공인인증서 요청형식 프로토콜 규격

Accredited Certificate Request Message Format Specification

v1.21

2009년 9월

KISA 한국인터넷진흥원
Korea Internet & Security Agency

목 차

공인인증서 요청형식 프로토콜 규격

Accredited Certificate Request Message Format Specification

1. 개 요

본 규격에서는 전자서명법에 따라 구축된 공인전자서명인증체계 공인인증서비스에서 사용되는 유·무선 공인인증서 요청형식을 규정한다.

2. 규격의 구성 및 범위

본 규격은 [RFC2511], [WAPWPKI] 등 국제표준을 준수하여, 국내 공인전자서명인증체계 내에서 사용되는 유·무선 공인인증서 요청 메시지 형식을 정의한다.

첫 번째로 제6장에서는 유선 공인인증서비스에서의 온·오프라인 인증서 요청형식을 정의한다.

두 번째로 제7장에서는 무선 공인인증서비스에서의 온·오프라인 인증서 요청형식을 정의한다.

3. 관련 표준 및 규격

3.1 국외 표준 및 규격

[RFC2119]	IETF RFC 2119 (1997), *Key Words for use in RFCs to Indicate Requirement Levels*
[RFC2510]	IETF RFC 2510 (1999), *Internet X.509 Public Key Infrastructure Certificate Management Protocol*
[RFC2511]	IETF RFC 2511 (1999), *Internet X.509 Certificate Request Message Format*
[WAPWPKI]	WAP Forum Approved Version 24-April-2001, WAP-217-WPKI, : *Wireless Application Protocol Public Key Infrastructure Definition*
[WAPWTLS]	WAP Forum Approved Version 6-April-2001, *Wireless*

Transport Layer Security

[PKCS10] RSA Laboratories PKCS#10 v1.7 (2000), *Certification Request Syntax Standard*

3.2 국내 표준 및 규격

[KCAC.TS.RS] KISA, KCAC.TS.RS, v1.11, *공인인증서 발급을 위한 참조번호/인가코드 기술규격,* 2009

[KCAC.TS.SIVID] KISA, KCAC.TS.SIVID, v1.21, *식별번호를 이용한 본인확인 기술규격,* 2009

[KCAC.TS.E2E] KISA, KCAC.TS.E2E, v1.30, *무선 응용계층 보안 프로토콜 기술규격,* 2003

[KCAC.TS.DSIG] KISA, KCAC.TS.DSIG, v1.30, *전자서명 알고리즘 규격*, 2009

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
나) 공인인증기관
다) 가입자

4.2 용어의 정의

가) 유선 공인인증서비스 : 인터넷 기반의 전자거래를 위해 공인인증서를 이용하는 서비스

나) 무선 공인인증서비스 : 무선 단말기 기반의 전자거래를 위해 공인인증서를 이용하는 서비스

다) 개인키 소유여부 검증정보 : 인증서 요청정보에 포함된 공개키가 대응되는 개인키를 가입자가 소유하고 있음을 증명하기 위한 정보

4.3 용어의 효력

본 규격에서 사용된 다음의 용어들은 전자서명인증체계 공인인증서 요청형식 프로토콜의 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

본 규격에서는 다음의 약어가 이용된다.

가) CA : Certification Authority, 인증기관

나) RA : Registration Authority, 등록대행기관

6. 유선 공인인증서비스에서의 공인인증서 요청형식

6.1 온라인 요청형식

온라인 요청형식은 인증서 요청정보와 개인키 소유여부 검증정보 및 추가 등록 정보로 구성되며, 온라인 요청형식의 구성은 다음과 같다.

```
CertReqMessage ::= SEQUENCE SIZE (1..MAX) OF CertReqMsg

CertReqMsg ::= SEQUENCE {
        certReq         CertRequest,
        pop             ProofOfPossession     OPTIONAL,
        regInfo         SEQUENCE SIZE (1..MAX) of AttributeTypeAndValue
                        OPTIONAL}
```

certReq는 인증서 요청정보로 인증서에 포함될 내용만을 포함해야 한다.

pop는 가입자의 개인키 소유여부 검증정보로 가입자의 키가 전자서명키인 경우, 인증서 요청정보에 대한 서명값을 포함해야 한다.

regInfo는 인증 요청을 위해 추가적인 등록정보가 요구될 때 사용되는 정보로, [KCAC.TS.SIVID]에 따른 가입자 신원확인 정보 및 가입자의 접속 정보, 지불 정보 등이 포함될 수 있다.

6.1.1 인증서 요청정보

인증서 요청정보의 구성은 다음과 같다.

```
CertRequest ::= SEQUENCE {
        certReqId       INTEGER,
        certTemplate    CertTemplate,
        controls        Controls OPTIONAL }
```

certReqID는 인증 요청 메시지와 이에 대응하는 응답 메시지를 확인하기 위해 사용되는 정보로, 대응되는 메시지간에 certReqID 값은 서로 동일해야 한다.

certTemplate는 [RFC2510]에 따라 인증서 발급에 대한 가입자의 요청에 따른 각각의 정보를 포함하며, 다음과 같이 구성된다.

```
CertTemplate ::= SEQUENCE {
        version         [0] Version                 OPTIONAL,
        serialNumber    [1] INTEGER                 OPTIONAL,
        signingAlg      [2] AlgorithmIdentifier     OPTIONAL,
```

```
        issuer          [3] Name                    OPTIONAL,
        validity        [4] OptionalValidity        OPTIONAL,
        subject         [5] Name                    OPTIONAL,
        publicKey       [6] SubjectPublicKeyInfo    OPTIONAL,
        issuerUID       [7] UniqueIdentifier        OPTIONAL,
        subjectUID      [8] UniqueIdentifier        OPTIONAL,
        extensions      [9] Extensions              OPTIONAL }

    OptionalValidity ::= SEQUENCE {
        notBefore   [0] Time OPTIONAL,
        notAfter    [1] Time OPTIONAL }
```

control은 인증서 발급에 영향을 주는 정보를 포함하는 것으로, regToken, authenticator, pkiPublicationInfo, oldCertID의 control이 사용될 수 있다.

```
Contorls ::= SEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue
```

regToken control은 가입자의 신원을 확인하기 위한 일회성 정보를 가지며, 이 정보는 CA에서 생성할 수도 있고 out-of-band로 가입자에게 제공될 수 있다. 이 control은 PKI 서비스에 신규 가입한 가입자에 대해서만 사용가능하며, UTF8으로 인코딩된다.

authenticator control은 가입자의 신원을 확인하기 위해 CA와 가입자간에 공유된 가입자 정보의 일부가 포함될 수 있다. 이 control은 신규 가입자뿐만 아니라 기존 가입자들의 인증서 요청 시에도 사용될 수 있다.

pkiPublicationInfo control은 인증서 공고 방법 및 공고 위치 정보를 포함한 것으로, control의 형식은 [RFC2511]을 준용한다.

oldCertID control은 가입자의 기존 인증서가 가입자의 인증 요청 메시지에 포함된 정보로 갱신될 수 있도록 기존 인증서의 발급자 DN과 인증서 일련번호를 포함해야 한다. control의 형식은 [RFC2511]을 준용한다.

6.1.2 개인키 소유여부 검증정보

가입자의 정당한 개인키 소유여부를 검증하기 위해 온라인 요청형식은 개인키 소유여부 검증정보를 포함해야 하며, 이 정보는 CA 혹은 RA에 의해 검증되

어야 한다. 인증기관의 정책에 따라 개인키 소유여부 정보를 out-of-band로 검증하는 경우, 이 필드는 사용되지 않는다.

개인키 소유여부 검증정보의 구성은 다음과 같다.

```
ProofOfPossession ::= CHOICE {
        raVerified          [0] NULL,
        signature           [1] POPOSigningKey,
        keyEncipherment     [2] POPOPrivKey,
        keyAgreement        [3] POPOPrivKey }
```

가입자의 전자서명생성키에 대한 소유여부 검증을 위한 정보는 signature에 포함되어야 하며, 다음과 같이 구성된다.

```
POPOSigningKey ::= SEQUENCE {
        poposkInput         [0] POPOSigningKeyInput OPTIONAL,
        algorithmIdentifier     AlgorithmIdentifier,
        signature               BIT STRING }
```

인증서 요청 메시지의 CertTemplate에 subject와 publicKey가 포함되는 경우, poposkInput 필드는 생략될 수 있으며, signature는 인증서 요청정보의 DER 인코딩 값에 대한 서명값을 가진다. 이에 반해, 인증서 요청 메시지의 CertTemplate에 subject와 publicKey가 없는 경우, poposkInput 필드는 반드시 존재해야 하며, 이 필드에 대해 DER 인코딩된 값을 signature의 값으로 갖는다. poposkInput 필드는 다음과 같이 구성된다.

```
POPOSigningKeyInput ::= SEQUENCE {
        authInfo        CHOICE {
                sender [0] GeneralName,
                publicKey       PKMACValue},
        publicKey       SubjectPublicKeyInfo }
```

sender 필드는 인증서 요청자가 사전에 인증된 가입자인 경우에만 사용되며, 신규 가입자 등 사전에 인증되지 않은 가입자의 경우, publicKeyMAC 필드가 사용된다. publicKeyMAC 필드는 가입자의 공개키의 DER 인코딩 값에 대해 password-based MAC을 설정한다.

6.2 오프라인 요청형식

오프라인 요청형식은 [PKCS10]을 준용하여야 한다.

7. 무선 공인인증서비스에서의 공인인증서 요청형식

7.1 무선 공인인증서 관리형식 타입 스트링 및 인코딩 규칙

본 규격에서 요청(발급)에 따른 해당 타입을 아래와 같이 정의하고, 3 바이트의 스트링(String)으로 표현한다.

종 류	키분배 및 전자서명	전자서명	키분배
요청(발급)	100	110	120

모든 바이너리 데이터는 base64 인코딩 규칙을 따라야 한다. 구분자는 버티칼 라인(|)을 사용하지만, 해쉬 메시지의 concatenation시에 구분자를 사용하지 않아야 한다. 또한, 참조번호로 사용할 수 있는 문자의 범위에서 버티칼 라인(|)은 제외해야 한다. 또한, 정의하고 있는 데이터를 등록기관(또는 공인인증기관)에게 전송할 때는 POST 방식을 사용해야 한다.

7.2 온라인 요청형식

온라인 요청형식은 공인인증서 생성을 위해 가입자가 공인인증기관(또는 등록기관을 통해)에게 공인인증서 요청을 전달하는 메시지이다.

가입자는 일회성 정보(ID, 패스워드), POP를 위한 방법, 가입자의 공개키(전자서명용 공인인증서인 경우에는 신원확인정보)를 포함하는 요청형식을 생성하여 Replay attack, 메시지 위·변조 방지할 수 있는 요청형식을 구성하여 공인인증기관(또는 등록기관을 통해)에 공인인증서 요청형식을 전달하여야 한다. 일회성 정보(ID, Passwd)는 [KCAC.TS.RS]를 준수하여 이용하고, 가입자가 공인인증서를 요청할 때에는 [KCAC.TS.E2E]에서 정의한 signText함수를 사용하여 인증서 요청형식을 생성한다.

o 전자서명용 또는 키분배용으로 요청하는 경우

가입자		등록기관(또는 공인인증기관)
		nonce 생성
	nonce →	
R 생성		
VID = H(H(IDN,R))		
EVID = E(VID|R)		
M = type| *PK*| ID|EVID		
N = Passwd		
SignedContent = signText(M|H(M,N), 1, 0,"")		
CR = SignedContent		
	← CR	
		SignedContent = CR
		PK, ID 추출 및 SignedContent 검증
		데이터베이스에서 Passwd 및 IDN′추출
		H′= H(M,Passwd)
		H′?= H(M, N)
		VID|R = D(EVID)
		VID?= H(H(IDN′, R))

※ 키분배용 인증서를 발급 받는 경우에는 EVID를 생략함
※ 여기서 nonce는 사용하지 않음

o 전자서명 및 키분배를 동시에 요청하는 경우

가입자		등록기관(또는 공인인증기관)
		nonce 생성
	nonce →	
R 생성 VID = H(H(IDN,R)) EVID = E(VID|R) SignValue$_{키분배}$는 nonce를 $SK_{키분배}$로 서명한 값 M = type| PK| ID| $PK_{키분배}$ | SignValue$_{키분배}$|nonce|EVID N = Passwd SignedContent = signText(M|H(M,N), 1, 0,"") CR = SignedContent		
	← CR	SignedContent = CR PK, ID 추출 및 SignedContent 검증 데이터베이스에서 Passwd 및 IDN′추출 $PK_{키분배}$로 SignValue$_{키분배}$ 검증 H′= H(M,Passwd) H′?=H(M, N) VID|R = D(EVID) VID?=H(H(IDN′, R))

※ nonce로 서명 생성후 요청하는 데 걸리는 시간은 인증기관의 정책에 따라 결정하고, 키분배용 ECDH키는 ECDSA 서명 생성하는 것으로 POP 확인함

등록기관(또는 공인인증기관)의 응답형식은 [부록 3]에 정의한다.

7.3 오프라인 요청형식

오프라인 요청형식은 [PKCS10]을 준용하여야 한다.

부록 1. VID 및 EVID 구조

1.1 VID 구조

```
enum { SHA1(0), (255) } HashAlgorithm ;

struct {
        HashAlgorithm       hash_alg ;
        opaque              virtualID<0..2^8-1> ;
} VID ;

// virtualID는 아래에 정의된 HashContent를 HashAlgorithm으로 2번 해쉬한
   값이다.


struct {
        opaque              idn<0..2^8-1>;
        opaque              randomNum[20];
} HashContent ;
```

1.2 EVID 구조

```
enum { RSA(0), (255) } EncryptionAlgorithm ;

struct {
        Identifier      issuer;
        opaque          serialNumber<0..2^8-1>;
} IssuerAndSerialNumber ;


struct {
        uint8                       version ;
        HashAlgorithm               vid_hash_alg ;
        EncryptionAlgorithm         vid_encyption_alg ;
```

```
        IssuerAndSerialNumber       certID ;
        opaque                      encryptedVID<0..2^16-1> ;
} EVID ;

// encryptedVID는 아래에 정의된 EncryptionContent를 암호화한 값이다.

struct {
        VID                 vid ;
        opaque              randomNum[20] ;
} EncryptionContent ;
```

부록 2. PK(Public Key)의 구조

```
enum { rsa(2), ecdh(3), ecdsa(4), (255) } PublicKeyType ;

struct {
    select (PublicKeyType) {
        case ecdh : ECPublicKey ;
        case ecdsa : ECPublicKey ;
        case rsa : RSAPublicKey ;
    }
} PublicKey ;

struct {
    opaque rsa_exponent<1..2^16-1> ;
    opaque rsa_modulus<1..2^16-1> ;
} RSAPublicKey ;

enum { ECunNamed(0), ECNamed(1), (255) } ECNameType  ;

struct {
    select (ECNameType) {
        case ECunNamed :
            ECParameters ;
        case ECNamed :
            opaque oid<1..2^8-1> ;
    }
    opaque point<1..2^8-1>;
} ECPublicKey ;
```

※ ECNamed에 포함되는 타원곡선암호화(ECC) 커브는 [KCAC.TS.DSIG]를 따름

```
enum { ec_prime_p(1), ec_characteristic_two(2), (255) } ECFieldID;
```

```
enum { ec_basis_onb(1), ec_basis_trinomial(2), ec_basis_pentanomial(3),
ec_basis_polynomial(4) } ECBasisType;

struct {
        opaque a <1..2^8-1>;
        opaque b <1..2^8-1>;
        opaque seed <0..2^8-1>;
} ECCurve;

struct {
        ECFieldID field;
        select (field) {
        case ec_prime_p: opaque prime_p <1..2^8-1>;
        case ec_characteristic_two:
                uint16 m;
                ECBasisType basis;
                select (basis) {
                        case ec_basis_onb:
                                struct { };
                        case ec_trinomial:
                                uint16 k;
                        case ec_pentanomial:
                                uint16 k1;
                                uint16 k2;
                                uint16 k3;
                        case ec_basis_polynomial:
                                opaque irreducible <1..2^8-1>;
                }
        }
ECCurve curve;
ECPoint base;
opaque order <1..2^8-1>;
opaque cofactor <1..2^8-1>;
} ECParameters;
```

※ ECParameters 구조에 대한 설명은 [WAPWTLS]를 참조한다.

부록 3. 인증서 요청에 대한 응답형식

3.1 성공(Success)

○ MIME Type : application/vnd.wap.cert-response
○ Content : Base64 인코딩된 CertResponse

```
enum { cert_info(0), cert(1), referral(2), (255) } CertRespType;

struct {
        CharacterSet            character_set;
        opaque                  displayName             <1 .. 2^8 - 1>;
} CertDisplayName;

struct {
        opaque                  url     <0 .. 128>;
} UrlPoint;

struct {
        unit8                   version;
        CertRespType            type;
        select (type) {
                case cert_info:
                        CertDisplayName     display_name[2];
                        Identifier          ca_domain[2];
                        UrlPoint            url[2];
                case cert:
                        CertDisplayName     display_name[2];
                        Identifier          ca_domain[2];
                        X509Certificate     cert[2];
                case referral:
                        UrlPoint            url[2];
                        unit32              seconds_to_wait[2];
        }
} CertResponse;
```

※ 각 배열의 첫 번째는 전자서명용, 두 번째는 키분배용 관련 정보이다.
※ CertResponse 구조에 대한 설명은 [WAPWPKI]를 참조한다.

3.2 실패(Fail)

○ MIME Type : text/plain
○ Content : ascii text 값의 에러 메시지

부록 4. 규격 연혁

버전	제・개정일	제・개정내역
v1.00	2004년 11월	・“공인인증서 요청형식 프로토콜 규격”으로 제정
v1.10	2007년 4월	・“무선 인증서 요청형식 프로토콜 규격 v1.32” 흡수 통합
v1.20	2008년 10월	・관련 국내 표준 및 규격 갱신 내용 반영 ・법률 공포번호가 해당 법률 개정시마다 변경되는 점을 고려하여 법령명으로 개정
v1.21	2009년 9월	・관련 국내 표준 참조 오류 정정