보안토큰 기반 공인인증서 저장형식 기술규격

HSM Storage Format Specification for Accredited Certificate

v1.3

2016년 1월

KISA 한국인터넷진흥원
Korea Internet & Security Agency

목 차

보안토큰 기반의 공인인증서 저장형식 기술규격
HSM Storage Format Specification for Accredited Certificate

1. 개 요

본 규격에서는 전자서명법에 따라 구축된 공인전자서명인증체계 공인인증서비스에서 사용되는 보안토큰 저장매체에 대한 공인인증서 저장 기술을 규정한다.

2. 규격의 구성 및 범위

본 규격은 공인전자서명인증체계에서의 보안토큰 기반의 공인인증서 저장정보 형식 등에 대해 정의하고 있다.

3. 관련 표준 및 규격

3.1 국외 표준 및 규격

[PKCS15] RSA Laboratories, PKCS#15(2000), *Cryptographic Token Information Format Standard*

[RFC2119] IETF, RFC2119, *Key words for use in RFCs to Indicate Requirement Levels,* March 1997

3.2 국내 표준 및 규격

[KS X ISO/IEC 7816]	ID 카드 - 접촉식 IC카드
[KS X ISO/IEC 14443]	ID 카드 - 비접촉식 IC카드 - 근접식카드
[PKCS#1]	RSA 암호 표준
[PKCS#11],[KCAC.TS.HSMU]	암호 토큰 인터페이스 표준
[PKCS#15]	암호 토큰 정보 형식 표준
PKI 관련 스마트카드 기술규격(국내)	
금융IC카드 표준(국내)	

3.3 기타

해당사항 없음

4. 정의

본 규격에서 사용하는 용어의 정의는 제4장에서 정한 것을 제외하고는 관련 법령 등이 정하는 바에 의한다.

4.1 전자서명법 용어 정의

본 규격에서 사용된 다음의 용어들은 전자서명법 및 동법 시행령, 공인인증기관의 시설 및 장비 등에 관한 규정(미래창조과학부 고시)에 정의되어 있다.

가) 공인인증서

4.2 용어의 정의

가) 보안토큰(HSM) : 전자서명생성키 등 비밀정보를 안전하게 저장·보관하기 위하여 키 생성, 전자서명 생성 등이 기기 내부에서 처리되도록 구현된 하드웨어 기기

나) 무선통신단말 API : 무선통신을 지원하는 보안토큰에 대한 응용프로그래밍 인터페이스

다) 보안토큰 구동프로그램 : 보안토큰 API를 구현한 프로그램을 말하며, 보안토큰과 가입자 소프트웨어간의 인터페이스를 담당

4.3 용어의 효력

본 규격에서 사용된 다음의 용어들은 전자서명인증체계 가입자 소프트웨어가 보안토큰을 이용함에 있어서 보안토큰 정보형식에 대한 구현 정도를 의미하는 것으로 [RFC2119]를 준용하며 다음과 같은 의미를 지닌다.

가) 해야한다, 필수이다, 강제한다 (기호 : M)
반드시 준수해야 한다.

나) 권고한다 (기호 : R)

보안성 및 상호연동을 고려하여 준수할 것을 권장한다.

다) 할 수 있다, 쓸 수 있다 (기호 : O)

주어진 상황을 고려하여 필요한 경우에 한해 선택적으로 사용할 수 있다.

라) 권고하지 않는다 (기호 : NR)

보안성 및 상호연동을 고려하여 사용하지 말 것을 권장한다.

마) 금지한다, 허용하지 않는다 (기호 : X)

반드시 사용하지 않아야 한다.

바) 언급하지 않는다, 정의하지 않는다 (기호 : -)

준수 여부에 대해 기술하지 않는다.

5. 약어

가) USIM : Universal Subscriber Identity Module, 범용사용자식별모듈
나) AID : Application IDentifier
다) ADF : Application Dedicated File
라) APDU : Application Protocol Data Unit
마) CBC : Cipher Block Chaining
바) CLA : CLAss byte
사) CSN : Card Serial Number
아) DF : Dedicated File
자) DK : Derivation Key
차) EF : Elementary File
카) FCI : File Control Information
타) FID : File IDentifier
파) HEX : HEXadecimal
하) IV : Initialization Vector
갸) INS : INStruction byte
냐) Lc : Length of command data
댜) Le : Length of expected data
랴) Lf : Linear fixed record
먀) MAC : Message Authentication Code
뱌) MK : Master Key
샤) P1-P2 : Parameter byte
야) PIN : Personal Identification Number

자) PIX : Proprietary application Identifier Extension
차) R_{ICC} : Random number of ICC (IC Card)
카) RFU : Reserved for Future Use
타) RID : Registered application provider IDentifier
파) RSA : Rivest Shamir Adleman
하) SW1-SW2 : Status byte
거) TR : TRansparent 파일구조
너) Var : Variable

6. 보안토큰 기반의 공인인증서 저장형식 요구사항

보안토큰 기반의 공인인증서, 전자서명키, 비밀키 등 객체에 대한 저장위치, 저장형식 요구사항은 [금융IC카드 보안토큰 표준]을 준용할 것을 권고한다.

또한 사용자가 타 저장매체에 저장된 전자서명생성키 및 공인인증서를 보안토큰 또는 USIM에 저장할 수 있도록 보안토큰으로 가져오기 기능을 제공할 것을 권고한다.

7. 무선통신 지원 보안토큰 기반의 공인인증서 저장형식 요구사항

무선통신 지원 보안토큰의 공인인증서, 전자서명키, 비밀키 등 객체에 대한 저장형식 요구사항은 【부록. 무선통신 지원 보안토큰 기반의 공인인증서 저장형식 규격】을 준용할 것을 권고한다.

부록의 규격은 무선통신단말 지원 보안토큰을 위해 「금융IC카드 표준(보안토큰)」를 바탕으로 공인인증서 기술 규격에 맞게 수정/보완하였으며, 무선통신단말 지원 보안토큰 표준이 제정되기 전까지 한시적으로만 제공 합니다.

부록. 무선통신 지원 보안토큰 기반의 공인인증서 저장형식 규격

1. 무선통신 지원 보안토큰 개요

1.1 요구기능

□ 암호전용 프로세서(Crypto-coprocessor)를 탑재

□ RSA 1,024비트 이상을 지원

□ 전자서명 및 암호키 복호를 위해 사용되는 개인키를 안전하게 저장하고 외부로 유출되지 않아야 하며, 갱신을 지원해야 함.

□ 난수 생성은 전자서명인증체계에 맞는 난수 생성기를 사용

1.2 설계

□ 파일구조

o 디렉터리 형태의 계층구조(Tree Structure)
o 인증서 전용파일(DF) 아래에 요소파일(EF) 존재
파일관리정보(FCI), 사용자PIN, 키 정보, 개인키, 인증서, 공개키

□ 파일관리정보(FCI)

o FCI 파일은 DF의 파일정보를 수록
o FCI 템플릿, DF Name, FCI 전용템플릿, 파일정보로 구성

□ 사용자PIN

o 사용자 PIN은 로컬 PIN을 사용
o 인스톨 파라미터에서 재시도 가능 횟수 입력

□ 키 정보파일(Key Information File)

o 암호알고리즘 ID, 키 ID로 구성

o 암호알고리즘 ID : RSA Value값, 키 ID : 공개키의 해쉬값

□ 키 생성(Key Generation)

o 키는 전자서명 생성 및 암호키 복호화에 필요한 개인키와 전자서명 검증에 필요한 공개키를 카드 내 암호알고리즘을 통해 생성

□ 난수 생성 (Random Number Generation)

o 난수는 카드내 난수 생성기(Random Number Generator)에 의해 추측이 불가능하도록 임의적으로 생성

□ 전자서명 생성

o 전자서명은 카드 내에서 사용자 소프트웨어로부터 전달받은 서명할 메시지(M)을 사용자 개인키(RK_U)로 암호화하여 생성 <전자서명(S)= $E(M, RK_U)$>

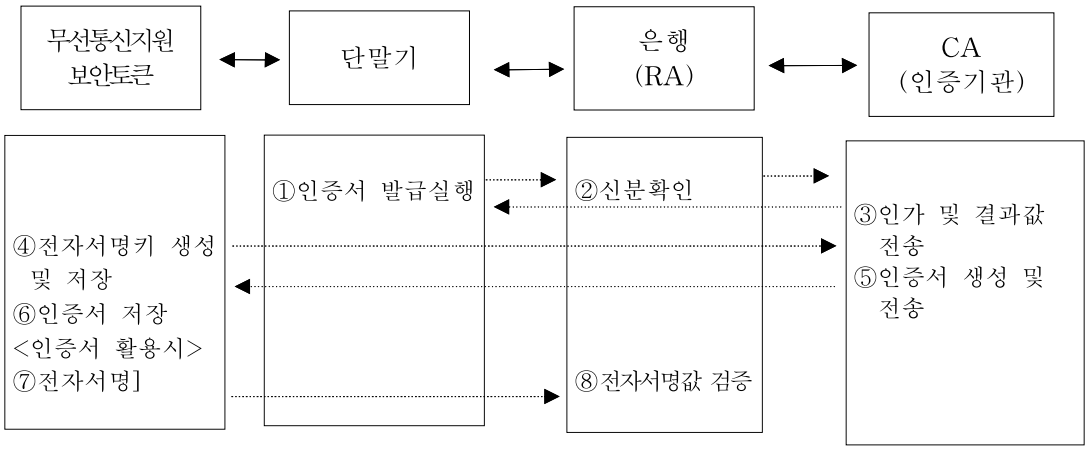
1.3 명령어

o 본 규격에서 사용되는 명령어는 다음과 같다.

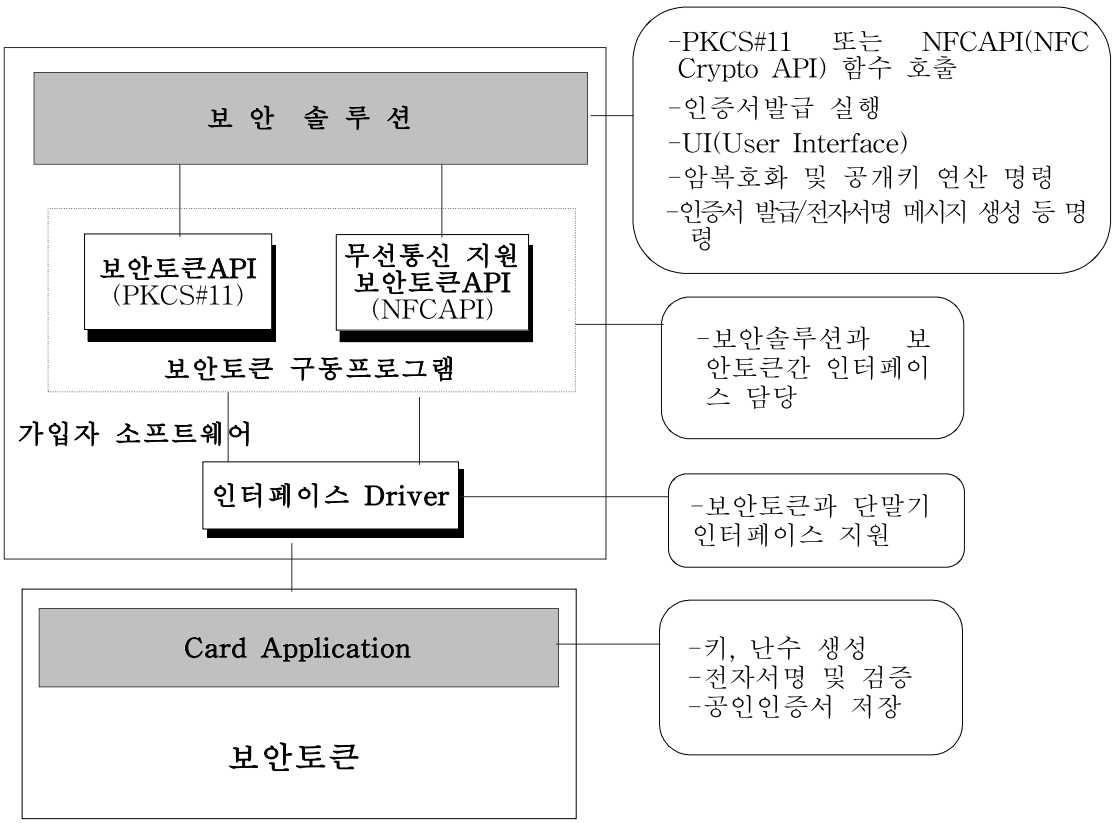
명령어	내 용
SELECT FILE	- DF Name(AID)에 의한 애플리케이션(DF) 선택 - 파일식별자에 의한 요소파일(EF) 선택 - 애플리케이션 선택 성공시, FCI데이터를 응답
READ BINARY	- EF(TR)파일의 내용을 Read
UPDATE BINARY	- 명령어 APDU에서 주어진 Data를 EF(TR)파일에 Update
VERIFY	- 단말기로부터 전송된 검증 데이터와 카드내에 저장된 참조 데이터(예:PIN)를 비교·검증
PUT KEY	- 현재 선택되어진 애플릿하에 존재하는 KEY/PIN을 Write
GET DATA	- 애플릿으로부터 필요한 정보를 읽어오는데 사용
PUT DATA	- 특정 정보를 애플릿의 특정영역에 기록하는데 사용
GET CHALLENGE	- Security 관련 Procedure에서 사용하기 위하여 난수 발생을 요구
CLEAR	- EF를 지정하여 EF의 내용을 초기 상태('00')로 함
GENERATE PUBLIC KEY PAIR	- 카드에서 서명용 또는 암호키 분배용 공개키와 개인키를 생성하고 카드 내에 저장하도록 하기 위해 사용
INITIALIZE CRYPTO	- INITIALIZE CRYPTO 명령어는 PERFORM CRYPTO 명령어를 수행하기 위한 파라미터들을 설정
PERFORM CRYPTO	- PERFORM CRYPTO 명령어는 암호 연산 수행
STORE PRIVATE KEY	- 비대칭키 알고리즘의 개인키를 토큰 내부에 저장하기 위해 사용
MUTUAL AUTHENTICATE	- 카드에 저장된 발행기관 키를 이용하여 발행기관을 검증하고, 발행기관이 보안토큰을 검증하기 위한 인증 데이터를 생성

<표 1-1> 보안토큰(공인인증서 기반 거래용) 명령어

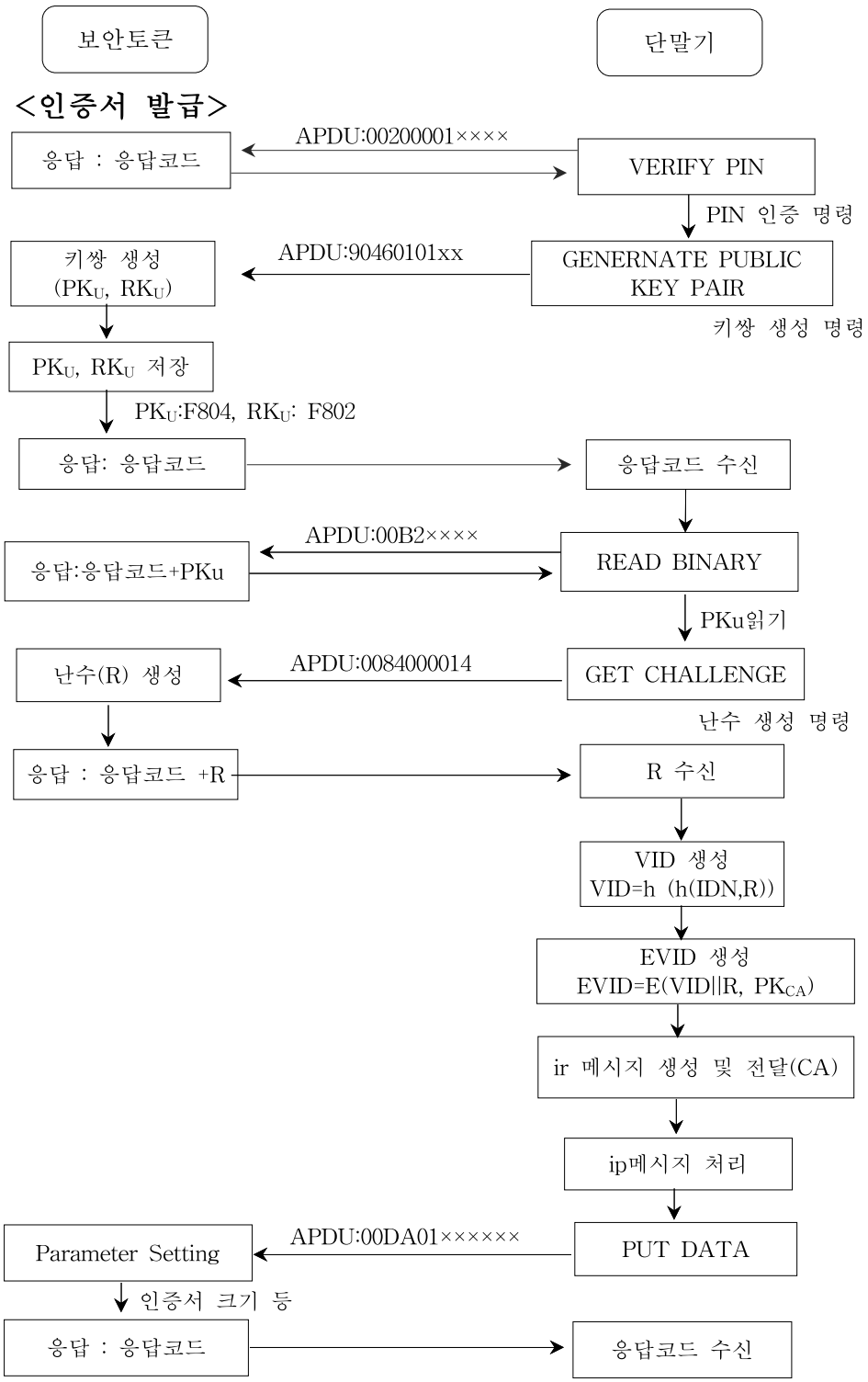
1.4 무선통신을 지원하는 보안토큰 거래 모델

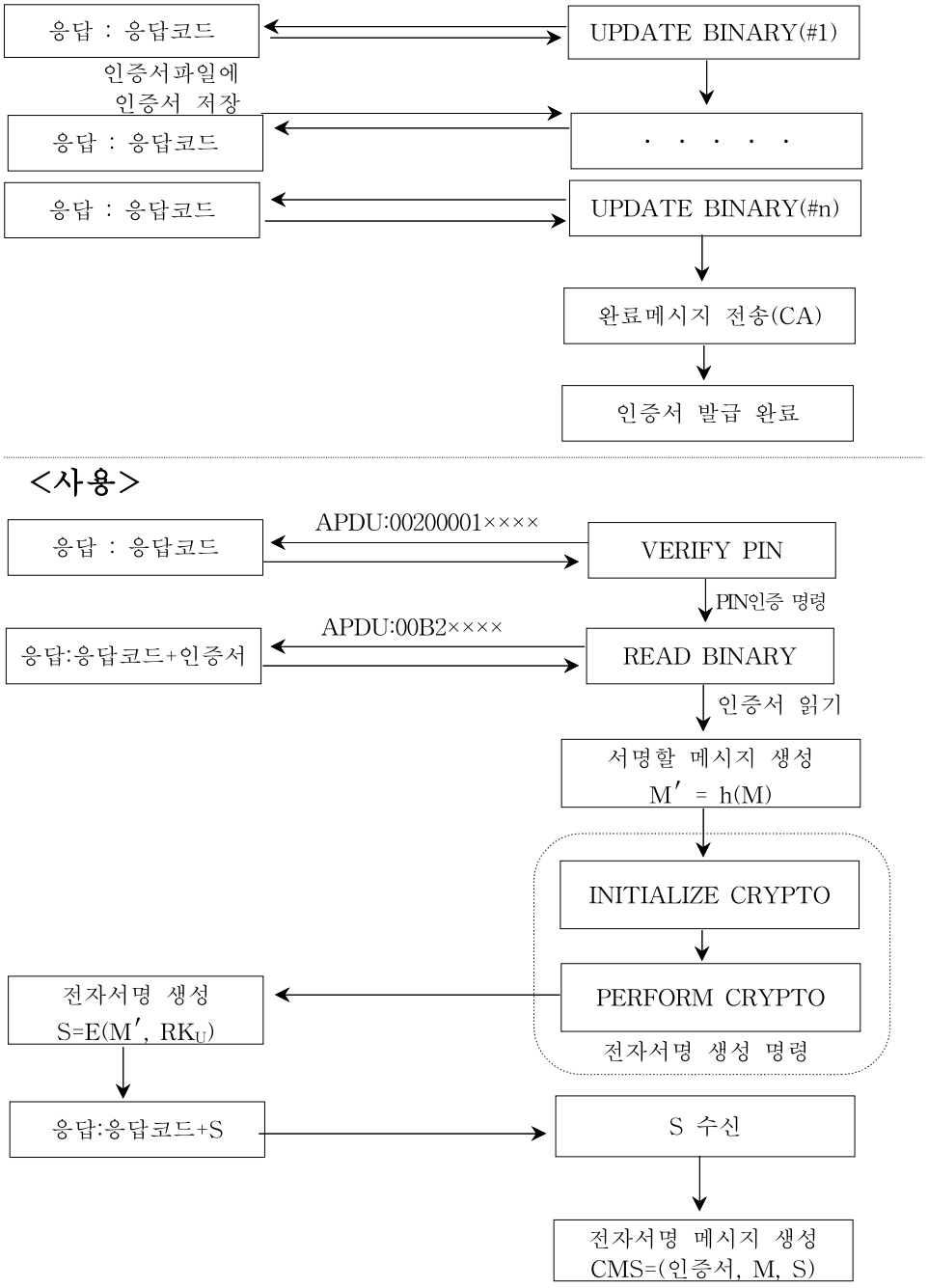


<그림 1-1> 인증서 발급 및 활용



<그림 1-2> 보안토큰과 보안토큰 API간 Interface


보안토큰
단말기
<인증서 발급>
응답 : 응답코드
APDU:00200001××××
VERIFY PIN
PIN 인증 명령
키쌍 생성
(PKU, RKU)
APDU:90460101xx
GENERNATE PUBLIC KEY PAIR
키쌍 생성 명령
PKU, RKU 저장
PKU:F804, RKU: F802
응답: 응답코드
응답코드 수신
APDU:00B2××××
응답:응답코드+PKu
READ BINARY
PKu읽기
난수(R) 생성
APDU:0084000014
GET CHALLENGE
난수 생성 명령
응답 : 응답코드 +R
R 수신
VID 생성
VID=h (h(IDN,R))
EVID 생성
EVID=E(VID||R, PKCA)
ir 메시지 생성 및 전달(CA)
ip메시지 처리
Parameter Setting
APDU:00DA01××××××
PUT DATA
인증서 크기 등
응답 : 응답코드
응답코드 수신



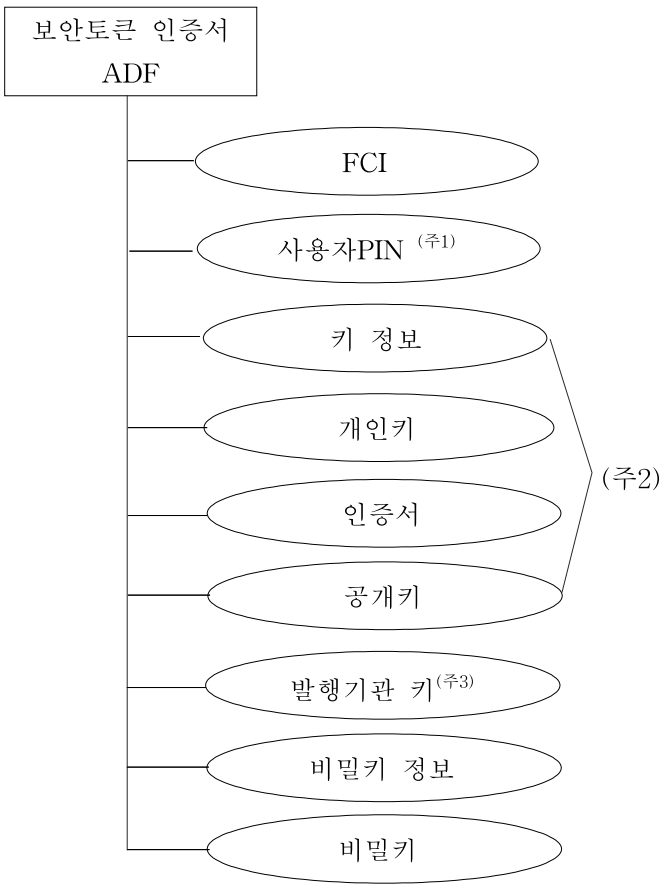
<그림 1-3> 프로토콜 개괄도

PK_U : 사용자 공개키, RK_U : 사용자 개인키, R : 난수(본인확인을 위한 랜덤수),
VID : 가상식별번호, IDN : 식별번호, EVID : 암호화된 가상식별번호,
h ( ) : 해쉬함수, E( ) : 암호화 함수, VID : 가상식별번호(본인확인용-인증서에 포함),
PK_{CA} : CA 공개키, ir : 인증서 발급요청메시지, ip : 인증서발급 응답메시지,
S : 전자서명값, M : 원본 메시지, M' : 서명할 메시지(원본 메시지 해쉬값),
CMS : 암호메시지구문(Cryptographic Message Syntax)

2. 보안토큰(공인인증서 기반 거래용) AID

구 분	RID	PIX
직 불 DF	D410650990	3010
금융공동망 DF	D410650990	0010
전자화폐 DF	D410650990	0020
공인인증서(서명용) DF	D410650990	0030
공인인증서(키분배용) DF	D410650990	0040
보안토큰 DF	**D410650990**	**0080**

3. 보안토큰(공인인증서 기반거래용) 파일구조



(주1) 사용자 PIN은 개방형카드(Open Platform)의 Local PIN을 사용하되, 최소 6자리 이상 설정하고 나머지 부분은 NULL값으로 채워서 사용한다.
* PIN 재시도 가능횟수는 인스톨파라미터에서 정의한다.

(주2) 발급기관 선택에 따라 N개까지 발급할 수 있다.

(주3) 발급기관이 보안토큰에 분배한 키로 PUT KEY 명령어를 통해 분배 가능하다.
* 키 분배가 이루어져 발행기관 키가 설정된 후에 인증(MUTUAL AUTHENTICATE) 명령어 사용이 가능하며, 카드의 발행기관 키 설정 여부는 GET DATA 명령어를 통해 확인할 수 있다.

4. 인증서 DF

인증서 DF에 저장되는 요소파일은 다음과 같고, 발행기관에서는 초기 발급 시 각 요소 파일에 NULL을 Write하여 발급한다.

AID				D4106509900080
내 용	**UPDATE ACCESS 조건**	**READ ACCESS 조건**	**파일식별자**	**조건**
FCI 파일	Forbidden	FREE	F81E	필수
키정보 파일1	PIN 인증	FREE	F801	필수
개인키 파일1	Forbidden	Forbidden	F802	필수
인증서 파일1	PIN 인증	FREE	F803	필수
공개키 파일1	PIN 인증	FREE	F804	필수
...	...	...		선택
키정보 파일N	PIN 인증	FREE	F801+4*(N-1)	선택
개인키 파일N	Forbidden	Forbidden	F802+4*(N-1)	선택
인증서 파일N	PIN 인증	FREE	F803+4*(N-1)	선택
공개키 파일N	PIN 인증	FREE	F804+4*(N-1)	선택
비밀키 정보파일	PIN 인증	FREE	F81D	필수
비밀키 파일	Forbidden	Forbidden	F81F	필수

4.1 FCI(File Control Information) 파일

파일 식별자	F81E
파일 구조	TR
파일 크기	27 Bytes

Tag	**항 목**	**크기**	**Value**	**비고**	**조건**
6F	FCI Template	25			필수
84	DF Name	7	D4106509900080	Cn	필수
A5	FCI Proprietary Template	14			필수
BF 0C	파일정보	11(주)	파일정보	Cn	필수

(주) 파일정보(11)
= 발행기관식별자(3) + 버전(1) + 인증서 최대 저장 개수(1) + 카드지원 기능(1) + 사용자인증방법(1) + 기본암호화알고리즘(1) + RFU(3)

(예) 발행기관식별자(3바이트) : BCD로 표기된 ‘산업분류코드(금융:065)+발행기관코드’로 인스톨 파라미터를 통해 설정
* 금융결제원 발급 애플릿(금융결제원 발행기관코드: 099)의 경우 ‘06 50 99’ 값을 가짐

표준화버전 (1바이트) : 01, 02, 03(2013년 추가), 04(2015년 추가)
인증서 최대 저장 개수 (1바이트) : 인스톨 파라미터를 통해 설정

카드지원기능(1바이트) : 애플릿에서 내부적으로 설정

Bit 정의	내 용	비 고
1 x x x x x x x	RSA CRT Private Key 생성 지원	미지원시 해당 bit값은 0
x 1 x x x x x x	RSA Private Key생성 지원	
x x x x x x x 1	RSA 1024bit Key 지원	
x x x x x x 1 x	RSA 2048bit Key 지원	

사용자인증방법(1바이트) : 장치에서 사용자인증을 수행하는 방법으로 인스톨 파라미터를 통해 설정

Bit 정의	내 용
x x x x - - - -	사용자인증 알고리즘 정보(미지원시 해당 bit값은 0)
- - - - - - - 0	보안토큰 단말기로부터 사용자 인증
- - - - - - - 1	구동프로그램에서 입력받은 PIN을 이용한 사용자인증
기타	RFU

* 사용자인증 알고리즘 정보는 별도 정의

기본암호화알고리즘(1바이트) : 인스톨 파라미터를 통해 설정
- 알고리즘 : 3DES(10), SEED(20), AES-128(40) 중 선택 1지원
- 운영모드 : CBC 지원
- PUT KEY, MUTUAL AUTHENTICATE 등의 명령어에서 대칭키 알고리즘 사용시 활용

4.2 키 정보파일1

파일 식별자			F801		
파일 구조			TR		
파일 크기			365 Bytes		
번호	**항 목**	**크기**	**Value**	**비고**	**조건**
1	알고리즘 ID	1	RSA:10	HEX	필수
2	키 ID	32	공개키의 해쉬값	HEX	필수
3	SUBJECT	256	DER 인코딩된 소유자 이름	HEX	선택
4	LABEL	64	객체 설명	HEX	선택
5	공개키 FLAG	4	공개키에 대한 속성 FLAG(주1)	HEX	선택
6	개인키 FLAG	4	개인키에 대한 속성 FLAG(주2)	HEX	선택
7	RFU	4		HEX	선택

(주1) 공개키 FLAG (0x00 : 지원안함, 0x01 : 지원)

바이트	4	3	2	1
속성	암호화 지원여부	부가형 전자서명 검증 지원여부	복원형 전자서명 검증 지원여부	키 암호화 지원여부

(주2) 개인키 FLAG (0x00 : 지원안함, 0x01 : 지원)

바이트	4	3	2	1
속성	복호화 지원여부	부가형 전자서명 지원여부	복원형 전자서명 지원여부	키 복호화 지원여부

4.3 개인키 파일1

파일 식별자			F802		
파일 구조			TR		
파일 크기			642 Bytes		
번호	**항 목**	**크기**	**Value**	**비고**	**조건**
1	개인키	642	사용자의 개인키 정보(주)	HEX	필수

(주)

구 분	Data 항목	크기	Value	비 고
RSA Private	Key Type	1	02	-
	Key length	1	XX	bytes/4("0x20"-1024bits key, "0x40"-2048bits key)
	Modulus	N	XX	N=128(1024bits key) N=256(2048bits key)
	Private exponent	N	XX	Private exponent
RSA Private CRT	Key Type	1	03	
	Component length	1	XX	bytes/4("0x20"-1024bits key, "0x40"-2048bits key)
	Prime-1	N	XX	Prime-1(P) N=64(1024bits key) N=128(2048bits key)
	Prime-2	N	XX	Prime-2(Q)
	Exponent-1	N	XX	exponent-1(DP)
	Exponent-2	N	XX	exponent-2(DQ)
	Coefficient	N	XX	Coefficient(QP)

4.4 인증서 파일1

파일 식별자			F803		
파일 구조			TR		
파일 크기			2080 Bytes		
번호	**항 목**	**크기**	**Value**	**비고**	**조건**
1	인증서	2080	사용자의 인증서	HEX	필수

4.5 공개키 파일1

파일 식별자			F804		
파일 구조			TR		
파일 크기			262 Bytes		
번호	**항 목**	**크기**	**Value**	**비고**	**조건**
1	인증서	262	사용자의 공개키(주)	HEX	필수

(주)

Data 항목	크기	Value	비 고
Key Type	1	01	
Key length	1	XX	bytes/4("0x20"-1024bits key, "0x40"-2048bits key)
Modulus	N	XX	N=128(1024bits key) N=256(2048bits key)
Public exponent[1)]	4	XX[2)]	Public exponent

1) Public exponent(e) : Prime Number < ex) 65537 미만 숫자 사용 불가 >

2) GENERATE PUBLIC KEY PAIR명령의 Command Data값(4bytes)이 기록되며, GENERATE PUBLIC KEY PAIR명령의 Command Data가 존재하지 않는 경우 (Lc==0인 경우), (0x00010001)을 기록한다.

4.6 비밀(대칭)키 정보파일

파일 식별자			F81D		
파일 구조			TR		
파일 크기			69 * N(주1) Bytes		
번호	**항 목**	**크기**	**Value**	**비고**	**조건**
1	알고리즘 ID 1	1	비밀(대칭)키1의 알고리즘 ID(주2)	HEX	필수
2	LABEL 1	64	비밀(대칭)키1의 객체설명	HEX	필수
3	FLAG 1	4	비밀(대칭)키1에 대한 속성 FLAG(주3)	HEX	필수
...					
3*(N-1)+1	알고리즘 ID N	1	비밀(대칭)키N의 알고리즘 ID	HEX	선택
3*(N-1)+2	LABEL N	64	비밀(대칭)키N의 객체설명	HEX	선택
3*(N-1)+3	FLAG N	4	비밀(대칭)키N에 대한 속성 FLAG	HEX	선택

(주1) 인증서 최대저장개수와 동일(인스톨 파라미터로 입력)
(주2) 10: 3DES, 20: SEED, 30: ARIA-128, 31: ARIA-192, 32: ARIA-256,
40: AES-128, 41: AES-192, 42: AES-256
(주3) 비밀키 FLAG (0x00 : 지원안함, 0x01 : 지원)

바이트	4	3	2	1
속성	암호화 지원여부	복호화 지원여부	키 암호화 지원여부	키 복호화 지원여부

5.7 비밀(대칭)키 파일

파일 식별자			F81F		
파일 구조			TR		
파일 크기			32 * N Bytes		
번호	**항 목**	**크기**	**Value**	**비고**	**조건**
1	비밀키1	32	비밀(대칭)키1 데이터	HEX	필수
...					
N	비밀키N	32	비밀(대칭)키N 데이터	HEX	선택

* 데이터의 크기가 32bytes 보다 작을 경우 왼쪽부터 저장한다.

5. 명령어

5.1 SELECT FILE

5.1.1 기능 및 특징

○ SELECT FILE 명령어는 다음과 같이 사용된다.

— DF Name에 의한 애플리케이션(DF) 선택
— 파일식별자에 의한 파일 선택

○ DF Name에 의한 애플리케이션(DF) 선택 시, FCI파일의 내용이 응답된다.

5.1.2 Command message

코 드	내 용
CLA	00
INS	A4
P1	Selection 제어 - (주) 참조
P2	00
Lc	후속 데이터 필드 길이 - (주) 참조
Data	P1에 따름 - (주) 참조
Le	최대 예상 응답 데이터 길이

<표 5-1> SELECT FILE Command APDU

(주) Selection 제어 P1과 Lc의 코딩

P1								Lc	의 미
b8	b7	b6	b5	b4	b3	b2	b1		
0	0	0	0	0	0	0	0	02	파일 식별자에 의한 EF선택(데이터필드=식별자)
0	0	0	0	0	1	0	0	07	DF Name에 의한 선택 (데이터필드=DF Name)
기타 다른 값									RFU

5.1.3 Response message

코 드	내 용	길 이
DATA	FCI (DF Name에 의한 애플리케이션 선택의 경우)	27
	Empty (P1=00일 경우)	0
SW1, SW2	COMPLETION CODE	2

<표 5-2> SELECT FILE Response APDU

코 드	내 용
SW1-SW2	Success : 90 00 Warning 62 84 - ISO에 따라 포맷되지 않은 FCI Error 6A 82 - 존재하지 않는 파일 6A 86 - 오류 파라미터(P1,P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6E 00 - 지원되지 않는 CLA

<표 5-3> 에러 코드

5.2 READ BINARY

5.2.1 기능 및 특징

○ READ BINARY 응답 메시지는 EF파일의 내용을 나타내는 명령어이다.

○ 단, 이 명령어는 Non-Transparent 구조의 요소파일(EF)에는 적용하지 못한다.

○ Le = 00일 때 최대 256Byte를 Read한다.

5.2.2 Command message

코 드	내 용
CLA	00
INS	B0
P1-P2	P1의 b8=1이면 P1의 b5~b1은 Short 파일식별자 P2는 Read할 Offset을 표시 P1의 b8=0이면 P1-P2는 Read할 Offset
Lc	Empty
Data	Empty
Le	Read될 바이트의 수

<표 5-4> READ BINARY Command APDU

5.2.3 Response message

코 드	내 용	길 이
DATA	Data Read(Le bytes)	Var
SW1, SW2	COMPLETION CODE	2

<표 5-5> READ BINARY Response APDU

코 드	내 용
SW1-SW2	Success : 90 00 Warning 62 81 - 응답 데이터의 일부 훼손 Error 67 00 - 오류 길이 69 81 - 파일구조와 호환되지 않는 명령어 6A 82 - 발견되지 않는 파일 69 82 - 시큐리티 Status 불충족 6A 86 - 오류 파라미터 (P1, P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6C xx - 오류 길이 (xx는 정확한 길이를 표시) 6E 00 - 지원되지 않는 CLA

<표 5-6> 에러 코드

5.3 UPDATE BINARY

5.3.1 기능 및 특징

○ UPDATE BINARY 명령어 메시지는 명령어 APDU에서 주어진 Data를 EF 파일에 UPDATE 한다.

○ 단, 이 명령어는 Non-Transparent 구조의 요소파일(EF)에는 적용하지 못한다.

○ Lc의 최대 Byte 수는 'FF(255 Byte)'이다.

5.3.2 Command message

코 드	내 용
CLA	00
INS	D6
P1-P2	P1의 b8=1이면 P1의 b5~b1은 Short 파일식별자 P2는 Read할 Offset을 표시 P1의 b8=0이면 P1-P2는 Update할 Offset
Lc	Update할 Data의 길이
Data	Update 되어질 Data
Le	Empty

<표 5-7> UPDATE BINARY Command APDU

5.3.3 Response message

코 드	내 용
SW1-SW2	Success : 90 00 Error 65 81 - Memory Failure(Updating 실패) 67 00 - 오류 길이 69 81 - 파일구조와 호환되지 않는 명령어 69 88 - MAC이 틀림 6A 82 - 발견되지 않는 파일 69 82 - 시큐리티 Status 불충족 6A 81 - 지원되지 않는 기능 6A 82 - 발견되지 않는 파일 6A 86 - 오류 파라미터 (P1, P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6E 00 - 지원되지 않는 CLA

<표 5-8> 에러 코드

5.4 VERIFY

5.4.1 기능 및 특징

○ VERIFY 명령어는 단말기로부터 전송된 검증 데이터와 카드 내에 저장된 참조 데이터(예:PIN)를 비교・검증하는 명령어이다.

5.4.2 Command message

코 드	내 용
CLA	00
INS	20
P1	00 또는 80~8F(reserved)
P2	01(사용자 PIN)
Lc	00 또는 검증 데이터의 길이
Data	Empty 또는 검증 데이터
Le	Empty

<표 5-9> VERIFY Command APDU

5.4.3 Response message

코 드	내 용
SW1-SW2	Success : 90 00 Warning 63 CX - 카운터 X는 앞으로 남은 재시도 가능 횟수- (주)참조 Error 69 84 - PIN이 Block되어 있음 6A 86 - 오류 파라미터 (P1,P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6E 00 - 지원되지 않는 CLA

(주) 검증 데이터가 틀렸거나, Lc가 00인 경우, PIN검증이 성공한 상태이면 9000을 응답하고, PIN검증이 안된 상태이면 63CX(X : 재시도 가능횟수)를 응답한다.

<표 5-10> 에러 코드

5.5 PUT KEY

5.5.1 기능 및 특징

○ 현재 선택되어진 애플릿 하에 존재하는 KEY/PIN을 Write하는 명령어이다.
○ PIN 갱신은 PIN인증 또는 MUTUAL AUTHENTICATE 이후에 가능하다. 단, 발행기관 키가 설정되어 있고 PIN 초기화(PIN 미갱신) 상태에서는 MUTUAL AUTHENTICATE 이후에만 가능하다.
○ KEY 갱신은 오직 Secure Messaging에 의해서만 가능하다.

5.5.2 Command message

코 드	내 용
CLA	KEY : A4, PIN : 90
INS	24
P1	KEY : 01, PIN : 00 또는 80~8F(reserved)
P2	01
Lc	KEY/PIN 데이터 길이(KEY:14, PIN:XX)
Data	KEY(주1)/PIN 데이터
Le	Empty

<표 5-11> PUT KEY Command APDU

(주1) MK : 발행기관 마스터키
DK : 카드에 신규로 주입될 발행기관 키
Key : 현재 카드에 저장되어 있는 키

Data 암호화 방법

○ DK 생성<Server(HOST)>
DK = E(CSN ‖ CSN^{-1} , MK)

○ DK 암호화
Ciphered Data = CBC(DK, Key)

Server(HOST)의 MAC 생성 방법

○ MAC 생성
MAC=CBC(APDU헤더 ‖ R_{ICC}* ‖ Ciphered Data ‖ Padding Data**, Key)

* R_{ICC}는 Get Challenge명령으로 얻어진 상위 8byte가 사용되며, 한번 사용된 난수는 다시 사용될 수 없으므로 MAC생성이 요구될 때마다 Get Challenge명령이 선행되어야 함
** Paddinng 규칙은 금융IC카드 규격을 준용

5.5.3 Response message

코 드	내 용
SW1-SW2	Success : 90 00 해당 PIN/KEY File에 Update 성공 Error 69 82 - 시큐리티 Status 불충족 69 88 - MAC이 틀림 6A 86 - 오류 파라미터 (P1, P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6E 00 - 지원되지 않는 CLA

<표 5-12> 에러 코드

5.6 GET DATA

5.6.1 기능 및 특징

○ GET DATA 명령어는 카드로부터 필요한 정보를 읽어오는데 사용한다.
○ 인증서 X를 위한 R을 얻기 위해 사용자 PIN이 제출되어야 한다.

5.6.2 Command message

코 드	내 용
CLA	00
INS	CA
P1	01(기타 : RFU)
P2	(주) 참조(기타 : RFU)
Lc	Empty
Data	Empty
Le	(주) 참조

<표 5-13> GET DATA Command APDU

(주) : P2, Le의 코딩

P2	내 용	Le
11	토큰 정보(주1) -Label(32) -ManufacturerID(32) -Model(16) -Card Serial Number(8) -MaxPinLen(1) -MinPinLen(1) -hardware version(2) -firmware version(2)	5E
4X	인증서X를 위한 R(주2)	14
5X	인증서X의 크기(주2)	02
91	현재 저장된 인증서 개수(01) + 인증서 유무(주3)	08
92	현재 발행기관 키/PIN 상태(주4)	01
F0~FF	Reserved	-

(주1) 토큰정보
- 애플릿을 인스톨 한 후 Put Data 명령어를 이용하여 토큰정보를 입력한다.
- 토큰정보의 입력은 키 발급이전에 이루어져야 한다.
- 토큰정보는 별도의 저장장소에 저장하여야 한다.

(주2) 인증서를 위한 R(41~4N)과 인증서의 크기(51~5N)는지정된 인증서 저장개수에 따라 유효하지 않은 값에 대해 에러 처리된다. 예를 들어 인증서가 첫 번째, 세 번째 위치에 저장되어 있는 경우 유효한 인증서를 위한 R은 41, 43이며 유효한 인증서의 크기는 51, 53이다.

(주3) 현재 저장된 인증서 개수(1) + 인증서 유무 및 종류(07)
현재 저장된 인증서 개수 (1바이트): 0 - N
- 인증서 유무 및 종류 : 각 인증서에 대해 한 바이트가 사용되며 각 바이트의 내용은 다음과 같다. (00: 없음, 01: 서명용 범용, 02: 서명용 용도제한용(은행, 보험용, 신용카드용) 03: 서명용 용도제한용(조달청용), 04: 서명용 용도제한용(증권, 보험용), 05: 키 분배용 범용, FF : 종류는 표기하지 않으나 인증서 존재, 06-FE : RFU)
- 인증서가 존재할 경우 개인키/공개키 또한 존재한다고 간주한다.

(주4) 발행기관 키/PIN 설정 상태는 다음과 같이 반환한다.
- 상위 4bit : 발행기관 키/PIN의 상태를 표기

Bit 정의	내 용
x - - -	발행기관 키 상태 표기 - 0 : 미설정 상태, 1 : 키 분배를 통해 발행기관 키 설정된 상태
- - - x	PIN 상태 표기 - 0 : 초기 PIN 상태, 1 : PIN이 업데이트 된 상태
기타	RFU

- 하위 4bit : PIN 검증 유무를 표시하며, PIN이 검증된 경우 0001(각 bit 단위)로 반환된다.

5.6.3 Response message

코 드	내 용	길 이
DATA	응답될 데이터	Var
SW1, SW2	COMPLETION CODE	2

<표 6-14> GET DATA Response APDU

코 드	내 용
SW1-SW2	Success : 90 00 Warning 62 81 - 응답데이터 일부분의 훼손 Error 69 82 - 시큐리티 Status 불충족 6A 86 - 오류 파라미터(P1, P2) 6A 87 - P1 P2와 일치되지 않는 Lc 6C XX - 오류 길이(XX는 정확한 길이 표시) 6E 00 - 지원되지 않는 CLA

<표 5-15> 에러 코드

5.7 PUT DATA

5.7.1 기능 및 특징

○ PUT DATA 명령어는 특정 정보를 카드의 특정영역에 기록하는데 사용한다.

○ 사용자 PIN이 제출된 경우에 실행될 수 있다.

○ 토큰정보에 대해서는 발급 시 한 번만 설정이 가능하다.

5.7.2 Command message

코 드	내 용
CLA	00
INS	DA
P1	01
P2	(주) 참조
Lc	(주) 참조
Data	Write될 데이터
Le	Empty

<표 5-16> PUT DATA Command APDU

(주) : P2 및 Lc

P2	내 용	Lc	재갱신
11	토큰 정보* -Label(32) -ManufacturerID(32) -Model(16) -Card Serial Number(8) -hardware version(2) -firmware version(2)	5C	*불가*
4X	인증서X을 위한 R (X: 1～최대인증서저장개수)	14	가능
5X	인증서X의 크기 (X: 1～최대인증서저장개수)	02	가능
91	현재 저장된 인증서 개수(01) + 인증서 유무 및 종류(07)	08	가능

* 인증서를 위한 R(41~4N)과 인증서의 크기(51~5N)는 인증서 저장개수에 따라 유효하지 않은 값에 대해 에러 처리된다. 예를 들어, 인증서 저장개수가 2인 경우 유효한 인증서를 위한 R은 41, 42이며 유효한 인증서의 크기는 51, 52이다.

** PUT DATA명령 수행 시 전달받은 '현재 저장된 인증서 개수'의 값이 '인증서 유무 및 종류'에 지정된 인증서 개수와 일치하는지 검증하여야 함.

5.7.3 Response message

코 드	내 용
SW1-SW2	Success : 90 00 Error 65 81 - Memory Failure 67 00 - 오류 길이 69 82 - 시큐리티 Status 불충족 69 85 - 사용 조건을 만족하지 못함 (재갱신 시도 시) 6A 80 - 명령어 데이터 오류 ('현재 저장된 인증서 개수'와 '인증서 유무 및 종류에 지정된 개수' 불일치) 6A 86 - 오류 파라미터(P1, P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6E 00 - 지원되지 않는 CLA

<표 5-17> 에러 코드

5.8 GET CHALLENGE

5.8.1 기능 및 특징

○ GET CHALLENGE 명령어는 외부에서 사용하거나 Security 관련 Procedure에서 사용하기 위하여 Challenge (Random Number)의 발생을 요구한다.

○ Security 관련 Procedure에서 사용하기 위하여 생성된 난수는 다음 명령어 수행 시까지 유효하다.

○ 난수는 사용한 후 다시 사용할 수 없다.

5.8.2 Command message

코 드	내 용
CLA	00
INS	84
P1	00(외부사용을 위하여 생성) 01(Security 관련 Procedure에서 사용하기 위하여 생성)
P2	00
Lc	Empty
Data	Empty
Le	XX (P1=00 : 최대 256 바이트 생성) (P1=01 : 최대 32 바이트 생성)

<표 5-18> GET CHALLENGE Command APDU

5.8.3 Response message

코 드	내 용	길 이
DATA	Random Number	Le
SW1,SW2	COMPLETION CODE	2

<표 5-19> GET CHALLENGE Response APDU

코 드	내 용
SW1-SW2	Success : 90 00 Error 6A 86 - 오류 파라미터 P1-P2 6A 87 - P1, P2와 일치되지 않는 Lc 6C XX - 오류 길이(XX는 정확한 길이 표시) 6E 00 - 지원되지 않는 CLA

<표 5-20> 에러 코드

5.9 CLEAR

5.9.1 기능 및 특징

○ CLEAR 명령어는 개인키, 인증서 및 공개키 파일의 내용을 초기 상태(‘00’)로 하는 명령이다.

○ 사용자 PIN이 제출된 경우에 실행될 수 있다.

5.9.2 Command message

코 드	내 용
CLA	90
INS	0C
P1	Reference Control - (주) 참조
P2	00
Lc	Empty
Data	Empty
Le	Empty

<표 5-21> CLEAR Command APDU

(주) : Reference Control P1의 코딩

b8	b7	b6	b5	b4	b3	b2	b1	의 미	비 고
0	0	0	-	-	-	-	-		고정된 값
-	-	-	x	x	x	x	x	SFI	

5.9.3 Response message

코 드	내 용
SW1-SW2	Success : 90 00 Error 69 82 - 시큐리티 Status 불충족 69 85 - 사용 조건을 만족하지 못함 6A 82 - 존재하지 않는 파일 6A 86 - 오류 파라미터 (P1, P2) 67 00 - 오류 길이 6E 00 - 지원되지 않는 CLA

<표 5-22> 에러 코드

5.10 GENERATE PUBLIC KEY PAIR

5.10.1 기능 및 특징

○ GENERATE PUBLIC KEY PAIR 명령어는 카드에서 공개키와 개인키를 생성하고 카드 내에 저장하도록 하기 위해 사용한다.
○ 사용자 PIN이 제출된 경우에 실행될 수 있다.
○ PIN 초기화(PIN 미갱신) 상태에서는 수행 불가하다.
○ 발행기관 키가 설정된 경우에는 MUTUAL AUTHENTICATE 명령어를 선행해야 한다.

5.10.2 Command message

코 드	내 용
CLA	90
INS	46
P1	(주1) 참조
P2	(주2) 참조
Lc	00, 04
Data	Lc가 00일 때, Empty Lc가 04일 때, Public exponent(e)
Le	응답에서 예상되는 최대 데이터 길이

<표 5-23> GENERATE PUBLIC KEY PAIR Command APDU

(주1)

P1	내 용
0N	인증서N을 위한 키 생성 (N: 1～최대인증서저장개수)

(주2) P2의 상위 4bits는 RSA Private Key 및 RSA CRT Private Key를 생성하기 위한 플래그, 하위 4bits는 생성될 키의 길이를 의미한다.

P2	내 용	비고
0X	RSA Private Key	
1X	RSA CRT Private Key	
X1	RSA1024용	
X2	RSA2048용	

5.10.3 Response message

코 드	내 용	길 이
Data	공개키 Public Key*	Var.
SW1,SW2	COMPLETION CODE	2

*4.5 공개키 파일구조를 따른다.

<표 5-24> GENERATE PUBLIC KEY PAIR Response APDU

코 드	내 용
SW1-SW2	Success : 90 00 Error 67 00 - 오류 길이 69 82 - 시큐리티 Status 불충족 6A 86 - 오류 파라미터 (P1, P2) 6A 80 - Data 오류 6E 00 - 지원되지 않는 CLA

<표 5-25> 에러 코드

5.11 INITIALIZE CRYPTO

5.11.1 기능 및 특징

○ INITIALIZE CRYPTO 명령어는 PERFORM CRYPTO, STORE PRIVATE KEY 명령어를 수행하기 위한 파라미터들을 설정한다.
○ INITIALIZE CRYPTO 명령어 수행 후 1회의 PERFORM CRYPTO 또는 STORE PRIVATE KEY 명령어 수행이 가능하다.
○ 사용자 PIN이 제출된 경우에 실행될 수 있다.
○ PIN 초기화(PIN 미갱신) 상태에서는 수행 불가하다.

5.11.2 Command message

코 드	내 용
CLA	90
INS	2A
P1	(주1) 참조
P2	(주2) 참조
Lc	00, IV의 길이(주3)
Data	Empty, IV
Le	Empty

<표 6-26> INITIALIZE CRYPTO Command APDU

(주1) P1은 암호 알고리즘, 암호 연산의 종류를 표시한다.

b8	b7	b6	b5	b4	b3	b2	b1	의 미	비 고
x	x	x	x	-	-	-	-	알고리즘	0000: RSA 0001: 3DES 0010: SEED 0011: ARIA(Optional) 0100: AES
-	-	-	-	x	-	-	-	암호화/복호화	0: 암호화(서명검증) 1: 복호화(서명생성)
-	-	-	-	-	x	x	x	운영모드	000: ECB 001: CBC 010: CFB(Optional) 011: OFB(Optional) 100: CTR(Optional)

* 지원하지 않는 운영모드 또는 알고리즘의 경우 Error처리

(주2) P2는 암호 연산의 대상을 표시한다.

b8	b7	b6	b5	b4	b3	b2	b1	의 미	비 고
x	-	-	-	-	-	-	-	토큰에 임시 저장된 키 사용 여부****	0: 파일에 저장된 키 사용 1: 임시 저장된 키 사용
-	x	x	x	-	-	-	-	N번째 암/복호화용 키	3비트로 1~7 표시***
-	-	-	-	x	-	-	-	복호화된 대칭키를 토큰에 임시저장 여부**	0: 토큰에 저장 안함 1: 임시저장
-	-	-	-	-	x	x	x	복호화 결과 저장**	0: 토큰에 저장 안함 1~7: 저장 위치*

* Perform Crypto를 수행할 경우 8.7 비밀키 파일구조에 따라, Store Private Key를 수행 할 경우 8.3 개인키 파일구조에 따라 저장한다.

** 공개키 기반 복호화된 키를 토큰에 임시저장하거나 토큰 내 파일에 저장하는 경우 PKCS#1 Encryption v1.5을 사용하며, 키 저장 시 패딩으로 사용된 문자열은 제외하도록 한다. 이 외 일반적인 복호화 결과로 반환되는 값의 패딩 여부는 고려하지 않는다.

*** b8이 0인 경우, 0의 값은 RFU

b8이 1인 경우, 0의 값을 가진다. 단, 1~7의 값은 별도 목적으로 사용한다.

**** P1의 암호 알고리즘이 RSA이면, 1의 값은 RFU

(주3) IV의 길이가 00이고 암/복호화 모드가 ECB가 아닐 경우, IV값으로 지정된 암호화 알고리즘의 Data Block Size 만큼의 0x00을 사용하여야 한다.

5.11.3 Response message

코 드	내 용	길 이
SW1,SW2	COMPLETION CODE	2

<표 5-27> INITIALIZE CRYPTO Response APDU

코 드	내 용
SW1-SW2	Success : 90 00 Error 69 82 - 시큐리티 Status 불충족 69 85 - 사용 조건을 만족하지 못함 6A 81 - 지원하지 않는 기능 6A 86 - 오류 파라미터 (P1, P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6E 00 - 지원되지 않는 CLA

<표 6-28> 에러 코드

5.12 PERFORM CRYPTO

5.12.1 기능 및 특징

○ PERFORM CRYPTO 명령어는 암호 연산을 수행한다.

○ 이 명령어 수행 직전에 INITIALIZE CRYPTO 명령어를 수행하여야 한다.

5.12.2 Command message

코 드	내 용
CLA	90
INS	2E
P1	(주1) 참조
P2	(주1) 참조
Lc	입력 데이터 길이 - (주2) 참조
Data	입력 데이터 - (주2) 참조
Le	출력 데이터 길이

<표 5-29> PERFORM CRYPTO Command APDU

(주1)

P1	P2	Lc	Data	비 고
00	00	xx	입력 데이터	
01	입력 데이터의 첫번째 바이트	FF	입력 데이터의 나머지 바이트들	ㅇ2048-bit RSA 연산에서 사용함 ㅇ대칭키 알고리즘에서 256바이트 데이터 처리시 사용함

(주2) 비대칭키 암호 연산은 1024-bit(128-byte), 2048-bit(256-byte) 데이터에 대해서 암호 연산을 수행할 수 있으며, 대칭키 암호 연산은 256-byte 이하의 블록 단위로 이루어진 데이터에 대해서 암호 연산을 수행할 수 있다. 카드는 별도의 패딩을 적용하지 않는다.

5.12.3 Response message

코 드	내 용	길 이
DATA	출력 데이터	Var
SW1,SW2	COMPLETION CODE	2

<표 5-30> PERFORM CRYPTO Response APDU

코 드	내 용
SW1-SW2	Success : 90 00 61 XX - More Data bytes(XX) Available Error 69 85 - 사용 조건을 만족하지 못함 6A 86 - 오류 파라미터 (P1, P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6E 00 - 지원되지 않는 CLA

<표 5-31> 에러 코드

5.13 STORE PRIVATE KEY

5.13.1 기능 및 특징

○ INITIALIZE CRYPTO 명령에 의한 선행 작업이 이루어져야 한다.
○ 비대칭키 알고리즘의 개인키를 토큰 내부에 저장하는 역할을 한다.
○ 키쌍 복원용 대칭키가 사전에 카드에 저장되어 있어야 한다.
○ 금융IC카드 패딩규칙에 따라 패딩한다.

5.13.2 Command message

코 드	내 용
CLA	90
INS	3C
P1	파일의 오프셋 상위바이트
P2	파일의 오프셋 하위바이트
Lc	입력데이터의 길이
Data	암호화된 데이터 - (주1)
Le	Empty

<표 5-32> STORE PRIVATE KEY Command APDU

(주1) Initialize Crypto에 의해서 지정된 대칭키 알고리즘의 데이터 블록 길이의 배수 이어야 하며, 데이터를 각각 암호화하여 보낸다. 카드 내에서 복호화된 개인키 데이터는 각 파일의 구조와 일치하도록 저장하여야 한다.

(주2) 외부 인증서 Import 방법(예시)

<순서1> 대칭키 전달용 RSA 키쌍 생성 요청
<순서2> 카드로부터 생성된 키쌍의 공개키 읽기
<순서3> PKCS11에서 대칭키 생성
<순서4> <순서2>에서 얻은 공개키를 이용하여 대칭키를 암호화
<순서5> 암호화된 대칭키 카드로 전달
(P2에 RSA키파일 ID 및 저장될 위치 지정)
* 카드 내에 공유된 대칭키가 존재하는 경우 순서1~5 생략 가능
<순서6> 전달할 인증서의 비대칭키를 대칭키로 암호화하여 카드로 저장

5.13.3 Response message

코 드	내 용	길 이
SW1,SW2	COMPLETION CODE	2

<표 5-33> STORE PRIVATE KEY Response APDU

코 드	내 용
SW1-SW2	Success : 90 00 Error 69 85 - 사용 조건을 만족하지 못함 6A 86 - 오류 파라미터 (P1, P2) 6A 87 - P1, P2와 일치되지 않는 Lc 6E 00 - 지원되지 않는 CLA

<표 5-34> 에러 코드

5.14 MUTUAL AUTHENTICATE

5.14.1 기능 및 특징

○ 발행기관 키가 설정된 경우에만 사용 가능하다.
○ GET CHALLENGE 명령에 의한 선행 작업이 이루어져야 한다.
○ 카드에 저장된 발행기관 키를 이용하여 발행기관을 검증하고, 발행기관이 보안토큰을 검증하기 위한 인증 데이터를 생성하는 명령어이다.

5.14.2 Command message

코 드	내 용
CLA	00
INS	82
P1	00 (주1)
P2	00
Lc	20
Data	암호화된 데이터 - (주2)
Le	04

<표 5-35> MUTUAL_AUTH Command APDU

(주1) FCI에 정의된 기본암호화알고리즘을 활용한다.

(주2) R_{ICC} : 카드 내에서 생성한 난수(16바이트)
R_S : 발행기관 서버가 생성한 난수(16바이트)
Key : 발행기관 키
SKey : 암호화된 데이터를 생성할 키

○ 암호화된 데이터 생성 방법 <Server(HOST)>

$SKey = CBC(R_{ICC}, Key)$
$Ciphered\ Data = CBC(R_S \| R_{ICC}, SKey)$

5.14.3 Response message

코 드	내 용	길 이
DATA	생성된 인증데이터 - (주)	4
SW1,SW2	COMPLETION CODE	2

<표 5-36> MUTUAL_AUTH Response APDU

(주) R_{ICC} : 카드 내에서 생성한 난수(16바이트)
R_S : 발행기관 서버가 생성한 난수(16바이트)
Key : 발행기관 키
SKey : 암호화된 데이터를 생성할 키

○ 인증데이터 생성 방법 <IC Card>

$SKey = CBC(R_{ICC}, KEY)$
$MAC = CBC(R_{ICC} \| R_S, SKey)$

코 드	내 용
SW1-SW2	Success : 90 00 Error 67 00 - 오류 길이 69 82 - 시큐리티 Status 불충족 69 85 - 사용조건을 만족하지 못함 6A 81 - 지원되지 않는 기능 6A 82 - 지정된 Key/PIN이 존재하지 않음 6A 86 - 정확한 파라미터가 아님

<표 5-37> 에러 코드

6. 보안토큰 인스톨 - 공인인증서 기반거래용

□ 자바카드

1. Package AID : D410650990008000
2. Applet(Module) AID : D4106509900080
3. Install Parameter

TAG	LENGTH	VALUE	설 명
C9	09	AA BBBBBB CC DD EE FF GG (HEX)	AA : PIN 재시도 가능 횟수(최대 5회) BBBBBB : 발행기관식별자 CC : 인증서 최대 저장 개수(최대 7개) DD : PIN 최소 길이(6이상) EE : PIN 최대 길이 FF : 사용자인증방법 GG : 기본암호화알고리즘

* PIN 재시도 가능횟수는 최대 5회로 한다.

** 발행기관식별자는 BCD로 표기된 '산업분류코드(금융:065)+발행기관코드'로 인스톨 파라미터를 통해 설정

※ 금융결제원 발급 애플릿(금융결제원 발행기관코드: 099)의 경우 '06 50 99' 값을 가짐

*** 인스톨 후에 초기 PIN값은 Min길이만큼 0x30으로 설정한다.

**** 사용자인증방법

Bit 정의	내 용
x x x x - - - -	사용자인증 알고리즘 정보(미지원시 해당 bit값은 0)
- - - - - - - 0	보안토큰 단말기로부터 사용자 인증
- - - - - - - 1	구동프로그램에서 입력받은 PIN을 이용한 사용자인증
기타	RFU

*****기본암호화알고리즘은 3DES(0x10), SEED(0x20), AES-128(0x40), 미사용(0x00) 중에서 선택하여 설정한다.

□ 멀토스카드

1. 이 명령어는 1회만 수행되어야 함
2. Command message

코드	내 용
CLA	90
INS	30
P1	05 : 보안토큰 규격 거래를 위한 Installation
P2	00
Lc	P1이 05인 경우 : 0x09
Data	(주1) 참조
Le	없음

(주1) : DATA

P1	의 미	길 이
05	PIN 재시도 가능 횟수	01
	발행기관식별자	03
	인증서 최대 저장 개수(최대 7개)	01
	PIN 최소 길이(6이상)	01
	PIN 최대 길이	01
	사용자인증방법	01
	기본암호화알고리즘	01

* PIN 재시도 가능횟수는 최대 5회로 한다.

** 발행기관식별자는 BCD로 표기된 '산업분류코드(금융:065)+발행기관코드'로 인스톨 파라미터를 통해 설정

※ 금융결제원 발급 애플릿(금융결제원 발행기관코드: 099)의 경우 '06 50 99' 값을 가짐

*** 인스톨 후에 초기 PIN값은 Min길이만큼 0x30으로 설정한다.

**** 사용자인증방법

Bit 정의	내 용
x x x x - - - -	사용자인증 알고리즘 정보(미지원시 해당 bit값은 0)
- - - - - - - 0	보안토큰 단말기로부터 사용자 인증
- - - - - - - 1	구동프로그램에서 입력받은 PIN을 이용한 사용자인증
기타	RFU

*****기본암호화알고리즘은 3DES(0x10), SEED(0x20), AES-128(0x40), 미사용(0x00) 중에서 선택하여 설정한다.

3. Response Message

코 드	내용
SW1-SW2	Success : 9000 Error 6700 - 오류길이 6A87 - 허가되지 않은 명령어 9501 - 이미 명령어가 수행됨 9502 - 데이터가 허가 범위를 초과함

부록 2. 규격 연혁

버전	제·개정일	제·개정내역
v1.00	2007년 4월	o “보안토큰 기반의 공인인증서 저장 기술규격”으로 제정
v1.10	2008년 10월	o 보안토큰 기반의 공인인증서 등 객체에 대한 저장위치, 저장형식을 금융 IC 카드 표준을 준용하도록 개정
v1.11	2009년 9월	o 공인전자서명인증체계 기술규격 개정에 따라 본문 내용 중 관련 기술규격 참조 변경 사항 개정
v1.2	2015년 2월	o 기술규격 참조 변경사항 반영
v1.3	2016년 1월	o 무선통신단말을 지원하는 보안토큰 기반 공인인증서 저장형식 규격 반영