, and Use of Transport Layer Security(TLS) Implementations, NIST SP 800-52 Rev. 2(DRAFT)

[표 18] 공개키 암호 운영모드

용도	공개키 암호 운영모드	금융권 이용분야	예시
전자서명	RSASSA-PSS	키 전송 시 서명	계좌이체 부인방지
	RSASSA-PKCS#1 (v1.5)	거래내용 전자서명	거래내용에 대한 PKCS#7 SignedData 전자생성문 생성
	KCDSA	-	행정전자서명인증서에 이용
	ECDSA, EC-KCDSA	키패드보안	상호인증 ECDSA
키 교환 및 암호화용	DH, ECDH	스마트 뱅킹	가상키패드에서 ECDH 사용
			VPN 키 교환
	RSAES-OAEP	세션키 교환	증권 시스템 등

하지만 RSA는 같은 평문을 암호화할 경우 항상 같은 암호문을 출력하기 때문에 암호문으로 쉽게 평문을 추측할 수 있다. 따라서 메시지에 난수를 넣어 같은 평문을 암호화해도 매번 다른 암호문을 출력하도록하는 OAEP 패딩기법을 적용해야 한다. OAEP기술은 규격화되어 PKCS#1와 RFC 2437에 표준화 되어 있다.[29)]

EMC가 제정한 RSA 암호 표준에 의하면 데이터 암·복호화에 사용되는 운영모드로는 RSAES-PKCS#1(v1.5)와 RSAES-OAEP를 이용할 수 있는데, 국내 환경의 경우 RSAES-OAEP를 권장한다. 또한, 전자서명용으로는 RSASSA[30)]-PKCS#1(v1.5), RSASSA-PSS[31)], KCDSA, ECDSA, EC-KCDSA를 권장한다.

29) 암호기술의 이해, 암호분석, 2013.3, 도서출판 그린
30) RSASSA=RSA Signature Scheme with Appendix
31) RSASSA-PSS(Probabilistic Signature Scheme) 전자서명 알고리즘, 대부분의 전자서명 기법과 비슷하게 메시지를 hash한 값을 전자 서명하는 방식을 따름

금융권에서 이용되는 대표적인 공개키 운영모드는 키 전송 시 사용되는 RSA-OAEP와 키 전송 과정의 전자서명 시 사용되는 RSASSA-PSS, 보안 관리자를 인증하거나 거래 내용에 대한 전자서명 시 사용하는 RSASSA-PKCS#1이 있다.

○ 미래 운영모드(형태보존암호)

형태보존암호(FPE, Format-Preserving Encryption)란 기존의 운영모드와 달리 암호화를 거치더라도 메시지의 형태와 길이가 동일하도록 생성하는 암호기술이다. 대표적인 형태보존암호 방식으로는 '블록암호 운영모드를 이용한 방식'이 있다.

<그림 15> 카드번호를 암호화할 경우 암호문의 형태(예시)

그림 15는 카드번호를 형태보존 암호로 암호화할 경우 암호문의 형태를 표현[32]한 것이다. 기존 블록암호로 카드번호를 암호화할 경우 메시지 길이 및 자료형 변화가 불가피하지만 형태보존암호로 암호화할 경우에는 평문과 암호문의 길이가 변하지 않는다. 형태보존암호의 이러한 장점은 암호화 메시지 저장을 위한 추가 저장 공간 확보가 불필요하고 시스템 변경 최소화로 인한 암호화 시스템 구축비용 절감 효과를 기대할 수 있다.

32) 금융보안연구원, 2014년 6월 전자금융 보안 동향 & 연구, 용어정의

이 운영모드는 NIST에서 2016년에 관련 표준[33]을 제정하였고, 국내의 대표적인 형태보존암호로는 FEA(Format-Preserving Encryption)[34]가 있다.

33) NIST SP 800-38G (Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption), 2016.3.
34) 형태 보존 암호 FEA, TTAK.KO-12.0275, 2015.12. TTA.

제4절 의사난수 생성기 선택 시 고려사항

난수는 대칭키 암호의 키 생성, 공개키 암호 또는 전자서명 시스템의 매개변수 생성, 보안 통신 프로토콜 등 많은 분야에서 사용된다. 또한 암호시스템에서 사용되는 암호키는 난수의 예측하기 어려운 성질에 안전성을 의존하고 있다. 따라서 난수를 생성하고 관리할 때 난수 발생기가 기본 요건이 만족되지 않는다면 암호 시스템 전체 안전성에 영향을 미치게 된다.

본 절에서는 난수가 만족해야하는 기본 성질과 이를 생산하기 위한 의사난수 생성기, 순수난수 생성기에 대해 살펴본다.

○ 기본 성질

이론적 의미의 난수는 기본적으로 표 19의 성질을 만족해야 한다.

[표 19] 난수의 기본성질

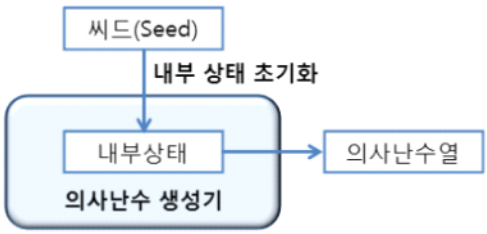
구분	설명	비고
무작위성	통계적인 편중 없이 수열이 무작위로 되어있는 성질	무작위성만을 가진다면, 암호기술에 사용할 수 없음
예측 불가능성	과거의 수열로부터 다음 수를 예측할 수 없다는 성질	알고리즘은 공격자에게 알려져 있다고 가정하기 때문에 의사난수 생성기에서는 씨드(Seed)를 사용
재현 불가능성	같은 수열을 재현할 수 없다는 성질, 재현하기 위해서는 수열 그 자체를 보존해야만 함	소프트웨어만으로는 만족 불가

난수를 생성하는 이상적인 방법은 물리적인 방법(방사능 물질의 붕괴, 대기소음, 하드웨어장비)을 이용하여 이론적 의미의 난수를

생성하는 순수난수 생성기를 이용하는 것이다. 하지만 효율과 비용의 문제로 인해 초기에 주어진 씨드(Seed)를 이용해서 난수처럼 보이는 의사난수 값을 생성하는 의사난수 생성기를 만들게 되었다.

○ 의사난수 생성기

의사난수 생성기는 '내부 상태'와 외부에서 주어진 '씨드(Seed)'를 기초로 의사난수를 생성하는 알고리즘이다. 이 알고리즘은 의사난수를 계산하는 기능과 내부 상태를 변화시키는 기능을 가진다.



<그림 16> 의사난수 생성기의 구조

초기 씨드값은 의사난수 생성기의 내부 상태 초기화에 필요한 값으로 무작위의 비트열이어야 하며, 비밀 값이어야 한다. 또한 씨드값을 통해 의사난수열을 생성하기 때문에 무엇으로 씨드값을 사용할지가 중요하다.

씨드값은 비용과 효율성 문제로 소프트웨어에서 가져오거나 외부 주변 장치로부터 입력 받는 경우가 대부분이다. 예를 들어 화면 정보, 시스템 클럭, 프로그램 카운터 등의 시스템 정보와 키보드 입력값, 마우스 좌표 값 정보 등을 이용하여 생성한다.

운영체제별로 이용 가능한 씨드로 이용될 수 있는 잡음원[35]을 표 19에 나타내었다.

35) 운영체제별 잡음원 수집 및 응용 지침, TTAK.KO-12.0235, 2013.12. TTA.

[표 20] 시스템별 사용 가능한 잡음원

시스템	잡음원	
Linux*	운영체제	키보드, 마우스, 디스크 이벤트, 인터럽트, 현재 프로세스 ID, 현재 프로세스의 부모의 ID, 그룹의 ID, 시스템의 현재 시간, 프로세스의 자원 사용량, 사용된 디스크의 크기 정보, 메모리 사용량 정보, 시스템 자원의 평균 부하량, 시스템 사용 통계, 저장 장치 IO 통계 현황, 가상 메모리 통계, 시스템 프로세스 전력 정보, 시스템 프로세스 휴면 정보 등
	하드웨어	GPU온도, 공유 메모리의 경쟁상태, 잡음원 생성기가 구현한 하드웨어 모듈
Windows	씨드	키보드, 마우스, 디스크이벤트, 인터럽트 등
	운영체제	현재 프로세스 ID, 현재 스레드 ID, 메모리의 상태 정보, 부팅한 이후 누적 된 CPU의 클럭, 1초 동안 발생한 CPU의 클럭, 프로세스 힙의 핸들, 부팅한 이후 경과된 시간, 남은 디스크 공간정보, 현재 프로세스 환경 블록의 해시값, 시스템 파일 캐시 정보, 시스템 프로세스 전력정보, 시스템 프로세스 휴면 정보, 시스템 인터럽트 정보 등
	하드웨어	GPU온도, 공유 메모리의 경쟁상태, 잡음원 생성기가 구현한 하드웨어 모듈
Android	씨드	키보드, 디스크 이벤트, 인터럽트 등
	운영체제	현재 프로세스의 ID, 현재 프로세스의 자원 사용량, 시스템 사용 통계, 저장장치 IO 통계 현황, 사용된 프로세스 시간, 수행중인 모든 프로세스의 정보 등
	하드웨어	잡음원 생성기가 구현된 하드웨어 모듈
iOS	씨드	키보드, 디스크, 인터럽트
	운영체제	현재 프로세스 ID, 그룹 ID, 네트워크 인터페이스와 IP정보, 메모리 사용량 정보, 수행 중인 모든 프로세스의 정보, 사용된 프로세스 시간 등
	하드웨어	잡음원 생성기가 구현된 하드웨어 모듈

* 리눅스 커널 환경에 따라 Java 기반의 SecureRandom 함수를 사용하여 잡음원 생성 시 프리징(Freezing) 등의 오류 발생에 유의

의사난수 생성기에서 초기 씨드 값과 출력값의 엔트로피는 예측 불가능성과 사용되는 데이터의 양에 따라 결정되는 성질로서 예측이 쉬운 정보라 할지라도 많은 양의 데이터를 사용하여 엔트로피를 향상시킬 수 있다.

○ 의사난수 생성기의 난수 생성 방법

의사난수 생성기의 난수 생성 방법으로 무작위 방법, 선형합동법, 일방향 해시함수를 사용하는 방법, 암호를 사용하는 방법, ANSI X9.17 등이 있다. 금융시스템에서는 ANSI X9.17과 FIPS186-2를 키 생성용 난수 발생기로 이용하며 160비트 이상의 보안 강도를 가진다.

암호를 사용하는 방법 중 한국에서 시행되는 암호모듈 검증제도(KCMVP[36])에서 그 안전성이 승인된 표준 의사 난수 생성기의 현황은 다음과 같다.

[표 21] 표준 의사 난수 생성기 현황

의사 난수 생성기	비 고
(해시함수 기반) HASH_DRBG	ISO/IEC 18031 Information technology – Security techniques – Random bit generation(2011), NIST SP 800-90 Recommendation for Random Number Generation using Deterministic Random Bit Generators
(HMAC 기반) HMAC_DRBG	
(블록암호 기반) CTR_DRBG	

○ 순수 난수 생성기

순수 난수를 생성하기 위해서는 TRNG(True Random Number Generator)과 같은 추가 장비가 필요하기 때문에 많은 분야에서 의

36) 암호모듈검증, 난수발생기 KCMVP: Korea Cryptographic Module Validation Program, http://service1.nis.go.kr/

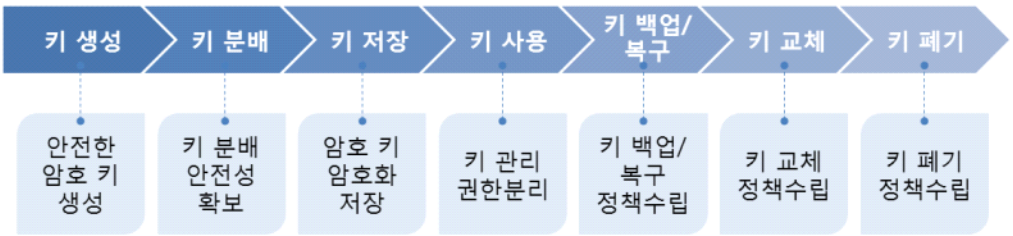
사난수 생성기를 사용해왔다. 하지만, 소프트웨어로 생성된 의사난수 생성기는 컴퓨터의 유한한 내부 환경에서 언젠가는 수열이 반복되고 주기를 가지게 되는 한계가 있다.

순수 난수 생성 방법 중 가장 주목 받는 것은 물리적 현상을 관측하여 이 값을 비트로 변환하는 것으로 최근에는 양자역학적 현상을 이용해 순수 난수를 만드는 방법들이 현실화되고 있다. 양자역학을 이용한 난수 생성은 기존보다 저비용으로 동작 속도까지 높일 수 있다는 장점이 있다.

제5절 암호키 관리 고려사항

암호에 사용된 키가 유출될 경우 누구든 유출된 키를 통해 암호문을 복호화할 수 있다. 따라서 키를 안전하게 관리하는 것이 매우 중요하다. 암호키의 종류는 특성과 사용처, 속성 등에 따라 다양하게 구분할 수 있지만, 모든 키가 생성부터 폐기까지 안전하게 관리되어야 하는 것은 동일하다.

본 절에서는 금융회사 서버에서의 키 관리 절차를 위주로 설명한다. 단계별 키 관리 절차는 그림 17과 같다.



<그림 17> 키 관리 절차

○ 키 생성

키를 생성하기 이전에 보안을 위해 유효기간을 설정할 필요가 있다. 유효기간은 사용자 또는 관리자가 암호키를 사용할 수 있도록 허용된 기간과 사용기간이 완료된 이후라도 추후 복호화 과정에서 해당 암호키를 사용하도록 허용된 기간으로 구분할 수 있다. NIST의 키 관리 권고안[37]에서는 유효기간을 송신자는 최대 2년, 수신자는 최대 5년으로 설정하도록 제시하였다.

37) NIST SP 800-57 (Recommendation for Key Management-Part1 : General), Rev.3, 2012.7, http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf

[표 22] 암호키 사용 유효기간(NIST권고안)

키 종류		사용 유효기간	
		송신자 사용기간	수신자 사용기간
대칭키 암호 알고리즘	비밀키	최대 2년	최대 5년[38]
공개키 암호 알고리즘	암호화 공개키	최대 2년	
	복호화 개인키	최대 2년	
	검증용 공개키	최소 3년	
	서명용 개인키	최대 3년	

키를 생성하기 위해서는 암호학적으로 안전한 키를 생성해야하는데, 이는 예측이 불가능하고 위조할 수 없는 난수를 사용해야 되고 난수라도 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.

키를 예측 불가하게 만들기 위해서는 난수와 비밀번호를 이용하는 방법이 있다. 난수를 이용한 키 생성은 하드웨어적 난수 생성기나 암호용 의사난수 생성기를 활용하고 비밀번호를 이용한 키 생성은 비밀번호를 해시함수에 입력해서 얻어진 해시 값을 키로 이용하거나 비밀번호에 '솔트(Salt)'라 불리는 난수를 부가해서 해시함수에 입력 하는 방법 등을 쓸 수 있다. 한편, 암호용으로 설계되지 않은 의사난수 생성기의 경우는 '예측 불가능'이라는 성질을 갖지 않기 때문에 이용하는 것을 피해야 한다.

38) 수신자의 경우 송신자가 전송한 암호화된 데이터를 수신 한 후 필요시에 복호화 할 수 있으므로 송신자보다 오랜 기간 동안 비밀키를 사용할 수 있음

키 생성 후에는 키를 등록하고 인증서를 생성하는 절차가 추가적으로 있을 수 있다. 등록 시 생성된 키를 정당한 사용자와 관련시키는 것은 등록기관(RA)에 의해 이루어지고, 등록기관은 키와 관련된 정보를 안전하게 유지하는 역할을 해야 한다.

또한 키 등록 기관은 키 등록뿐만 아니라 키를 말소시키는 역할도 수행한다. 키 인증서는 보통의 경우 키 생성 후, 공개키와 객체의 연관성을 보장하는 확인서를 뜻하며, 인증기관(CA)은 사용자로부터 키 인증에 대한 요구를 받은 경우 키 인증서를 생성한다.

❍ 키 분배

키 분배는 인가된 객체 사이에 키 또는 키 생성 데이터가 안전하게 공유되는 것을 의미한다. 키 분배 방법으로는 키를 사전에 공유하는 방법과 공개키 암호로 키를 분배하는 특정 메커니즘을 사용하는 방법, 대칭키 암호로 키 분배센터(KDC : Key distribution center)를 이용하는 방법이 있다.

키 분배센터란 모든 사용자가 완전히 신뢰하는 서버로서 각 사용자와 비밀키를 공유하고 있는 제3자의 입장에서 분배하고자 하는 키를 만들어 각 사용자의 비밀키로 암호화하여 분배해주는 역할을 한다.

분배 후에는 저장 등에 대한 단계를 진행하게 되며, 저장 시에는 키를 사용하거나 복구를 위한 백업을 위해 물리적으로 안전한 장치에 저장되는 것이 바람직하다. 키 재료에 대해 기밀성, 무결성을 제공해야 한다.

❍ 키 저장

암호키를 저장하는 방법 중 쉬운 방법은 파일에 암호키를 저장하는 것으로 이러한 방법은 암호문에 대한 접근권한이 있다면, 임의로 데이터를 복호화 할 수 있다. 또한 해당 파일과 함께 유출된 경우, 데이터를 해독할 수 있기 때문에 내·외부 위협에 노출될 가능성이 있다.

따라서 키를 반복해서 사용하는 경우에 반드시 키를 암호문과 동일한 장소에 보관하지 않고 분리된 공간이나 금융회사 HSM 등의 안전한 장소에 보관하거나 암호화하여 보관하여야 한다.

한편 하나의 키로 여러 개의 키를 암호화하여 관리, 저장하는 경우는 안전한 장소에서 별도로 관리해야할 키의 개수를 줄일 수 있어서 집중도 높은 관리를 할 수 있는 장점이 있으나 암호화에 사용한 키의 유출로 인해 피해가 발생할 수 있음을 감안해야 한다.

❍ 키 사용

키 사용에서는 암호키에 대해 비인가접근을 방지하기위해 암호키 분리정책과 암호키에 대한 접근제어 정책을 수립하여야 한다.

키는 통신에 사용되는 키와 저장에 사용되는 키로 구분할 수 있으며 통신용 암호키 중 세션키는 원칙적으로 통신 1회에 한해 사용해야 한다. 저장용 암호키는 정보를 처리하는 담당자와 보안 관리자의 권한을 분리해서 관리해야 한다.

특히 정보 처리 담당자가 암호화된 데이터를 직접 조회하거나 이 데이터에 접근통제 권한을 가질 수 없도록 운영되어야 한다.

공유 메모리에 호출된 암호키를 평문으로 저장해서는 안 되며, 키를 암호화하는 키 같은 핵심보안 매개변수를 안전하게 관리하고 키 암호화에 사용한 키는 공개키를 이용하여 암호화 하는 등 안전하게 사용할 수 있는 방안을 마련하여야 한다.

❍ 키 백업, 복구

암호키 및 비밀번호의 분실 또는 훼손과 키 관리자의 퇴사 등으로 암호키와 비밀번호를 알 수 없는 경우에 대비하여 암호키를 복구하기 위한 백업 및 복구 절차를 수립하여야 한다.

암호키는 백업 정책에 따라 주기적으로 백업되어야 하며, 백업된 키 정보는 암호화하여 저장해야 한다. 또한 키 백업에 사용한 비밀번호는 별도로 관리되어야 한다. 암호화된 데이터를 백업하는 경우 해당 데이터를 암호화한 암·복호화 키도 백업해야 향후 백업된 암호화 데이터를 복호화하여 이용할 수 있다.

또한 암호키 분실 시 데이터 복호화가 불가능하므로 분실, 훼손, 파괴 등의 경우에 대비하여 키를 복구하기 위한 방안도 마련되어야 한다. 키 복구는 합법적 상황에서 암호문을 복호화 하거나, 사용자가 자신의 비밀키를 분실했을 경우 유사시에 허가된 사용자만이 복호화를 할 수 있는 기능을 제공하기 위해 수행된다.

❍ 키 교체(갱신)

암·복호화 키는 키의 용도에 따라 법령 또는 내부 정책에 부합하는 기간마다 교체, 갱신하여 보안성을 유지하도록 해야 한다. 특히 통신의 경우 기밀성을 높이기 위해 정기적인 키 교환이 필요하다.

갱신 시에는 원본 키로부터 파생키를 유도하는 방법이 있다. 이 방법을 쓸 경우, 유도된 키가 원본 키를 노출시키지 않도록 현재 키의 해시 값을 다음 키로 갱신하는 것이 일반적이다.

○ 키 폐기

키 폐기는 더 이상 사용될 필요가 없는 키에 대한 안전한 폐기를 의미한다. 키를 폐기한다는 것은 키와 관련된 모든 기록을 제거하므로 폐기된 키를 다시 복구 시킬 수 없도록 하는 것을 의미한다.

또한 사용되는 키뿐만 아니라 보관된 모든 복사본 키에 대한 폐기도 포함한다. 보관된 키를 폐기할 때는 보관된 키에 의해 보호된 자료가 더 이상 필요 없는지의 여부를 사전에 판단하여야 한다.

암호키는 여러 가지 이유로 폐기 될 수 있다. 예를 들어, 암·복호화 키 또는 마스터키를 분실하거나 키의 비밀성이 깨진 경우 등에 대비하여 키를 폐기하기 위한 절차 및 방법을 수립함으로써 허가된 관리자가 절차에 따라 폐기할 수 있도록 하여야 한다.

암호키를 폐기하게 되면 해당 키로 암호화 된 데이터는 더 이상 복호화 할 수 없으므로, 암호키 폐기 전 해당 키로 암호화된 데이터는 복호화 후 암호키를 폐기하여야 한다.

또한 제품 종료 후 메모리에 로드된 암호키와 핵심보안 매개변수는 모두 제거하여 유추할 수 없도록 해야 한다.

제6절 암호통신 프로토콜 설계 시 고려사항

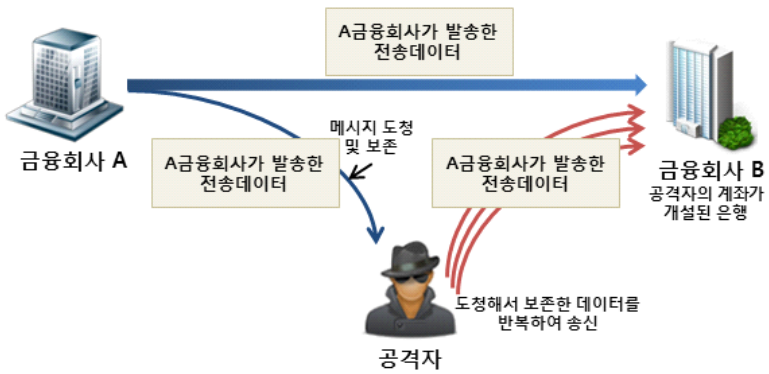
6.1 암호통신 공격 및 대응 방안

금융권 암호통신 프로토콜에 대해서는 대표적인 공격으로 재전송 공격이나 반사 공격 등이 있다. 대표적인 공격에 대해 시나리오를 살펴보고, 이에 대한 대응 방안과 프로토콜 설계 시 인지해야할 고려사항을 알아보도록 한다.

❍ 재전송 공격(Replay Attack)

- 공격 시나리오 1: 매 거래 시 동일한 암호문이 사용되면 공격자는 암호문에 대응되는 평문을 알지 못해도 암호문을 재전송하여 거래를 성공시킬 수 있다.

- 공격 시나리오 2: 공격자가 사용자의 아이디, 비밀번호를 알지 못하더라도 전송되는 아이디, 비밀번호의 암호문을 수집한 후 수집된 암호문을 재전송하여 사이트에 로그인할 수 있다.

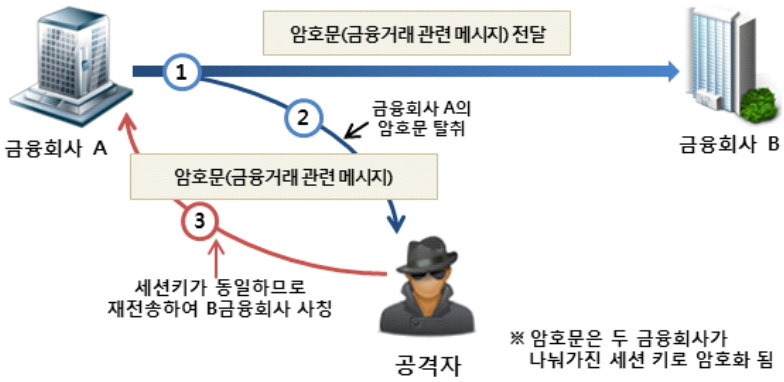


<그림 18> 재전송 공격 시나리오

재전송 공격에 대응하기 위해서는 같은 암호문이 생성되지 않도록 시간(Time-Stamp)정보, 임의의 값(Nonce), 상대방과 사전에 공유한 값(Sequence)을 이용하여 평문과 함께 암호화하여 전송하는 방법을 이용하거나, OTP를 이용하여 매 전송 데이터가 유일성을 가진 데이터로 생성되고 수신자 측에서는 정당한 값인지 검증하는 방법이 있다.

❍ 반사 공격(Reflection Attack)

반사 공격은 동일한 세션키를 사용하는 대상이 통신할 때, 상대방을 간단히 인증하기 위해 전송되는 내용에 구체적인 정보 및 전송자 유일 정보가 명시되어 있지 않는 상황에서 일어날 수 있다.



<그림 19> 반사공격 시나리오

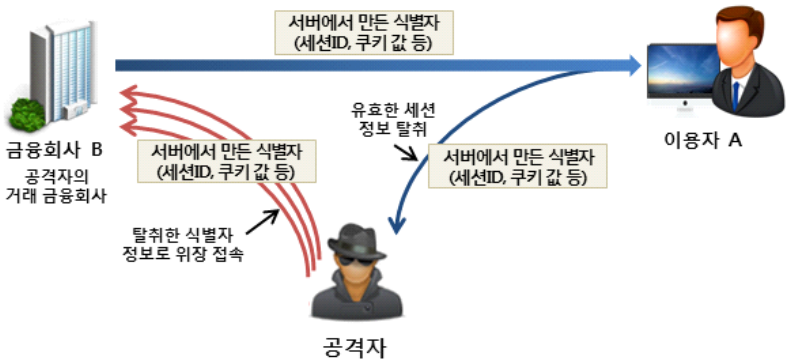
- 공격 시나리오: 고정키 암호 시스템의 경우 특정 메시지를 암·복호화 하는 방법으로 상대방을 인증하기 때문에 공격자는 중간에서 세션키를 갖고 있지 않지만, 전송된 값을 탈취하여 동일키를 가진 상대인 것처럼 속일 수 있다.

반사 공격은 동일한 키를 사용하는 환경의 경우 평문, 송수신자의 아이디, 전송 방향(예: A→B, B→A)을 함께 명시하여 암호화하고, 복호화 후 송수신자의 아이디와 전송 방향이 올바른지 체크하는 과정을 삽입하고 각 방향에 따라 키를 다르게 설정하여 대응할 수 있다.

❍ 세션 훔치기 공격(Session Hijacking)

이용자와 서버 간에 연결된 세션에서 서버는 자신에게 접속되어 있는 이용자를 구분하기 위해 일반적으로 세션 아이디를 발급한다.

- 공격 시나리오: 공격자는 세션 아이디[39]만 알면 해당 이용자의 로그인 세션을 사용할 수 있어 이용자 인증 정보(예: 아이디, 비밀번호)를 모르더라도 서버에 접속해 여러 정보를 조회하거나 거래를 수행할 수 있다.



<그림 20> 세션 훔치기 공격 시나리오

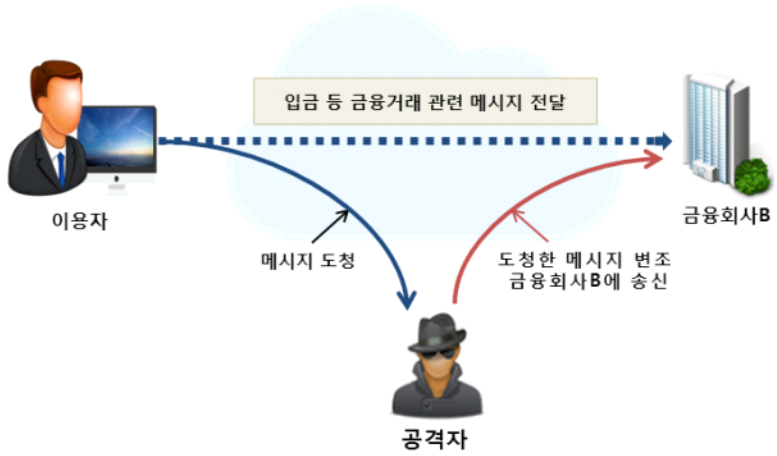
현재 알려진 세션 훔치기 공격을 대응하기 위해서는 세션 이용 거래 시 반드시 암호화된 요청 및 응답을 하고 최초 키 교환(Full Handshake)이 일어날 때 세션을 초기화하는 방법을 이용 할 수 있다.

39) 웹 환경에서는 서버가 발급하는 쿠키에 담김

❍ 중간자 공격(Man in the Middle Attack)

중간자 공격이란 통신 대상 사이에서 내용을 도청 및 변조하는 공격을 의미하며 이러한 공격을 수행하는 공격자를 능동형 공격자(Active Attacker)라 정의한다.

- 공격 시나리오: 중간자 공격자은 통신 대상이 정상적으로 암호화를 통해 정당한 이용자와 통신하고 있다고 생각하지만 실제로는 공격자와 통신하거나, 변조된 데이터를 정당한 이용자가 보낸 것으로 위조하는 것이다.

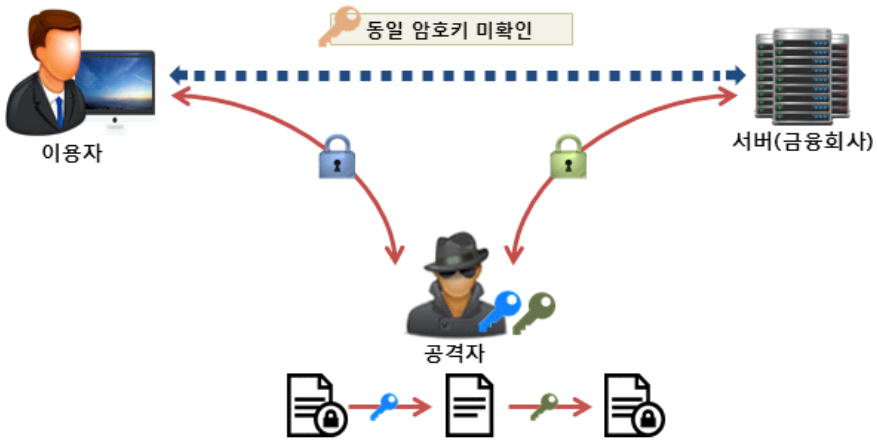


<그림 21> 중간자 공격 시나리오

금융권에서는 중간자 공격에 대한 대응을 위해 암호기술 구현방법을 조합하여 대응하고 있다. 예를 들어 거래를 진행하는 기기 외에 별도의 서명 기기를 통한 본인확인으로 중간자의 개입이 더 어렵게 구성하는 방식이 있다.

❍ 동일 암호키 확인(Key Confirmation)

동일 암호키를 확인하는 것은, 암호키 교환 과정에서 양 상대방이 공격자의 방해 없이 동일한 키를 가졌는지 여부를 확인하는 것으로 암호 통신 시에는 서로 동일한 암호키를 공유했는지 확인이 이루어진 후에야 통신을 수행해야 한다.



<그림 22> 동일 암호키 확인

- 공격 시나리오: 동일 키 공유 여부를 확인하지 않을 경우 공격자가 의도한 키로 공유될 가능성이 존재하고 이 경우 공격자가 암호화 된 메시지를 복호화해 볼 수 있게 된다.

암호 프로토콜에서는 상대의 신원검증을 위해 인증서 등의 수단을 이용하여 이용자 확인과 동시에 동일 암호키 여부를 반드시 확인해야 한다.

❍ 전방향 안전성(Forward Secrecy)

전방향 안전성은 세션키가 유출되더라도 이전 세션에서 전송된 암호문으로부터 평문에 대한 정보를 알 수 없어야 함을 의미한다.

- 공격 시나리오: 내부자에 의해 개인키가 유출되거나 관리상의 문제로 개인키를 도난당한다면 도청만으로 세션키가 유출될 수 있다.

<그림 23> 전방향 안전성

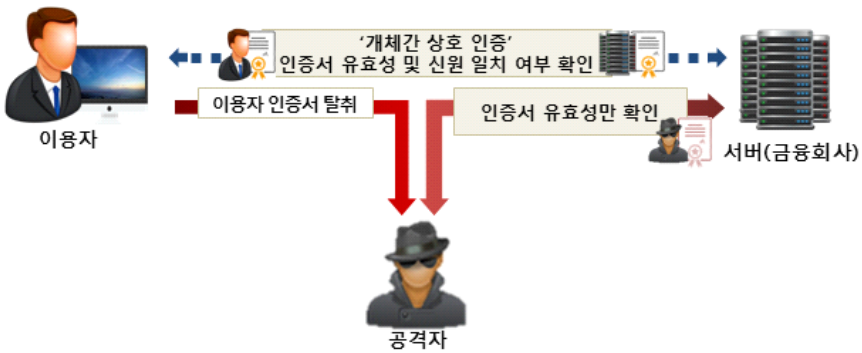
일부 금융회사에서는 개인키 유출 방지를 위해 HSM(Hardware Security Module) 장비를 도입하는 등 위협에 대비하고 있지만 백업을 위해 개인키를 파일로 보관하거나 개인키에 쉽게 접근할 수 있도록 하는 등 관리상 문제가 발생하고 있다.

따라서 데이터 암호를 위해서 이용되는 키는 세션별로 상이해야하며, 현재의 세션키를 통해 이전의 세션키가 유추되지 않도록 해야 한다.

❍ 개체 간 상호 인증(Entity Authentication)

암호 통신에서 개체 간 상호 인증을 통해 신원이 확인된 상대방과 기밀 데이터를 교환하므로 보안성을 향상시킬 수 있다. 금융거래에서 상호 인증은 이용자와 서버 간에 인증을 의미하며 키 교환 단계에서 상대방에 대한 인증은 반드시 함께 이루어져야 한다.

- 공격 시나리오: 이용자가 인증서를 보낼 때 이를 가로채고 공격자의 유효한 인증서로 금융회사 서버에 접근할 경우 금융회사가 상대의 인증서 유효성 및 신원을 확인하지 않고 인증서의 유효성만을 확인한다면 공격자는 이용자를 사칭할 수 있다. 반대로 공격자가 서버를 사칭해서 이용자의 정보를 수집할 수 있다.



<그림 24> 개체 간 상호 인증

서버 인증을 위해 인증서에 대한 유효성 검사만 하는 경우, 유효한 공격자의 인증서도 서버 인증에 성공할 수 있으므로 이용자는 접속한 서버의 인증서가 맞는지 반드시 확인해야 한다. 서버에 대한 인증은 서버 인증서로 수행하고 이용자에 대한 인증은 이용자 인증서 또는 아이디, 비밀번호 등으로 수행한다.

6.2 안전한 암호통신 프로토콜

❍ TLS 1.2 이상의 암호 프로토콜 사용

통신을 위해 사용되는 암호 알고리즘으로 TLS 1.0이나 1.1을 사용할 경우 BEAST(Browser Exploit Against SSL/TLS) 공격 등에 취약할 수 있다. 예를 들어, 공격자는 BEAST 공격으로 세션 ID를 추측하여 '세션 훔치기 공격'을 수행할 수 있다. 따라서 통신을 위한 암호 프로토콜은 공격으로부터 안전한 TLS 1.2 이상의 버전을 사용해야 한다[40].

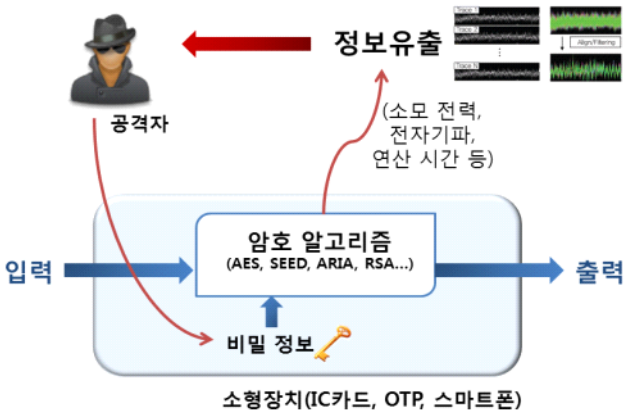
하지만 TLS 1.2도 POODLE, SLOTH, DROWN 공격 등에 취약할 수 있기 때문에 TLS 1.3인 RFC 8446[41]의 사용을 권장한다. TLS 1.3은 SHA-1, RC4 등 취약한 알고리즘을 제거하고, 모든 핸드셰이크 과정에 서명함으로써 중간자 공격에 안전하도록 하는 등 TLS 1.2보다 보안성을 향상시켰으며, 핸드셰이크 과정을 TLS 1.2보다 간소화하여 처리 속도를 향상시켰다.

40) 마이크로소프트(Microsoft), 구글(Google), 애플(Apple)은 2020년 초까지 인터넷 익스플로러(Internet Explorer), 크롬(Chrome) 등의 브라우저에 TLS 1.0과 TLS 1.1에 대한 지원을 점차 중단하기로 하였다.

41) https://datatracker.ietf.org/doc/rfc8446/?include_text=1

제7절 암호 알고리즘 구현 시 고려사항

IC카드, OTP, 스마트폰 등 소형장치에 탑재되는 암호 알고리즘은 제한된 하드웨어 성능과 메모리 용량 등 제약된 환경에서 구동된다. 따라서 동시에 수행할 수 있는 연산이 제한적이라는 특징이 있다. 이로 인해 소형장치에서는 일반 PC에 비해 암호 알고리즘이 구동될 때 발생한 정보로 비밀 정보를 유추하는 부채널 공격에 더 취약하다. 그림 25는 부채널 공격의 원리를 간단히 나타낸 그림이다.



<그림 25> 부채널 공격 원리

부채널 공격은 암호 모듈 개발자가 의도하지 않았지만, 연산을 수행할 때 물리적으로 발생한 전력, 전자기파, 연산 시간 등의 부채널 정보를 가지고 암호에 사용된 키를 찾아낸다.

금융권에서 많이 이용되는 IC카드, OTP 등의 소형장치는 IC칩 복제가 어렵다는 것에 모든 보안 강도를 신뢰하기보다는 암호 모듈의 경량화와 더불어 부채널 공격에 대응할 수 있는 하드웨어 또는 소프트웨어 적용에 대해 고려해야 한다. 2013년에는 금융보안연구원에서 스마트폰을

대상[42]으로 암호 연산 수행 시 발생한 전자기파를 이용하여 암호에 사용된 키를 유출하는 부채널 공격에 성공했다.

한편, 기본적으로 IC칩을 탑재한 스마트카드는 제조 시 칩의 보안성을 시험・인증하는 CC인증 등을 받고 있으며 스마트카드가 금융 거래용으로 이용될 경우에는 금융 거래용 IC카드의 보안성을 시험・인증하는 EMV 인증 등을 받은 제품이 배포되고 있다. 하지만 기본 탑재 암호 모듈이 아닌 자체 개발 응용 프로그램이나 국제 시험 인증에서 검토 대상이 아닌 암호 알고리즘에 대해서는 물리적 취약성을 별도로 검토할 필요가 있다.

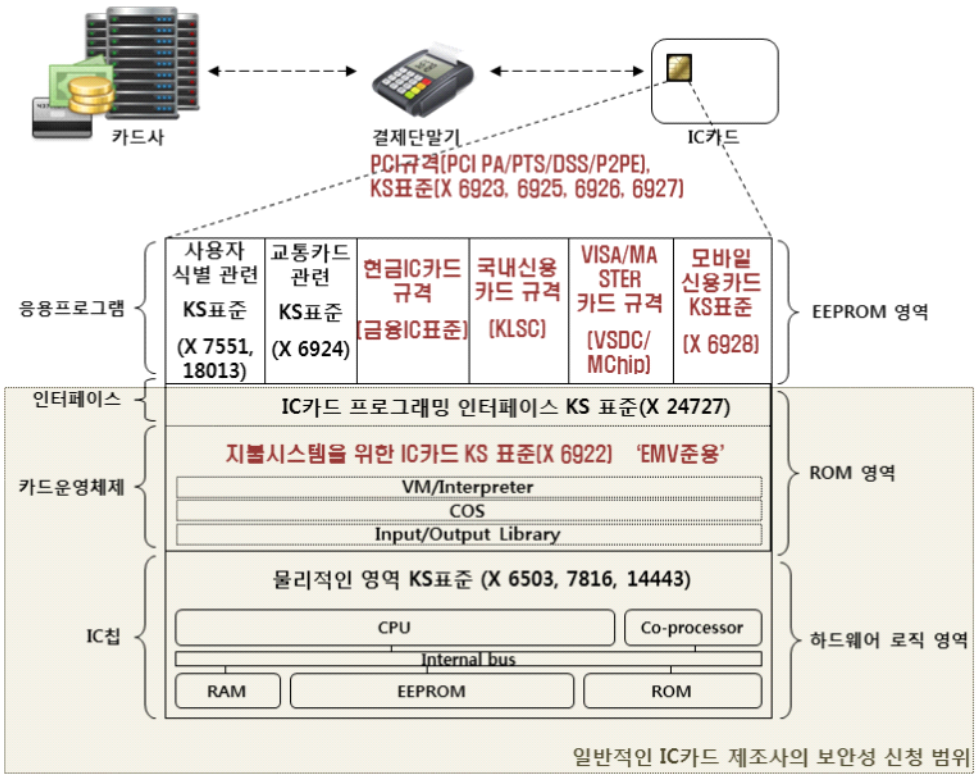
결국 금융권에서 인증된 스마트카드를 활용할 때에는 제품의 보안 레벨과 서비스에 이용된 알고리즘 및 암호 구현에 대한 취약점을 검토하였는지 확인할 필요가 있다. 또한 스마트카드가 금융권에서 사용되기 위한 많은 관련 규격과 보안 요구사항이 있으므로 관련 기준을 준수하는 제품을 선택해야 한다. 금융 서비스 관련 규격은 부록 1에서 다룬다.

42) 암호 연산을 수행한 앱은 공격 대상 알고리즘의 입출력에 이상이 없도록 개발된 테스트용 RSA 앱을 이용함

[부 록]

[부록1] 금융권 암호기술 적용 시 준수 규격

1. 권역별 IC카드 관련 준수 규격



<그림 26> IC카드 이용에 따른 보안성 시험 범위 및 참고 규격

USIM이나 신용카드 등 일반적인 IC카드는 제조 시, IC칩과 카드의 운영체제(COS) 및 인터페이스에 대해 CC 등의 보안성 심사를 받은 후, 응용 프로그램인 애플릿을 탑재한다. 그림 26은 IC카드의 이용에 따른 보안성 시험 범위와 금융 관련 규격에 대한 영역(붉은 글씨로 표시)을 나타낸다.

❍ 은행권 IC카드 관련 부분

- (국내) 현금 IC카드 「금융IC표준」: 은행 간 CD, ATM 등 자동화 기기 공동 사용을 위해 금융감독원, 금융결제원 및 은행이 공동 개발

[표 23] 금융 IC카드 표준

구분	구 성
폐쇄형	접촉식 IC카드의 물리적 및 전기적 특성, 비접촉식 IC카드의 물리적 및 전기적 특성, 파일 구조 및 명령어, 보안관리, 전자화폐
개방형	일반사항, 전자화폐, 금융공동망, 직불공동망, 공인인증서(서명용), 공인인증서(키 분배용)

❍ 카드권 IC카드 관련 부분

- (국내외) 신용 카드 「EMV」: 국제적 호환을 위해 Europay, Master, VISA가 공동으로 사용하는 국제 표준, 가장 기본이 되는 규격이며 결제 시스템에 대한 IC카드 관련 규격을 총 4권의 책으로 정의

[표 24] EMV 규격

구 분	내 용
Book1	IC카드 및 단말기 인터페이스 요구사항
Book2	보안 및 키 관리
Book3	응용 프로그램 규격
Book4	카드 소유자, 보조원, 매입 기관 인터페이스 요구사항

- (국외) 신용카드 및 단말기 결제 환경 「PCI-DSS(Payment Card Industry-Data Security Standard)」: 카드 정보 해킹, 도난, 분실 사고로부터 고객의 신용카드 정보를 보호하기 위하여 국제 브랜드사가 공동으로 마련하여 운영하는 카드 산업 보안 표준

[표 25] PCI 보안 표준 종류

구 분	내 용	보안대상	적용대상
PCI DSS	PCI 데이터 보안 표준	안전한 환경	가맹점, 서비스 제공자
PCI PTS	PIN 거래 보안 요구사항	PIN 입력 장치	제조사
PA-DSS	결제 응용프로그램 데이터 보안 표준	결제 응용 프로그램	소프트웨어 개발사
P2PE	PCI 단대단 암호 표준	거래 구간	전체

- (국내) 신용 카드 「KLSC(Korea Local Smart Card)」[43]: 여신금융협회에서 2009년 개발한 국내 전용 IC칩 신용카드 표준

- (국내) 카드사 가맹점 단말기 「POS/CAT 단말기 보안 표준」: 여신금융협회에서 2014년 제정한 보안 표준으로서 시험 및 단말기 인증 규격

43) 국내 환경에 적용하기 위한 기반환경이 마련되지 않아 상용화 되지 못하였고, 국내 신용카드사들은 변화된 기술 및 서비스 환경(쿠폰, 멤버십, 스탬프, 포인트 등 로열티 서비스 환경)에 사용 가능한 로컬 IC 신용카드 규격 개발이 요구됨

[표 26] 카드사 가맹점 단말기 보안 표준

구 분	내 용
POS 단말기 보안 표준	신용카드 가맹점에서 거래승인을 위해 사용되는 결제용 POS 단말기의 운영 환경 및 필수 보안 요구사항 정의
POS 단말기 보안 기능 시험 요구사항	POS 단말기 보안 규격의 세부 요구사항, 시험 요구사항 권고사항 등 정의
CAT 단말기 보안 표준	신용카드 가맹점에서 거래 승인을 위해 사용되는 결제용 CAT 단말기의 운영 환경 및 필수 보안 요구사항 정의
CAT 단말기 보안 기능 시험 요구사항	CAT 단말기 보안 규격의 세부 요구사항, 시험 요구사항 권고사항 등 정의

2. 대표적인 알고리즘의 보안 강도 및 참조 규격

금융권을 포함하여 가장 많이 이용되는 블록암호 알고리즘의 입출력 크기와 비밀키 크기, 보안 강도, 구현 시 참조 규격보안 강도는 표 27에 나타내었다.

[표 27] 대표적인 대칭키 블록암호 알고리즘

종류	입출력 길이 (비트)	비밀키 길이(비트)	보안 강도 (비트)	참조규격
SEED	128	128	128	TTA[44] TTAS.KO-12.0004/R1
AES	128	128	128	NIST[45] FIPS 197
		192	192	
		256	256	

금융권을 포함하여 가장 많이 이용되는 해시 알고리즘의 해시값 길이와 보안 강도, 구현 시 참조 규격은 표 28에 나타내었다.

[표 28] 대표적인 권고 해시 알고리즘

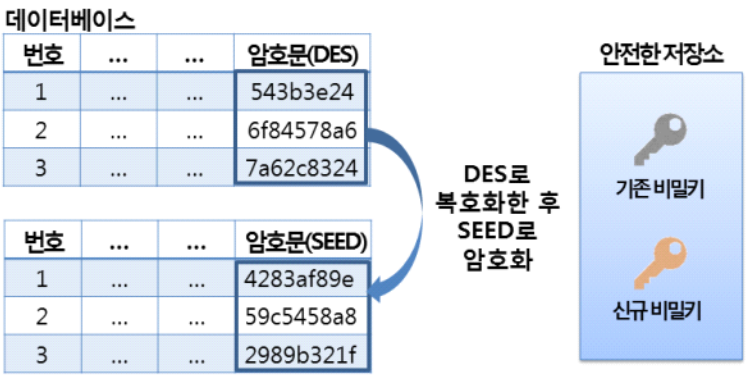
종류		출력값 길이 (비트)	보안 강도 (비트)	참조규격
SHA-2	224	224	112	NIST FIPS 180-4
	256	256	128	
	384	384	192	
	512	512	256	
SHA-3	224	224	112	NIST FIPS 202
	256	256	128	
	384	384	192	
	512	512	256	

44) TTA 자료검색(표준) 사이트: http://www.tta.or.kr/data/ttassearch.jsp
45) NIST: National Institute of Standards and Technology, http://csrc.nist.gov/publications/fips

2.1과 2.2에서는 각각 기존에 사용하고 있는 블록암호 알고리즘과 해시 알고리즘의 변경 예시[46]를 다룬다.

2.1 블록암호 알고리즘 변경 예시

기존에 사용하고 있는 블록암호 알고리즘이 DES나 TDES 등 보안 강도가 낮은 알고리즘이라면 이를 SEED와 같은 안전한 알고리즘으로 변경해야 한다. 이를 위해 안전한 장소에서 기존에 저장된 암호문을 복호화한 후, 기존의 비밀키와는 다른 새로운 비밀키를 생성하고 안전한 알고리즘으로 다시 암호화하여 저장하는 방법을 사용할 수 있다. 이때, 기존 및 신규 비밀키는 안전한 장소에 따로 보관해야 한다.



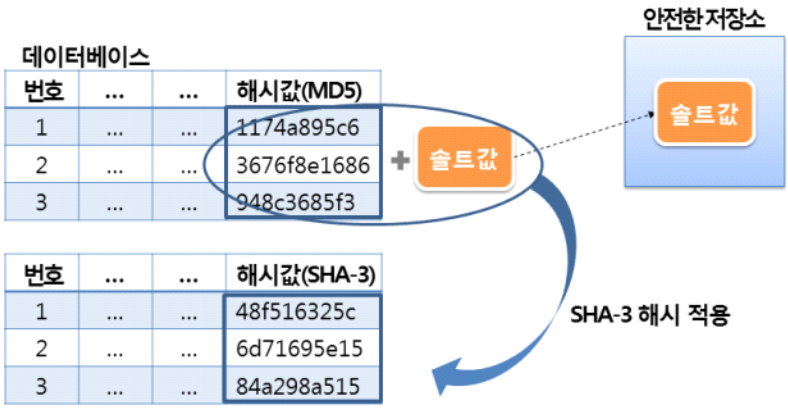
<그림 27> 블록암호 알고리즘 변경 예시

2.2 해시 알고리즘 변경 예시

기존에 사용하고 있는 해시 알고리즘이 MD5나 SHA-1 등 보안강도

46) KISA 암호기술 구현 안내서 '2.4 암호기술 적용 시나리오' 참조

가 낮은 알고리즘이라면 이를 SHA-3와 같은 안전한 알고리즘으로 변경해야 한다. 그러나 해시함수는 일방향 함수이므로 해시값으로부터 원문을 계산할 수 없다. 따라서 각 해시값에 대응되는 평문이 시스템에 새로 입력될 수 있도록 사용자에게 안내하거나, 새로운 값을 입력받기 전까지 이미 저장되어있는 모든 해시값을 안전한 해시 알고리즘으로 재연산 후, 평문이 입력될 때 재연산 된 해시값을 업데이트하는 방법을 활용할 수 있다. 업데이트를 위해 기존 해시값의 재연산 및 새로운 평문을 입력받아 해시 알고리즘을 변경할 때에는 솔트값을 추가하는 등의 안전한 해시 연산을 해야 한다.



<그림 28> 해시 알고리즘 변경 예시

3. 국내외 표준 암호 관련 라이브러리

국내외 표준 암호 알고리즘의 라이브러리를 제공하는 대표적인 사이트는 다음과 같다. 해당 사이트의 라이브러리 사용 시에는 저작권 및 최신 버전 여부를 확인해야 하며, 운용 시에는 해당 라이브러리의 취약점 및 패치를 주기적으로 확인하고 적용할 필요가 있다.

❍ (국내) 암호 라이브러리

KISA, http://seed.kisa.or.kr/	
국산 암호 (SEED, HIGHT, ARIA, LEA, LSH)	- SEED 128,256: 128비트 블록암호 알고리즘 - HIGHT: 64비트 블록암호 알고리즘 - ARIA: 128비트 블록암호 알고리즘 - LEA: 128비트 블록암호 알고리즘 - LSH: 해시 함수
스마트폰용 암호 라이브러리	- 운영체제(Android, iOS, Windows Mobile) - 블록암호 알고리즘(SEED, HIGHT) - 운영모드(CBC) - 전자서명(KCDSA)

❍ (해외 및 국제 표준) 암호 라이브러리

BeeCrypt, http://beecrypt.sourceforge.net/	
오픈소스 형태의 멀티플랫폼 라이브러리	- 엔트로피 소스, 난수 생성기, 블록암호, 해시함수, 메시지 인증 코드 등 제공
Botan, http://botan.randombit.net/	
C++ 기반 암호 라이브러리	- SSL/TLS, X.509 인증서, ECDSA, AES, BCM 등 제공

Crypto++, http://www.cryptopp.com/	
C++ 기반 암호 라이브러리	- 대칭키 암호(고속 스트림 암호, SEED 등) - 운영모드(GCM, CCM, ECB, CBC, CFB, CTR 등) - 메시지 인증 코드(HMAC, CMAC, CBC-MAC 등) - 해시 알고리즘(SHA2, SHA3 등) - 공개키 암호(RSA, DSA 등) - 키 교환 알고리즘(DH, ECDSA 등) 등
GNU Crypto, http://www.gnu.org/software/gnu-crypto/	
Java 기반 암호 툴	- AES, 블록암호운영모드, SHA, RSA-PSS, DES 등
Libgcrypt, http://www.gnu.org/software/libgcrypt/	
코드기반 암호화 라이브러리	- 대칭키 암호(SEED 등), 해시알고리즘(SHA2 등), 공개키 암호(RSA, ECDSA 등), 암호 연산을 지원하는 함수 등
MCrypt, http://mcrypt.sourceforge.net/	
C기반 암호 라이브러리	- 블록암호(AES 등), 스트림암호(RC6 등), 운영모드(CBC, CFB, CTR 등), 공개키암호(RSA 등) 제공
MIRACL, http://www.certivox.com/miracl	
고성능용 암호 라이브러리	- 임베디드 환경이나 모바일 스마트 기기에서 호환 가능한 암호 SDK(Software Development Kit, 소프트웨어 개발 도구)제공 - AES, RSA, ECC, FPE 등 지원
MSDN, http://msdn.microsoft.com/ko-kr/library/92f9ye3s(v=vs.110).aspx	
마이크로소프트 개발자 네트워크[47]	- .NET Framework 암호화 모델 - 대칭키 암호, 공개키 암호, 디지털서명, 해시함수 등 제공

Nettle, http://www.lysator.liu.se/~nisse/nettle/	
객체지향언어용 암호 툴(C++, Python, Pike 등)	- 해시함수(SHA2 등), 암호함수(AES등), 운영모드(CTR 등), 메시지 인증코드(HMAC 등), 키 유도 함수, 공개키 암호(RSA, DSA 등) 제공

47) 마이크로소프트용 도구, 서비스 및 기술을 사용하여 작업할 때 필요한 API 집합

[부록2] 암호 관련 금융IT 보안 컴플라이언스 통제사항

○ 암호 관련 금융 IT 보안 컴플라이언스 통제사항 및 준수 근거[48]는 표 29에 나타내었다.

[표 29] 암호 관련 금융 IT 보안 컴플라이언스 통제사항 및 준수 근거

통제 항목	통제사항	준수 근거
암호 프로그램 담당자 지정	암호프로그램에 대하여 담당자를 지정하고, 담당자 이외의 이용을 통제하여야 한다.	**전자금융거래법 제21조(안전성의 확보의무)** ② 금융회사 등은 전자금융거래의 안전성과 신뢰성을 확보할 수 있도록 전자적 전송이나 처리를 위한 인력, 시설, 전자적 장치, 소요경비 등의 정보기술부문, 전자금융업무 및 「전자서명법」에 의한 인증서의 사용 등 인증방법에 관하여 금융위원회가 정하는 기준을 준수하여야 한다. **전자금융감독규정 제31조(암호프로그램 및 키 관리 통제)** ① 금융회사 또는 전자금융업자는 암호프로그램에 대하여 담당자 지정, 담당자 이외의 이용 통제 및 원시프로그램(source program) 별도 보관 등을 준수하여 유포 및 부당 이용이 발생하지 않도록 하여야 한다.
암호키 관리	암호 및 인증시스템에 적용되는 키에 대하여 주입, 운용, 갱신, 폐기에 대한 절차 및 방법에	**전자금융거래법 제21조(안전성의 확보의무)** ②금융회사 등은 전자금융거래의 안전성과 신뢰성을 확보할 수 있도록 전자적 전송이나 처리를 위한 인력, 시설, 전자적 장치, 소요경비 등의 정보기술부문, 전자금융업무 및 「전자서명법」에 의한 인증서의 사용 등 인증방법에 관하여 금융위원회가 정하는

48) 금융보안원, 「금융 IT 보안 컴플라이언스 가이드」, 2017.10

	따라 안전하게 관리하여야 한다.	기준을 준수하여야 한다. **전자금융감독규정 제31조(암호프로그램 및 키 관리 통제)** ② 금융회사 또는 전자금융업자는 암호 및 인증시스템에 적용되는 키에 대하여 주입·운용·갱신·폐기에 대한 절차 및 방법을 마련하여 안전하게 관리하여야 한다.
비밀번호 암호화 보관	비밀번호 보관 시 암호화하여야 한다.	**전자금융거래법 제21조(안전성의 확보의무)** ② 금융회사 등은 전자금융거래의 안전성과 신뢰성을 확보할 수 있도록 전자적 전송이나 처리를 위한 인력, 시설, 전자적 장치, 소요경비 등의 정보기술부문, 전자금융업무 및 「전자서명법」에 의한 인증서의 사용 등 인증방법에 관하여 금융위원회가 정하는 기준을 준수하여야 한다. **전자금융감독규정 제32조(내부사용자 비밀번호 관리)** 금융회사 또는 전자금융업자는 내부사용자의 비밀번호 유출을 방지하지 위하여 다음 각 호의 사항을 정보처리시스템에 반영하여야 한다. 1. 담당업무 외에는 열람 및 출력을 제한할 수 있는 접근자의 비밀번호를 설정하여 운영할 것 2. 비밀번호는 다음 각 목의 사항을 준수할 것 나. 비밀번호 보관 시 암호화
이용자 비밀번호 암호화 보관	정보처리시스템 및 전산자료에 보관하고 있는 이용자의 비밀번호를 암호화하여 보관하여야 한다.	**전자금융거래법 제21조(안전성의 확보의무)** ② 금융회사 등은 전자금융거래의 안전성과 신뢰성을 확보할 수 있도록 전자적 전송이나 처리를 위한 인력, 시설, 전자적 장치, 소요경비 등의 정보기술부문, 전자금융업무 및 「전자서명법」에 의한 인증서의 사용 등 인증방법에 관하여 금융위원회가 정하는 기준을 준수하여야 한다. **전자금융감독규정 제33조(이용자 비밀번호 관리)** ① 금융회사 또는 전자금융업자는 정보처리시스템

		및 전산자료에 보관하고 있는 이용자의 비밀번호를 암호화하여 보관하며 동 비밀번호를 조회할 수 없도록 하여야 한다. 다만, 비밀번호의 조회가 불가피하다고 인정되는 경우에는 그 조회사유·내용 등을 기록·관리하여야 한다.
인증방법 사용기준	전자금융거래의 종류·성격·위험수준 등을 고려하여 안전한 인증방법을 사용하여야 한다.	**전자금융거래법 제21조(안전성의 확보의무)** ② 금융회사 등은 전자금융거래의 안전성과 신뢰성을 확보할 수 있도록 전자적 전송이나 처리를 위한 인력, 시설, 전자적 장치, 소요경비 등의 정보기술부문, 전자금융업무 및 「전자서명법」에 의한 인증서의 사용 등 인증방법에 관하여 금융위원회가 정하는 기준을 준수하여야 한다. **전자금융감독규정 제37조(인증방법 사용기준)** 금융회사 또는 전자금융업자는 전자금융거래의 종류성격·위험수준 등을 고려하여 안전한 인증방법을 사용하여야 한다.
암호화 통신	전화 등 거래수단 성격상 암호화가 불가능한 경우를 제외한 전자금융거래는 암호화 통신을 사용하여야 한다. (예외: 전용선을 사용하는 경우로 「전자금융감독규정」 제36조의 자체 보안성심의를 실시한 경우)	**전자금융거래법 제21조(안전성의 확보의무)** ② 금융회사 등은 전자금융거래의 안전성과 신뢰성을 확보할 수 있도록 전자적 전송이나 처리를 위한 인력, 시설, 전자적 장치, 소요경비 등의 정보기술부문, 전자금융업무 및 「전자서명법」에 의한 인증서의 사용 등 인증방법에 관하여 금융위원회가 정하는 기준을 준수하여야 한다. **전자금융감독규정 제34조(전자금융거래 시 준수사항)** 금융회사 또는 전자금융업자는 전자금융거래와 관련하여 다음 각 호의 사항을 준수하여야 한다. 1. 전화 등 거래수단 성격상 암호화가 불가능한 경우를 제외한 전자금융거래는 암호화 통신을 할 것 (다만, 전용선을 사용하는 경우로서 제36조의 규정에 따라 보안성심의를 받은 경우에는 그러하지 아니하다)
보안프	전자금융거래에서	**전자금융거래법 제21조(안전성의 확보의무)** ② 금융

로그램 설치 및 거래프로그램 무결성 검증 방법 제공	이용자에게 제공하거나 거래를 처리하기 위한 전자금융거래프로그램(거래 전문포함)의 위·변조 여부 등 무결성을 검증할 수 있는 방법을 제공하여야 한다.	회사 등은 전자금융거래의 안전성과 신뢰성을 확보할 수 있도록 전자적 전송이나 처리를 위한 인력, 시설, 전자적 장치, 소요경비 등의 정보기술부문, 전자금융업무 및 「전자서명법」에 의한 인증서의 사용 등 인증방법에 관하여 금융위원회가 정하는 기준을 준수하여야 한다. **전자금융감독규정 제34조(전자금융거래 시 준수사항)** 금융회사 또는 전자금융업자는 전자금융거래와 관련하여 다음 각 호의 사항을 준수하여야 한다. 5. 금융회사 또는 전자금융업자는 전자금융거래에서 이용자에게 제공하거나 거래를 처리하기 위한 전자금융거래 프로그램(거래전문포함)의 위·변조 여부 등 무결성을 검증할 수 있는 방법을 제공할 것
암호화 정보 해독 및 중요 데이터 변경 금지	전자금융거래를 위한 외부주문등의 경우 금융기관과 이용자 간 암호화정보 해독 및 원장 등 중요 데이터 변경을 금지하여야 한다	**전자금융거래법 제40조(외부주문등에 대한 감독 및 검사)** ① 금융회사 및 전자금융업자는 전자금융거래와 관련하여 전자금융보자업자와 제휴, 위탁 또는 외부주문(이하 이 조에서 "외부주문등"이라 한다)에 관한 계약을 체결하거나 변경하는 때(전자금융보조업자가 다른 전자금융보조업자와 외부주문등에 관한 계약을 체결하거나 변경하는 때를 포함한다)에는 전자금융거래의 안전성 및 신뢰성과 금융회사 및 전자금융업자의 건전성을 확보할 수 있도록 금융위원회과 정하는 기준을 준수하여야 한다. **전자금융감독규정 제60조(외부주문등에 대한 기준)** ① 금융회사 또는 전자금융업자는 전자금융거래를 위한 외부주문등의 경우에는 다음 각 호의 사항을 준수하여야 한다. 2. 금융회사와 이용자 간 암호화정보 해독 및 원장 등 중요 데이터 변경 금지

접근 매체 위·변조 등에 대비한 보안 대책 수립	전자금융거래를 위한 외부주문등의 경우 접근매체 위·변조, 해킹, 개인정보유출 등에 대비한 보안대책을 수립하여야 한다.	**전자금융거래법 제40조(외부주문등에 대한 감독 및 검사)** ① 금융회사 및 전자금융업자는 전자금융거래와 관련하여 전자금융보자업자와 제휴, 위탁 또는 외부주문(이하 이 조에서 "외부주문등"이라 한다)에 관한 계약을 체결하거나 변 **전자금융감독규정 제60조(외부주문등에 대한 기준)** ① 금융회사 또는 전자금융업자는 전자금융거래를 위한 외부주문등의 경우에는 다음 각 호의 사항을 준수하여야한다. 4. 접근매체 위·변조, 해킹, 개인정보유출 등에 대비한 보안대책 수립

[부록3] 암호기술 활용 시 고려사항(요약)

❍ 본 금융 부문 암호 활용 가이드라인(3장)에서 기술된 암호기술 활용 시 고려사항을 정리하면 아래와 같다.

[표 30] 본 가이드에 기술된 암호기술 활용 시 고려사항

분류		항목
운영 고려사항	암호기술 분류 및 용도	- 암호화 대상과 목적에 적절한 기술인가? - (가용성) 속도, 구현모듈 사이즈 등 - 기밀성, 무결성, 인증 등
	암호기술 운영 시 고려사항	- 관련 암호 정책을 수립하였는가? - 암호기술이 조합될 때 전체의 보안 강도를 확인했는가? - 시스템 예상 수명을 고려하였는가?
암호 알고리즘 선택	보안 강도	- 알고리즘 키 길이에 따른 보안 강도 확인 - 알고리즘 논리적 취약성 여부 확인 - 새로운 취약성 주기적 확인
	알고리즘의 안전성 유지 기간	- 키 길이에 따른 알고리즘 안전성 유지기간 확인 - 알고리즘의 보안 수명 확인
운영모드		- 패딩 취약성 확인 - 적용된 운영모드 취약성 확인
의사난수 생성기		- 난수 기본성질 만족여부 확인
키 관리	생성	- 키 사용 유효기간 확인, 예측 불가한지, 위조 할 수 없는 난수 - 키 생성 시 이용되는 의사난수생성기는 암호용으로 사용하는 것인가?

키 관리	분배	- 저장 시 키를 물리적으로 안전한 장치에 저장하는가? - 키 재료에 대해 기밀성, 무결성을 제공하는가?
	저장	- 암호문과 키를 따로 저장하는지 또는 키 암호화 키로 암호화하여 보관하는가?
	사용	- 암호키 분리정책이 있는지 확인 - 암호키 접근제어 정책이 있는지 확인
	백업, 복구	- 주기적으로 백업되는가? - 백업된 키 정보는 암호화되었는가? - 비밀번호는 별도로 관리되는가? - 키 복구 방안 마련되어있는가? - 복구가 허가된 사용자만 복원 할 수 있는가?
	교체(갱신)	- 내부 정책에 따라 기간마다 교체하는가? - 키 유도 시 역변환은 불가한가?
	폐기	- 폐기 시 다시 복구 시킬 수 없는가? - 모든 복사본에 대한 폐기인가? - 보관된 키로 보호된 자료가 더 이상 필요 없는가?
암호 통신 프로토콜 설계		- 취약점에 대한 대응책이 있는가?(재전송 공격, 반사공격, 세션 훔치기 공격, 중간자 공격) - 동일 암호키 확인이 가능한가? - 전방향 안전성을 보장하는가? - 개체 간 상호 인증을 수행하는가?
암호 구현		- 최소 요구 보안레벨에 준하는 인증을 받은 제품인가? - 서비스에 이용된 알고리즘이 인증 대상에 포함되어있는가?

[부록4] 기타 암호 알고리즘

❍ 포스트 양자암호(Post Quantum Cryptography)[49]

양자컴퓨터의 고속 연산 알고리즘인 양자알고리즘은 현대 암호에 적용된 수학적 난제의 일부를 합리적인 수준(연산량, 비용, 시간 등)의 환경에서 해독할 수 있다. 대표적인 양자알고리즘은 검색에 탁월한 성능을 갖는 Grover 알고리즘과 인수분해 문제를 빠른 시간 안에 효과적으로 풀 수 있는 Shor 알고리즘이다. 양자 컴퓨터가 상용화 되면 양자알고리즘을 사용하여 대칭키와 단방향 해시함수의 입력 정보를 찾아 낼 수 있고 인수분해 문제에 기반한 공개키 암호의 비밀키가 유출 될 수 있다. 따라서 양자알고리즘으로 인한 암호 해독에도 안전한 ‘포스트 양자암호’가 등장하게 되었다.

[표 31] 양자알고리즘이 현대 암호에 미치는 영향

구분	현대 암호 알고리즘	목적	영향 및 대응방안
대칭키 암호	AES, DES, ARIA, SEED 등	암호화	o Grover 알고리즘의 빠른 검색으로 대칭키를 찾아냄 o 대칭키의 길이를 2배 증가
단방향 함수	SHA256 등	해시 함수	o Grover 알고리즘의 빠른 검색으로 입력정보를 찾아냄 o 출력 길이를 3배 증가
공개키 암호	RSA	서명, 키설정	o Shor 알고리즘의 빠른 인수분해로 비밀키를 찾아냄 o 새로운 구조에 기반한 알고리즘 개발이 필요
	ECDSA, ECDH, EC-KCDSA (타원곡선암호)	서명, 키교환	

49) 금융보안원, 양자컴퓨팅과 포스트 양자암호 동향, 2017.12

	DSA, KCDSA (유한체암호)	서명, 키교환	

포스트 양자암호는 양자의 특성을 이용한 큐비트 대신 고전비트(0 또는 1)를 사용하여 일반 컴퓨팅 환경에서도 사용이 가능한 암호로, 현재 개발된 양자알고리즘으로는 풀 수 없는 문제들을 기반으로 한다. 대표적인 포스트 양자암호 알고리즘으로는 표 32와 같이 Rainbow, QC-MDPC 등이 있으며 각 포스트 양자암호 알고리즘의 단점을 줄이고 장점을 부각시키기 위한 연구가 계속해서 수행되고 있다.

[표 32] 포스트 양자암호의 대표적인 알고리즘과 특징

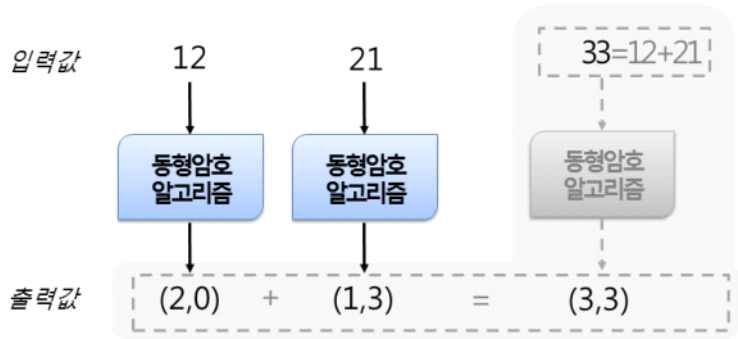
분야	설명	대표적인 알고리즘	특징
다항식 기반	수학적으로 특정 조건을 만족하는 집합에서 다변수 함수를 푸는 것의 어려움을 기반으로 하는 암호 시스템	Rainbow	o 서명 길이가 짧음 o 키 길이가 길
코드 기반	오류수정 코드[50]에 기반으로 하는 암호 시스템	QC-MDPC, Wild McEliece	o 암·복호화 속도 빠름 o 키 길이가 길
격자 기반	특정 조건을 만족하는 격자 구조에서의 계산 문제를 푸는 것의 어려움을 기반으로 하는 암호 시스템	SS-NTRU, NTRU Prime, LWE-Frodo	o 다양한 응용 존재 o 속도가 빠름 o 변수 설정 문제
Isogeny[51] 기반	유한체 상에서 정의된 타원곡선들 사이의 Isogeny를 계산하는 것의 어려움을 기반으로 하는 암호 시스템	SIDH	o 키 길이가 짧음 o 연산 속도가 비교적 느림
해시함수 기반	해시 함수의 안전성을 기반으로 하는 전자서명 시스템[52]	XMSS, SPHINCS	o 안전성 증명 가능 o 서명 길이가 길

50) 메시지와 추가 정보를 같이 수신하게끔 하여 전송 중에 생기는 오류에도 올바른 메시지를 복원할 수 있도록 하는 기법

미국표준기술연구소(NIST)에서는 포스트 양자암호를 표준화하기 위한 프로젝트[53]를 진행하고 있다. 해당 프로젝트는 포스트 양자암호 표준 제안서를 받아 2018년부터 제출된 신기술을 분석하여 검증하고 2024년 안에 최종 포스트 양자암호 표준 초안을 제정할 예정이다.

❍ 동형암호(Homomorphic Encryption)

동형암호는 암호화된 상태에서도 연산을 수행할 수 있는 암호로서 암호화된 데이터로 연산을 수행하여도 원본 데이터로 연산을 수행하여 암호화한 것과 같은 결과를 얻을 수 있다. 예를 들어, 그림 31과 같이 덧셈 연산 수행이 가능한 동형암호 알고리즘이 입력값을 '(5로 나눈 나머지, 6으로 나눈 나머지)' 의 형태로 변환한다고 할 때, 12와 21을 암호화하면 각각 (2,0), (1,3)의 출력값을 갖는다. 두 출력값을 각각 더하면 (3,3)이 되며 이는 입력값 12와 21을 더한 33에 동형암호 알고리즘을 적용한 값과 동일하다.



<그림 29> 동형암호 개념 설명 예시

51) [수학용어] Isogeny(아이소제니, 등원사상): 특정 조건을 만족하는 집합 사이의 특수한 함수
52) 각각의 적절한 해시함수를 이용하여 서로 다른 해시 기반 서명 체계를 갖춘 시스템으로서 사용되는 해시함수의 충돌 저항성에 의해 안전성을 보장
53) Post-Quantum Cryptography Standardization(https://csrc.nist.gov/Projects/Post-Quantum-Cryptography)

동형암호는 일부 연산만 수행할 수 있는 부분동형암호(Partial homomorphic encryption)와 모든 연산을 수행할 수 있는 완전동형암호(Fully homomorphic encryption)로 나눌 수 있다. 부분동형암호의 대표적인 알고리즘으로는 덧셈 연산만 가능한 Paillier와 곱셈 연산만 가능한 ElGamal 등이 있으며, 완전동형암호는 IBM의 동형암호[54]와 서울대학교의 '혜안(HEAAN)[55]' 등이 있다.

동형암호는 암호화된 데이터를 복호화하지 않고 연산할 수 있기 때문에 데이터 분석 및 활용을 위해 암호를 해제하는 과정에서 발생할 수 있는 데이터 노출 위험이 적다. 현재는 타 암호에 비해 연산 속도 향상 및 메모리 사용량을 개선하기 위한 연구가 진행 중이다.

54) Gentry C, et al. A fully homomorphic encryption scheme. 2009.
55) Cheon, J.H., et al. Homomorphic Encryption for Arithmetic of Approximate Numbers. 2017.

본 가이드는 다음 분들의 감수를 마쳤습니다.

2019년 1월

충남대학교	교	수	류재철
한국과학기술원	교	수	김용대
이니텍	소	장	이은배
NSHC	소	장	윤기순
금융보안원	팀	장	이수미

금융부문 암호기술 활용 가이드

발 행	2019년 1월
발 행 인	김 영 기
발 행 처	금융보안원
주 소	경기도 용인시 수지구 대지로 132