전자문서 증명서 포맷 및 운용절차 기술규격

The Technical/Administrative Standard for the Certificate Format of an Electronic Data Message and its Practical Procedures

v3.20

2026년 1월

KISA 한국인터넷진흥원
KOREA INTERNET & SECURITY AGENCY

목 차

부 록

1. 규격의 개요

1.1 목적

“전자문서 증명서 포맷 및 운용절차 기술규격”(이하 본 규격)은 표준화된 증명요청서 및 증명서의 프로파일을 정의하여 공인전자문서센터가 증명서를 발급함에 있어서 투명하고 효율적인 발급 서비스를 제공하고, 증명서의 호환성 확보로 인한 증명 서비스의 확대 및 서비스의 신뢰성을 제공 할 수 있도록 하는 것을 목적으로 한다.

또한, 증명서의 표준화된 검증 방법을 정의하여 증명서의 올바른 활용에 도움이 되도록 하는데 목적을 갖는다.

1.2 적용 대상 및 범위

본 문서에서는 증명서 요청 메시지와 증명서 응답 메시지 대하여 기술하고 있다.

증명서 요청 메시지는 증명서를 발급받기 위하여 증명서 요청자가 생성하는 증명요청서를 의미하고, 증명서 응답 메시지는 증명요청서에 대한 처리 결과로서 공인전자문서센터가 생성해 내는 메시지를 의미하는데 이는 증명서 또는 에러 메시지로 구성된다.

증명요청서 및 증명서를 구성하고 있는 각각의 필드에 대하여 요소별 의미와 사용 방법을 기술하고, 각각의 구조를 ASN.1 등 여러 구문을 이용하여 정의한다.

1.3 참고 자료

- □ X500; ITU-T Recommendation X.500 | ISO/IEC 9594-1:1998, Information technology - Open Systems Interconnection - The Directory : Overview Of Concepts, Models and Services
- □ X501; ITU-T Recommendation X.501 | ISO/IEC 9594-2:1995, Information technology - Open Systems Interconnection - The Directory : Part 2 : Models
- □ X509; ITU-T Recommendation X.509 | ISO/IEC 9594-8:1998, Information technology - Open Systems Interconnection - The Directory : Authentication Framework

☐ X680; ITU-T Recommendation X.680 | ISO/IEC 8824-1: Information Technology – Abstract Syntax Notation One (ASN.1) : Specification of basic notation

☐ X690; ITU-T Recommendation X.690 | ISO/IEC 8825-1: Information Technology – ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

☐ X697; ITU-T Recommendation X.697(02/21) | ISO/IEC 8825-8: Information Technology – ASN.1 Encoding Rules: Specification of Javascript Object Notation Encoding Rules(JER)

☐ X693; ITU-T Recommendation X.693(02/21) | ISO/IEC 8825-4: Information Technology – ASN.1 Encoding Rules: Specification of XML Encoding Rules(XER)

☐ CMP; Internet X.509 Public Key Infrastructure Certificate Management Protocol(CMP)(RFC 4210), IETF, 2005

☐ CMS; Cryptographic Message Syntax(RFC 3852), IETF, 2004

☐ CMS; Cryptographic Message Syntax(RFC 5035), IETF, 2007

☐ PAdES; ISO 32000-2:2020, ISO, 2020

☐ CAdES, PAdES; ETSI EN 319 122, ETSI, 2021

☐ Hash, RSA; 암호 알고리즘 및 키 길이 이용 안내서, KISA, 2018

☐ KCAC.TS.DN; 전자서명인증관리체계 DN 규격 v1.21, 한국인터넷진흥원, 2009

☐ KCAC.TS.CERTPROF; 전자서명 인증서 프로파일 규격 v1.70, 한국인터넷진흥원, 2009

☐ KCAC.TS.SIVID; 식별번호를 이용한 본인확인 기술규격 v1.21, 한국인터넷진흥원, 2009

☐ KISA-TS-PACKAGE; 전자문서 정보패키지 기술규격 v3.10, 한국인터넷진흥원, 2023

☐ KISA-TS-INTERFACE; 공인전자문서센터 연계 인터페이스 기술규격 v3.20, 한국인터넷진흥원, 2026

☐ KISA-TS-TRANSFER; 공인전자문서센터 이·수관 기술규격 v3.10, 한국인터넷진흥원, 2023

1.4 규격 어휘

본 규격에서 제시하고 있는 규칙 적용과 관련하여 다음과 같은 유형의 문장 어구를 사용하고 있다. 한글만으로 표현이 충분하지 않은 경우에는 영문을 병기하였다.

□ 필수 요소 : 이 규격에서 제시하는 규칙을 절대적으로 따라야 할 때 사용한다. 규격에 부합하기 위해서는 이것을 엄밀하게 따라야 하며, 이것을 벗어나는 것을 인정하지 않는다. (영문 : Must, Must Not)

- ~ 한다.
- ~ 하여야 한다.
- ~ 안된다.
- ~ 않는다.

□ 권고(선택) 요소 : 이 규격에서 제시하는 규칙을 따르는 것을 권고할 때 사용한다. 이는 이 밖의 것도 좋지만 이것이 특히 적당하다는 것을 나타낼 때 사용한다. (영문 : Should)

- ~ 하도록 한다.

□ 완곡한 금지 요소 : 규격의 입장에서 바람직하지 않지만, 반드시 금지하지 않는다. (영문 : Should Not)

- ~ 하지 않도록 한다.

□ 허용 요소 : 규격의 입장에서 허락한다는 것을 나타낸다. (영문 : May)

- ~ 할 수 있다.

1.5 버전 호환성

본 버전의 규격을 준수하여 구현된 시스템은 하위 버전의 규격을 준수하여 생성된 증명서에 대한 검증 및 처리가 가능해야 한다. 이때 검증의 기준은 하위 버전의 규격이며, 단 규격상의 오류 내용은 제외한다. 기존 증명서를 갱신하는 등의 작업 과정 중에 새로운 증명서가 생성된다면 해당 증명서는 본 버전의 규격을 준수하여 생성되어야 하며, 생성을 위해 필요한 정보가 기존 증명서에 부족한 경우는 증명서 생성 시스템에서 적절히 생성하도록 한다.

2. 용어 정의

2.1 용어

1) “증명서 발급자”(issuer)란 증명서비스를 수행하는 주체로서 증명서를 발급하는 자를 말한다.

2) “증명서 요청자”(requester)란 증명서를 요청하는 주체로서 증명요청서를 작성하고 증명서를 발급받는 자를 말한다.

3) “증명서 수임자“(nominee)란 증명서를 받아 이용하는 주체로서 증명서를 검증하거나 위임된 권한을 사용하는 자를 말한다.

4) “증명서 검증자“(verifier)란 증명서를 검증하는 주체를 말하며, 증명서 요청자 또는 증명서 수임자는 증명서 검증자가 될 수 있다.

5) “전자문서“란 정보처리시스템에 의하여 전자적 형태로 작성 및 송·수신되어 공인전자문서센터에 저장되는 정보를 말한다.

6) “증적”이란 공인전자문서센터가 전자문서에 대한 등록, 발급, 이관, 폐기 작업을 완료한 후 작업 내역을 구조체의 형식으로 생성한 데이터를 말하며, 증명서 발급 시 증명대상으로서 증명서에 포함된다.

7) “등록증명서“란 공인전자문서센터가 수행한 전자문서 등록 서비스에 대한 증적을 대상으로 발급한 보증서를 말한다.

8) “최초등록증명서“란 이용자의 전자문서가 처음 공인전자문서센터에 등록된 사실에 대해 보증하는 등록증명서를 말하며, 증명서의 유효기간이 전자문서의 보관기간과 동일하며 전자문서가 이·수관 되어도 최초의 전자문서 등록시점을 보증한다는 점에서 일반 등록증명서와 구분된다.

9) “발급증명서“란 공인전자문서센터가 수행한 전자문서 발급 서비스에 대한 증적을 대상으로 발급한 보증서를 말한다. 해당 증명서의 발급은 공인전자문서센터의 필요에 따라 발급 하도록 한다.

10) “이관증명서“란 공인전자문서센터가 수행한 전자문서 이관 서비스에 대한 증적을 대상으로 발급한 보증서를 말한다.

11) “폐기증명서“란 공인전자문서센터가 수행한 전자문서 폐기 서비스에 대한 증적을 대상으로 발급한 보증서를 말한다.

12) “원본증명서“란 공인전자문서센터가 발급한 전자문서가 공인전자문서센터가 보관 중인 원본 전자문서와 동일함을 증명하는 보증서를 말한다.

13) “불변경증명서“란 이용자에게 열람되는 변환본 전자문서의 내용이 공인전자문서센터가 보관 중인 원본 전자문서의 내용과 동일함을 증명하는 보증서를 말한다. 해당 증명서의 발급은 공인전자문서센터의 필요에 따라 발급 하도록 한다.

14) “시점확인증명서“란 어떤 데이터가 특정 시각에 존재하였음을 증명하는 보증서를 말한다. 해당 증명서의 발급은 공인전자문서센터의 필요에 따라 발급 하도록 한다.

15) ”연계 인터페이스”의 정의는 “공인전자문서센터 연계 인터페이스 기술규격 v3.10”(이하 연계 인터페이스 규격)의 정의를 따른다.

16) ”이관 정보패키지”(이하 TIP)의 정의는 “공인전자문서센터 이·수관 기술규격 v3.10”(이하 이·수관 규격)의 정의를 따른다.

2.2 약어

1. ASN.1 : Abstract Syntax Notation One, 추상적 구문 표기
2. CPS : Certificate Policy Statement, 인증업무준칙
3. DN : Distinguished Name, 식별명칭
4. IDN : Identification Number, 식별 번호
5. VID : Virtual ID, 가상 식별 정보
6. OID : Object Identifier, 객체 식별자
7. BER : Basic Encoding Rules, 기본 부호화 규칙
8. DER : Distinguished Encoding Rules, 식별 부호화 규칙
9. CA : CertificateAuthority, 공인인증기관
10. OCSP : Online Certificate Status Protocol : 실시간 인증서 상태 확인 프로토콜
11. AIP : Archival Information Package, 보존 정보패키지
12. DIP : Dissemination Information Package, 배부 정보패키지
13. TIP : Transfer Information Package, 이관 정보패키지

3. 증명서 발급 모델

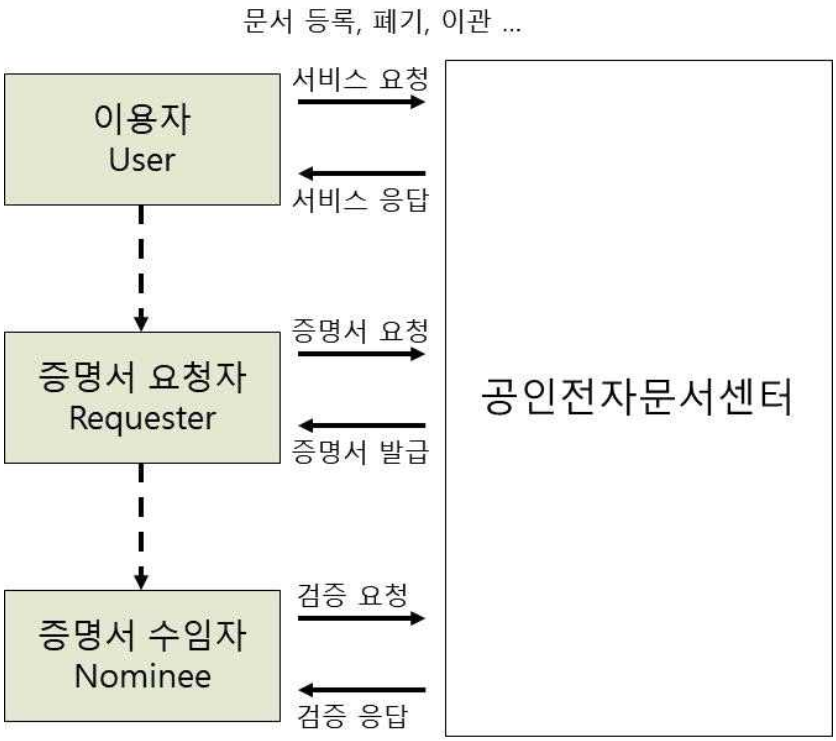
3.1 구성요소

공인전자문서센터의 증명서 발급 과정은 일반적으로 다음과 같은 구성요소를 갖는다.

□ 이용자(user) : 공인전자문서센터의 전자문서보관등 서비스를 이용하는 자를 말하며, 문서의 등록, 폐기, 이관, 발급 등의 일반적인 공인전자문서센터 서비스를 이용 한다. 이용자는 서비스 요청의 종류 및 행위에 따라서 증명서 요청자와 증명서 수임자로 전이가 가능하다.

□ 증명서 요청자(requster) : 증명서를 요청하는 주체로 증명요청서를 공인전자문서센터에 제출하고 증명서를 발급 받는다. 정당한 사유 및 절차에 의하여 증명서 발급 요청 권한이 부여된다면, 이용자 이외의 제 3자도 증명서 요청자가 될 수 있다.

□ 증명서 수임자(nominee) : 증명서를 받아 이용하는 주체로 발급받은 증명서를 검증하거나 위임된 권한을 사용한다.



3.2 서비스 요청 / 수행

공인전자문서센터는 전자문서의 등록, 폐기, 이관, 발급 등의 서비스를 제공한다.

□ 공인전자문서센터는 이용자의 요청에 대한 서비스를 수행하고, 이에 대한 기록을 생성한다.

□ 등록, 발급, 폐기, 이관 서비스에 대한 기록은 증적의 형태로 생성되며, 이를 저장·보관한다.

□ 생성된 기록들은 무결성을 보장하기 위하여 안전하게 보관한다.

□ 수행된 서비스에 대한 결과를 반환한다.

□ 등록 서비스의 경우 추가로 등록 증명서를 발급한다.

3.3 증명서 요청 / 발급

공인전자문서센터는 등록, 발급, 이관, 폐기, 보관, 원본, 불변경, 시점확인 증명서 발급 서비스를 제공한다.

3.3.1 일반적인 증명서 요청 / 발급

증명서 요청자에 의한 증명서 요청에 대하여 공인전자문서센터는 증명서를 발급한다.

□ 공인전자문서센터는 이용자에게 제공한 전자문서 등록, 발급, 이관, 폐기 서비스에 대한 증적을 생성한다.

□ 증명서 요청자는 증명요청서를 생성하여 공인전자문서센터에 전달한다.

□ 공인전자문서센터는 증명요청서의 무결성을 검증한다.

□ 공인전자문서센터는 증명의 대상이 되는 증적을 검색한다.

□ 공인전자문서센터는 증명서 정책, 증적 그리고 증명요청서를 이용하여 증명서를 생성한다.

□ 증명서 요청자에게 증명서를 전달한다.

□ 증명서 생성이 실패하였을 경우 원인을 포함한 에러메시지를 전달한다.

3.3.2 최초등록증명서 발급

전자문서 등록 시에 발급되어 이용자에게 전달되는 최초등록증명서는 증명요청서 없이 공인전자문서센터 시스템에서 자동으로 발급되어진다.

□ 공인전자문서센터는 이용자의 요청에 대한 전자문서 등록 서비스를 수행하고, 이에 대한 증적을 생성하여 저장·보관한다.

□ 공인전자문서센터는 증명서 정책과 증적을 이용하여 최초등록증명서를 생성한다.

□ 공인전자문서센터는 생성된 최초등록증명서를 전자문서 등록 요청에 대한 응답메시지에 포함하거나 증명서 다운로드 서비스와 같은 기타의 방법으로 이용자에게 전달하도록 한다.

□ 증명서 생성이 실패하였을 경우 원인을 포함한 에러메시지를 전달한다.

공인전자문서센터는 이용자가 동의하는 경우에 한하여, 전자문서 등록 후 최초등록증명서를 발급하지 않을 수 있다. 단, 해당 전자문서에 대한 이관 작업이 발생하는 경우, 공인전자문서센터는 반드시 최초등록증명서를 발급하여야 한다.

3.3.3 원본증명서 발급

발급된 전자문서가 원본임을 증명하는 원본증명서는 전자문서 발급 과정 중에 생성되어 전자문서와 함께 이용자에게 전달된다. 이를 위하여 이용자 시스템은 전자문서 발급 요청 시 증명요청서를 생성하여 함께 공인전자문서센터에 전송하여야 한다.

□ 이용자(이 경우는 이용자와 증명서 요청자가 동일함)는 증명요청서를 생성한다.

□ 이용자는 증명요청서와 함께 공인전자문서센터에 원본 전자문서 발급을 요청한다.

□ 공인전자문서센터는 증명요청서의 무결성을 검증한다.

□ 공인전자문서센터는 증명서 정책, 전자문서 발급 정보, 그리고 증명요청서를 이용하여 원본증명서를 생성한다.

□ 공인전자문서센터는 원본증명서를 이용자에게 전달한다.

□ 증명서 생성이 실패하였을 경우 원인을 포함한 에러메시지를 전달한다.

원본증명서와 전자문서를 제출용 등으로 활용하기 위해 한데 묶어 출력하는 경우, 수행 과정은 아래와 같다.

□ 이용자는 발급 받은 증명서의 무결성 검증을 수행한다.

□ 이용자가 원본증명서와 전자문서의 묶음 출력을 요청할 경우 원본증명서와 전자문서가 연접된 형태로 인쇄되도록 하고 원본증명서 첫 페이지부터 전자문서의 마지막 페이지까지 쪽 번호를 함께 인쇄하여야 한다. 인쇄 시 필요한 문서 및 페이지를 선택하여 출력할 수 있다.

□ 묶음 출력 시 전자문서에 원본증명서의 증빙 관련 정보 일부(증명서 일련번호 등)를 함께 인쇄하여야 한다.

□ 공인전자문서센터는 발급에 성공한 원본증명서의 진위확인을 할 수 있는 경로를 제공하여야 한다.

3.3.4 불변경증명서 발급

이용자에게 열람되는 변환본 전자문서의 내용이 원본 전자문서의 내용과 동일함을 증명하는 불변경증명서는 이용자의 요청이 있을 경우 전자문서 열람 과정 중에 생성되어 전자문서와 함께 이용자에게 전달된다. 이를 위하여 이용자 시스템은 전자문서 열람 요청 시 증명요청서를 생성하여 함께 공인전자문서센터에 전송하여야 한다.

□ 이용자(이 경우는 이용자와 증명서 요청자가 동일함)는 증명요청서를 생성한다.

□ 이용자는 증명요청서와 함께 공인전자문서센터에 변환본 전자문서에 대한 열람을 요청한다.

□ 공인전자문서센터는 증명요청서의 무결성을 검증한다.

□ 공인전자문서센터는 증명서 정책, 변환본 전자문서 정보, 그리고 증명요청서를 이용하여 불변경증명서를 생성한다.

□ 공인전자문서센터는 구현된 열람서비스 방식을 통하여 변환본 전자문서와 불변경증명서(또는 각각의 내용)를 이용자에게 전달한다.

□ 증명서 생성이 실패하였을 경우 원인을 포함한 에러메시지를 전달한다.

불변경증명서는 현재 이용자에게 열람되고 있는 변환본 전자문서의 내용이 원본 전자문서의 내용과 동일함을 증명하기 위한 것이기 때문에, 공인전자문서센터가 제공하는 열람 서비스의 다양한 구현방식에 종속되어 불변경 증명을 수행한다.

일반적으로 전자문서 열람은 전자문서의 내용을 확인하기 위한 일회성의 서비스로 제공되는 경우가 많기 때문에 이에 따라 불변경증명서도 해당 열람에 대한 일회성의 불변경 증명을 수행하게 된다. 물론 열람이 종료된 후 공인전자문서센터의 증명서 관리기능을 통하여 해당 불변경증명서를 다시 다운로드 받을 수 있으나, 열람이 종료되었기 때문에 증명할 대상은 없다.

만약 공인전자문서센터가 제공하는 열람 서비스가 변환본 전자문서 파일과 불변경증명서 파일로 저장될 수 있도록 구현되었다면, 이후에도 지속적으로 변환본 전자문서 파일에 대한 열람 및 해당 문서에 대한 불변경 증명을 수행할 수 있다.

참고로, 공인전자문서센터의 열람 서비스 정책 상 원본 전자문서에 대해서도 단순 열람이 가능하도록 기능이 구현된 경우이거나 또는 이용자가 등록한 원본 전자문서의 포맷이 공인전자문서센터가 지원하는 변환포맷이어서 열람이 가능한 경우, 해당 전자문서 파일들은 변환본이 아니라 이용자가 공인전자문서센터에 등록한 원본 전자문서이기 때문에 해당 문서를 열람 시 불변경증명서 발급 대상이 아님에 주의한다.

즉, 불변경증명서의 발급 대상은 열람되는 모든 문서가 아니라, 변환 작업을 통하여 생성된 변환본 문서로 한정된다.

3.3.5 시점확인증명서 요청 / 발급

증명서 요청자에 의한 증명서 요청에 대하여 공인전자문서센터는 증명서를 발급한다.

□ 증명서 요청자는 데이터 해시값을 포함한 증명요청서를 생성하여 공인전자문서센터에 전달한다.

□ 공인전자문서센터는 증명요청서의 형식이 전자서명 구조인 경우 무결성을 검증한다.

□ 공인전자문서센터는 증명요청서와 증명요청서에 기재된 데이터 해시값을 이용하여 증명서를 생성한다.

□ 증명서 요청자에게 증명서를 전달한다.

□ 증명서 생성이 실패하였을 경우 원인을 포함한 에러메시지를 전달한다.

3.4 증명서 검증

검증자는 증명서 검증을 위하여 필요할 경우 공인전자문서센터에 추가적인 데이터를 요청할 수 있으며, 검증자의 권한이 있다고 판단되는 경우 공인전자문서센터는 이에 대한 요청을 받아들여 데이터를 제공할 수 있다.

증명서 검증에 필요한 데이터를 가지고 검증자는 6장에서 기술한 대로 증명서 검증을 수행한다.

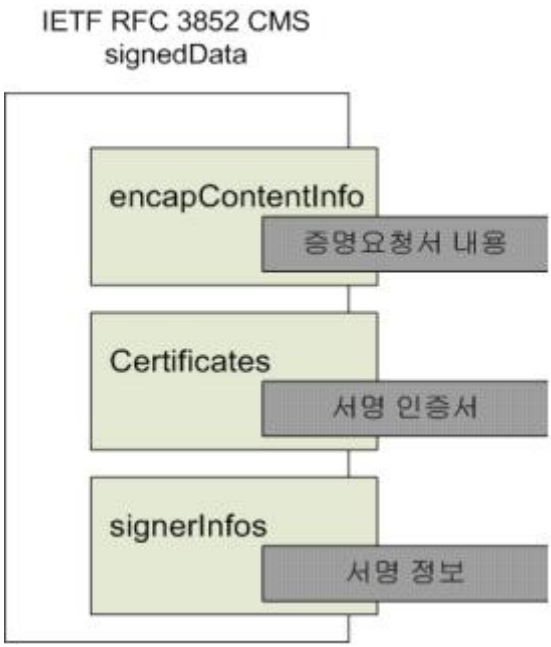
4. 증명서 요청 메시지

4.1 요청 메시지 구조

시점확인 증명요청서를 제외한 모든 증명요청서는 전자서명 구조인 IETF RFC 3852 CMS (Cryptographic Message Syntax)에서 제시하는 ContentInfo 구조체로 표현된 signedData를 사용한다. signedData는 일반적으로 많이 사용되는 구조이며, 다양한 추가 정보 및 기능을 부여할 수 있는 기능을 가지고 있다.

CMS는 기본적으로, 인코딩 방식은 ASN.1 Basic Encoding Rules(BER)을 따르며, 일부 정보에 대해서는 Distinguished Encoding Rules(DER)을 요구할 수도 있다. BER, DER을 요구하는 곳에서는 JSON Encoding Rules(JER), XML Encoding Rules(XER) 인코딩 방식 또한 사용할 수 있다.

CMS의 signedData는 다음과 같은 구조로 되어 있다.



SignedData 구조는 서명 대상, 서명 인증서 및 서명정보 등으로 구성되어 있다.

```
ContentInfo ::= SEQUENCE {
        contentType ContentType,
        content [0] EXPLICIT ANY DEFINED BY contentType }
```

ContentType ::= OBJECT IDENTIFIER

id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2 }

SignedData ::= SEQUENCE {

version	CMSVersion,
digestAlgorithms	DigestAlgorithmIdentifiers,
encapContentInfo	EncapsulatedContentInfo,
certificates	[0] IMPLICIT CertificateSet OPTIONAL,
crls	[1] IMPLICIT RevocationInfoChoices OPTIONAL,
signerInfos	SignerInfos }

증명요청서의 내용은 encapContentInfo에 포함되며, 증명요청서를 생성한 증명서 요청자의 전자서명은 signerInfos에 그리고 증명서 요청자의 인증서는 certificates에 포함된다.

전자서명을 첨부한 증명요청서의 경우 ContentInfo 구조체의 하위에 반드시 CMS의 signedData 형식을 사용하는 것과는 달리, 전자서명을 첨부하지 않은 증명요청서의 경우는 signedData의 하위 필드인 encapContentInfo에 설정되는 ARCCertRequest 구조체를 ContentInfo 구조체의 하위에 직접 설정할 수도 있다. 즉, 필요에 따라 증명요청서에는 전자서명이 첨부되지 않을 수도 있다.

증명요청서에 전자서명을 첨부하지 않는 경우, ContentInfo 구조체의 하위 필드인 contentType 필드와 content 필드에는, “4.1.1 encapContentInfo“ 항목에 정의된 id-kiec-arcCertRequest 및 “4.2.1 증명요청서 기본 필드“ 항목에 정의된 ARCCertRequest 구조체 형식을 사용하도록 한다. 다만, 첨부하지 않는다 해도 기존 형식을 따라서 작성도 가능하기 때문에 요청서를 수신하는 시스템에서는 두 가지 상황 모두 확인이 필요하다.

4.1.1 encapContentInfo

증명요청서의 실제 데이터인 ARCCertRequest 구조체를 포함하는 부분으로 무결성의 제공을 위하여 전자서명 되는 부분이다.

증명요청서 컨텐트의 구별을 위하여 식별자는 id-kiec-arcCertRequest를 사용한다.

id-kiec-arcCertRequest OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) 1 }

4.1.2 certificates

certificates 필드는 certificate의 집합으로 증명서 요청자가 증명요청서 서명에 사용한 인증서를 비롯하여 인증기관의 인증서와 최상위 인증기관 인증서를 포함 할 수 있다.

certificate의 형식은 공인인증체계에서 발급 하는 형식인 X.509 version 3 인증서를 의미한다.

certificates 필드는 선택적으로 사용 가능하지만, 본 규격에서는 증명서 요청자의 인증서를 포함하여 증명요청서를 생성하도록 한다.

4.1.3 signerInfos

signerInfos 필드는 signerInfo의 집합으로 서명자에 대한 정보를 나타내는 필드이다.

증명서 요청자는 증명요청서의 무결성을 위하여 signerInfo에 증명서 요청자의 전자서명을 포함할 수 있다.

signerInfo는 전자서명, 전자서명 알고리즘 및 속성들을 포함하고 있으며, 증명서 요청자는 CMS 표준에서 언급하고 있는 속성들을 추가적으로 사용 할 수 있다.

다음은 일반적으로 사용하는 서명 알고리즘의 객체식별자 이다.

iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-1(1) SHA1WithRSAEncryption(5)

4.2 증명요청서

시점확인 증명요청서를 제외한 모든 증명요청서는 ARCCertRequest를 포함한 signedData의 구조를 갖는다.

4.2.1 증명요청서 기본 필드

증명요청서 기본 필드는 증명 요청서의 확장 필드(extensions)를 제외한 부분으로 버전, 요청자, 요청 시간, 정책, 증적, 난수 등 증명 요청서 생성에 필요한 필수 정보를 나타내다.

증명요청서 기본 필드는 증명요청서에 반드시 포함되어야 하는 정보이다.

ARCCertRequest ::= SEQUENCE {

version	ARCVersion DEFAULT v1,
requester	Requester,
requestTime	RequestTime,
policy	ARCCertificatePolicies,
target	Target,
nonce	INTEGER,
extentions	[0] EXPLICIT Extensions OPTIONAL }

ARCCertRequest는 DER 인코딩 또는 JER, XER 인코딩 방식을 따르며, 각각의 항목들은 아래에서 설명 한다.

4.2.1.1 version, 버전

version 필드는 증명요청서의 버전을 표시한다.

target 필드에 targetHash 가 사용되면 버전 2를 사용하여야 하며, 그 외의 경우에는 버전 1을 사용하여야 한다.

ARCVerison ::= INTEGER { v1(1), v2(2)}

4.2.1.2 requester, 증명서 요청자

requester 필드는 증명서 발급을 요청한 주체를 나타낸다.

시점확인증명서 발급 요청의 경우를 제외하고 모든 증명서 발급요청 시, GeneralNames를 사용하여 증명서 요청자 정보를 설정하도록 하며, 시점확인증명서 발급 요청의 경우, 특별히 증명서 요청자를 명시할 필요가 없다면 NULL을 설정할 수 있다.

```
Requester      ::= CHOICE {
        generalNames            GeneralNames,
        null                    NULL    }
```

증명서 요청자 정보는 GeneralName 구조체의 otherName 필드를 이용하여, 증명서 요청자의 실명을 UTF8String으로 인코딩한 값과 증명서 요청자의 식별번호를 두 번 해시한 값을 사용하는 것을 기본으로 한다.

증명서 요청자가 사업자인 경우의 실명으로는 사업자명을 사용하며, 식별번호로는 사업자번호를 사용한다.

증명서 요청자가 개인이고 식별번호(CI/DI 포함)로 민감정보를 사용하게 될 경우 도용을 방지하기 위하여 안전한 난수를 생성하여 민감정보와 연접하여 사용하도록 한다. 단, 주민번호를 센터에 수집·보관·활용이 불가능하기 때문에 주민번호 사용에 유의해야한다. 민감정보가 아닌 것을 식별번호로 사용할 경우에는 사업자번호와 동일하게 처리가 가능하다.

requester 필드는 공인인증체계의 “전자서명 인증서 프로파일 기술규격”에서 정의된 IdentifyData 구조를 사용하는 공인인증서의 subjectAltName과 비슷한 구조로 되어 있으나 IdentifyData의 userInfo에 “식별번호를 이용한 본인확인 기술규격”에 정의된 VID 구조를 사용하지 않고 본 규격에서 정의하는 HashedIDNInfo 구조를 사용한다.

```
GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
        otherName                       [0] OtherName,
        rfc822Name                      [1] IA5String,
        dNSName                         [2] IA5String,
        x400Address                     [3] ORAddress,
        directoryName                   [4] Name,
        ediPartyName                    [5] EDIPartyName,
        uniformResourceIdentifier       [6] IA5String,
        iPAddress                       [7] OCTET STRING,
```

```
        registeredID                    [8] OBJECT IDENTIFIER }

  OtherName ::= SEQUENCE {
            type-id       OBJECT IDENTIFIER,
            value         [0] EXPLICIT ANY DEFINED BY type-id }

  id-kisa-identifyData OBJECT IDENTIFIER ::=  { id-attribute 1  }

  IdentifyData ::= SEQUENCE  {
        realName        UTF8String,
        userInfo        SEQUENCE SIZE (1..MAX) OF
                        AttributeTypeAndValue OPTIONAL  }

  id-kiec-HashedIDNInfo OBJECT IDENTIFIER ::=  { iso(1) member-body(2)
korea(410) kiec(200032) certificate(2) attributes(4) 1 }

  HashedIDNInfo::= SEQUENCE  {
        hashAlg         HashAlgorithm,
        hashedIDN       OCTET STRING  }

  HashAlgorithm ::= AlgorithmIdentifier
```

realName 필드에는 한글 실명을 사용하여야 한다.

hashedIDN은 증명서 요청자의 식별번호를 해시 알고리즘인 hashAlg를 사용하여 두 번 해시하여 계산된 값이며, 식별번호는 '-' 등과 같은 구분자를 제거한 후 PrintableString 으로 표현된 값을 사용하도록 한다.

이때 PrintableString에서 DER 인코딩을 위한 Type 과 Length에 대한 부분은 제거하고 Value 부분만을 사용하여야 한다.

requester의 신원 정보를 나타내는 IdentifyData의 자세한 생성방법 및 검증방법은 "부록 2. IdentifyData 구조체의 생성 및 검증"을 참조하도록 한다.

공인전자문서센터는 증명요청서를 수신하여 검증하는 과정 중에 반드시 requester 필드의 IdentifyData에 대한 검증을 수행해야 한다.

4.2.1.3 requestTime, 증명서 요청 시간

requestTime 필드는 증명서 요청 시간을 의미한다.

시점확인증명서 발급 요청의 경우를 제외하고 모든 증명서 발급 요청 시, GeneralizedTime 형식을 사용하여 증명서 요청 시간을 설정하도록 하며, 시점확인증명서 발급 요청의 경우, 특별히 증명서 요청 시간을 명시할 필요가 없다면 NULL을 설정할 수 있다.

```
RequestTime ::= CHOICE {
        generalizedTime         GeneralizedTime,
        null                    NULL    }
```

증명요청서의 증명서 요청 시간 값이 잘못된 시간이거나 증명시스템의 시간과 차이가 나는 경우에 증명시스템은 증명서 발급에 실패한 것으로 처리하고, 증명서 요청자에게 에러 응답 메시지를 전송해야 한다.

4.2.1.4 policy, 증명서 정책

policy 필드는 증명서 정책을 나타낸다.

증명서 요청자는 발급받고 싶은 증명서 정책을 선택하여 증명서를 요청 할 수 있다. 이 필드는 공인전자문서센터에 의하여 반드시 검토되어야 하며, 공인전자문서센터가 정책을 알고 있어야 한다.

정의된 구조 중에 policyQualifiers는 사용하지 않는다.

```
ARCCertificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
                policyIdentifier     CertPolicyId,
                policyQualifiers SEQUENCE SIZE (1..MAX) OF
                                        PolicyQualifierInfo OPTIONAL }
```

```
CertPolicyId ::= OBJECT IDENTIFIER
```

4.2.1.5 target, 증명대상

target 필드는 증명서 요청자가 증명서를 통하여 증명받고자 하는 대상을 포함하는 필드이며, 증명받고자 하는 대상에 따라 증적의 식별정보, 데이터의 해시정보, 또는 발급하고자 하는 전자문서의 식별정보 중 하나로 구성될 수 있다.

```
Target ::= CHOICE {
    targetRecord        TargetRecord,
    targetHash          [0] EXPLICIT HashedDataInfo,
    targetDocInfo       [1] EXPLICIT TargetDocInfo }
```

targetRecord 필드는 공인전자문서센터가 제공한 서비스에 대한 기록이자 증명서 요청자가 증명서를 통하여 증명받고자 하는 증적의 식별자를 포함한다.

등록증명서, 발급증명서, 이관증명서, 또는 폐기증명서를 발급받기 위한 증명요청서인 경우는 targetRecord 필드를 사용하여 target 필드를 설정하여야 한다.

```
TargetRecord ::= SEQUENCE {
    serialNo            INTEGER,
    opType              OperationType }

OperationType ::= ENUMERATED {
    register            (0),
    issue               (1),
    transfer            (2),
    delete              (3) }
```

serialNo 필드는 공인전자문서센터가 제공한 서비스를 기록한 증적의 식별번호로서, 발급될 증명서의 target 필드에 포함되는 OperationRecord 구조체의 하위 필드인 serialNo 필드와 동일한 값이어야 한다.

공인전자문서센터는 증적의 serialNo 생성 시, 순차적으로 증가하는 양의 정수를 사용하여 최대 20byte까지 생성할 수 있어야 하고, 이용자 소프트웨어는 최대 20byte 길이의 serialNo를 처리할 수 있어야 한다.

opType 필드는 공인전자문서센터가 제공한 서비스의 종류, 즉 작업 종류를 의미하며, 역시 발급될 증명서의 target 필드에 포함되는 OperationRecord 구조체의 하위 필드인 opType 필드와 동일한 값이어야 한다.

각 서비스의 의미는 다음과 같다.

- □ register : 전자문서 등록
- □ issue : 전자문서 발급
- □ transfer : 전자문서 이관
- □ delete : 전자문서 폐기

targetHash 필드는 어떤 데이터가 특정 시각에 존재하였음에 대한 증명을 받고자 할 때 사용되는 필드로서 이는 HashedDataInfo 구조체 형식으로 표현된다.

시점확인증명서를 발급받기 위한 증명요청서인 경우는 targetHash 필드를 사용하여 target 필드를 설정하여야 한다.

```
HashedDataInfo ::= SEQUENCE {
        hashAlg             HashAlgorithm,
        hashedData          BIT STRING      }
```

hashedData 필드는 해시 알고리즘인 hashAlg를 사용하여 증명받고자 하는 데이터를 한 번 해시 한 값이다.

targetDocInfo 필드는 이용자에게 발급되는 전자문서가 원본임을 증명하는 원본증명서를 발급받거나, 이용자에게 열람되는 변환본 전자문서의 내용이 원본과 동일함을 증명하는 불변경증명서를 발급받기 위하여 사용되는 필드로서, 대상 전자문서의 식별정보들을 포함한다.

```
TargetDocInfo   ::= SEQUENCE {
        packageID           PackageIdentifier,
```

```
    docID                [0] EXPLICIT DocumentIdentifier OPTIONAL,
    fileIDs              [1] EXPLICIT FileIDs OPTIONAL,
    issuedDocOriginal    BOOLEAN }

PackageIdentifier ::=  UTF8String

DocumentIdentifier ::=  UTF8String

FileIDs::= SEQUENCE SIZE (1..MAX) OF FileIdentifier

FileIdentifier ::=  UTF8String
```

packageID 필드는 발급 또는 열람하고자 하는 정보패키지의 패키지 식별자를 UTF8String 형식으로 변환한 값을 설정한다.

docID 필드는 발급 또는 열람하고자 하는 정보패키지 내 전자문서 영역의 전자문서 식별자를 설정하는데, 전자문서 발급의 경우는 원본 전자문서 영역의 전자문서 식별자를 UTF8String 형식으로 변환한 값을 설정하도록 하고, 전자문서 열람의 경우는 열람하고자 하는 변환본 전자문서 영역의 전자문서 식별자를 UTF8String 형식으로 변환한 값을 설정하도록 한다.

FileIDs 구조는 전자문서 영역 내 복수개의 첨부파일 가운데 일부의 첨부파일 ID를 나타내며, 패키지 규격에 정의된 첨부파일 식별자를 UTF8String 형식으로 변환한 값의 목록으로 표현한다. 원본증명서의 fileIDs 필드는, 원본 전자문서 영역 내 복수개의 첨부파일 가운데 일부만을 발급한 경우 해당 첨부파일들의 식별자를 설정하도록 하고, 만약 원본 전자문서 영역 내 첨부파일이 한 개인 경우나 또는 복수개의 첨부파일 모두를 발급한 경우라면 본 필드를 생략한다. 불변경증명서는 변환본 전자문서에 대한 불변경 여부를 증명하기 위한 증명서이므로, 열람하고자 하는 변환본 영역 내에 변환되지 않은 원본 전자문서가 포함되어 있다면, 해당 전자문서 파일의 식별자는 제외되어야 한다. 즉, 불변경증명서의 fileIDs 필드는 변환본 전자문서 영역에 포함된 모든 전자문서들이 변환작업으로 생성된 변환본 전자문서들이며 이용자가 변환본 전자문서 영역 전체에 대하여 열람요청을 한 경우에 한하여 생략하도록 하며, 변환본 전자문서 영역에 하나 이상의 원본 전자문서가 포함되어 있거나, 이용자가 일부 변환본 전자문서에 대하여 열람요청을 한 경우는 열람되는 변환본의 파일 식별자를 설정하도록 한다. 이는 불변경증명서의 orgAndIssued 필드 생

성 시에도 동일하다.

issuedDocOriginal 필드는 원본증명서 요청인지 불변경증명서 요청인지를 구분하기 위한 정보로서, 원본 전자문서 발급에 따른 원본증명서를 요청하는 경우는 issuedDocOriginal 필드의 값을 TRUE로 설정하고, 변환본 전자문서 열람에 따른 불변경증명서를 요청하는 경우는 issuedDocOriginal 필드의 값을 FALSE로 설정한다.

변환본 전자문서 열람에 따른 불변경증명서 요청과 관련하여, 만약 문서등록 시 변환본 전자문서 영역이 생성되지 않았다면 상기의 DocID 필드를 생략하도록 하며, 공인전자문서센터는 이에 대하여, 공인전자문서센터의 열람 정책 상 열람 서비스 제공 시 변환을 수행한다면 정상적으로 열람 서비스 제공 및 불변경증명서를 발급하도록 하고, 변환을 수행하지 않는다면 열람 서비스 제공 및 불변경증명서 발급 오류를 리턴하도록 한다.

4.2.1.6 nonce, 임시 값

nonce 필드는 증명서 요청자가 생성한 임시 값을 나타낸다.

nonce 필드는 기본적으로 재사용 공격 및 추측 공격 등을 방지하기 위하여 사용된다.

nonce 필드는 임의의 20byte INTEGER 값으로 설정하도록 한다.

4.2.2 증명요청서 확장 필드

증명요청서에는 기본 필드 이외에 다음과 같은 확장 필드를 사용 할 수 있다.

확장 필드는 다음과 같은 구조로 되어 있다.

```
Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension

Extension  ::=  SEQUENCE  {
        extnID          OBJECT IDENTIFIER,
        critical        BOOLEAN DEFAULT FALSE,
        extnValue       OCTET STRING  }
```

Extension은 확장 필드의 종류를 나타내는 extnID와 중요도를 의미하는 critical 그리고 확장 필드의 값을 포함하는 extnValue로 구성된다.

확장 필드의 critical이 TRUE로 설정 되어 있을 경우, 공인전자문서센터는 확장

필드를 반드시 이해하고 처리할 수 있어야 한다.

4.2.2.1 Qualifications, 자격 부여

Qualifications 확장필드는 증명서를 전달받아 이용하는 주체와 위임될 역할에 대한 정보를 나타낸다.

증명서를 전달받아 이용하는 주체가 한정되어야 할 경우에 이 필드를 사용하여 증명서에 대한 수임자를 지정할 수 있으며, 이 경우 해당 수임자가 반드시 공인전자문서센터의 이용자일 필요는 없다.

이 필드를 사용하여 특정인에게만 유효한 증명서를 생성할 수 있으며, 또한 특정인에게 문서에 대한 접근 권한을 부여할 수 있다.

증명서 검증 시 의도하지 않은 검증 오류가 발생하지 않도록 하기 위하여, 특정인에게만 유효한 증명서가 필요하거나 또는 특정인에게 문서에 대한 접근 권한을 부여할 필요가 없다면 본 확장필드를 생성하지 않도록 한다.

본 필드의 용도 중 특정인에게 문서에 대한 접근 권한을 부여하기 위하여 본 필드를 사용한다면 반드시 등록증명서를 요청하여야 한다.

Qualifications 확장필드는 증명서의 수임자 정보인 nomineeInfo와 위임될 역할 또는 권한인 nomineeRole로 이루어진 qualification 필드가 복수 개로 나열될 수 있는 구조로 되어 있다.

```
id-kiec-qualifications OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 1 }

Qualifications ::= SEQUENCE SIZE (1..MAX) OF Qualification

Qualification ::= SEQUENCE {
        nomineeInfo     NomineeInfo,
        nomineeRole     NomineeRole   }
```

증명서 수임자 정보는 증명서 수임자인 nominee와 증명서 수임자의 인증서 정보인 nomineeCert로 이루어져 있으며, 두 필드 중 하나는 반드시 존재해야 한다.

증명서 수임자가 개인일 경우는 반드시 nomineeCert만을 사용해야 하며, 사업자일 경우는 nominee 필드 또는 nomineeCert 필드 둘 다 사용 가능하다.

```
NomineeInfo  ::= SEQUENCE {
        nominee                 [0] EXPLICIT GeneralNames OPTIONAL,
        nomineeCert             [1] EXPLICIT CertIdentifier OPTIONAL }

CertIdentifier  ::= CHOICE {
        issuerAndSerialNumber       [0] EXPLICIT IssuerAndSerialNumber,
        subjectKeyIdentifier        [1] EXPLICIT SubjectKeyIdentifier }

IssuerAndSerialNumber ::= SEQUENCE {
        issuer          Name,
        serialNumber    CertificateSerialNumber }

CertificateSerialNumber ::= INTEGER

SubjectKeyIdentifier ::= OCTET STRING
```

nominee 필드는 수임자의 실명 및 식별번호를 사용하여 requester 필드와 동일한 방법으로 생성한다.

nomineeCert 필드는 수임자의 인증서에서 추출한 정보를 사용하여 구성한다.

```
NomineeRole ::= BIT STRING  {
        onlyForNominee      (0),
        readDocument        (1),
        downloadDocument    (2)  }
```

nomineeRole 필드는 수임자의 역할 또는 권한을 나타낸다.

각 bit 값의 의미는 다음과 같다.

- □ onlyForNominee : 증명서의 정당한 수신자임을 나타낸다.
- □ readDocument : 공인전자문서센터에서 문서를 열람할 수 있음을 나타낸다.
- □ downloadDocument : 공인전자문서센터에서 문서를 다운로드할 수 있음을 나타낸다.

onlyForNominee 값의 경우, 모든 Qualification에 동시에 설정하거나 동시에 해제해야 함에 주의한다.

nomineeRole 필드에 readDocument나 downloadDocument 값을 설정할 경우 반드시 targetRecord 필드의 하위 필드인 opType 필드의 값은 register로 설정되어야 한다. 반대로, targetRecord 필드의 하위 필드인 opType 필드의 값이 register가 아닌 경우 nomineeRole 필드에 readDocument나 downloadDocument 값을 설정하지 않아야 한다.

패키지 규격에 근거하여, 열람권한을 의미하는 readDocument가 설정된 경우 수임자는 공인전자문서센터로부터 열람용 변환본 전자문서(그리고 이용자가 불변경증명서 발급을 요청한 경우 불변경증명서)를 공인전자문서센터의 열람서비스 제공방식에 따라 열람할 수 있으며, 발급권한을 의미하는 downloadDocument가 설정된 경우는 원본 전자문서 및 원본증명서가 첨부된 DIP를 다운받을 수 있다. 참고로 수임자가 downloadDocument 권한을 이용하여 원본 전자문서를 발급받은 경우에만 발급 증적이 생성됨에 주의한다.

본 필드 생성 시 critical의 값은 TRUE로 설정해야 한다.

4.2.2.2 UsageType, 증명서 이용환경

UsageType 확장필드는 증명서의 이용 환경에 따른 분류 방식을 나타낸다.

증명서 요청자는 UsageType 필드를 사용하여 증명서의 이용 환경을 한정할 수 있다.

id-kiec-usageType OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 2 }

```
UsageType ::=   BIT STRING   {
        online                  (0),
        mobile                  (1),
        paperEnable             (2) }
```

각 bit 값의 의미는 다음과 같다.

- □ online : online 환경에서 사용할 수 있음을 나타낸다.
- □ mobile : mobile 기기에서 사용할 수 있음을 나타낸다.

□ paperEnable : 종이증명서로 출력하여 사용할 수 있음을 나타낸다.

본 필드 생성 시 critical의 값은 FALSE로 설정해야 한다.

4.2.2.3 DateOfExpiration, 증명서 효력 만기일

DateOfExpiration 필드는 증명서 요청자가 증명서의 효력 만기일을 지정할 때 사용한다.

이 필드를 사용하지 않을 경우 공인전자문서센터가 증명서 정책에서 정의한 유효기간을 사용하여 효력 만기일이 설정된다.

또한 본 필드의 critical 값이 FALSE 이며, 요청한 증명서 효력 만기일이 정책상 효력 만기일을 초과하는 경우, 공인전자문서센터는 정책상의 유효기간을 사용하여 증명서 효력 만기일을 설정한다.

증명서 효력 만기일은 증명서 요청 시간보다 미래이어야 한다.

시점확인증명서의 경우는 본 필드를 설정하지 않아야 한다.

id-kiec-dateOfExpiration OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 3 }

DateOfExpiration ::= GeneralizedTime

본 필드 생성 시 critical의 값은 TRUE 또는 FALSE가 가능하다.

4.2.2.4 CertifiedTime, 보증 일시

CertifiedTime 필드는, 전자문서가 등록 시점부터 일정 기간 동안 공인전자문서센터에 보관되어 있었음을 이용자가 보증받기를 원할 경우, 해당 기간의 마지막 일시를 기술하도록 한다.

보증 일시는 증명서 요청 시간보다 미래일 수는 없다.

만약 본 필드가 포함된 증명요청서가 정상적으로 처리되어 본 필드와 동일한 값이 설정된 증명서가 발급되었다면, 해당 증명서는 전자문서 등록시점부터 본 필드에 기재된 시점까지 전자문서가 공인전자문서센터에 보관되어 있었음을 보증하게 된다.

본 필드는 반드시 등록증명서를 요청할 경우에만 생성되어야 하며, 다른 종류의 증명요청서에는 포함될 수 없다.

id-kiec-certifiedTime OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 4 }

CertifiedTime ::= GeneralizedTime

본 필드 생성 시 critical의 값은 TRUE로 설정해야 한다.

4.2.2.5 CertUsage, 증명서 용도

CertUsage 확장필드는 증명서의 용도를 나타낸다.

증명서 요청자는 CertUsage 필드에 증명서의 용도를 명시할 수 있다.

id-kiec-certUsage OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 5 }

CertUsage ::= BMPString (SIZE (1..128))

CertUsage 필드는 이용자가 직접 기재한 값을 BMPString 형식으로 변환하여 설정하도록 한다.

본 필드 생성 시 critical의 값은 TRUE 또는 FALSE가 가능하다.

4.2.2.6 DocContentInfoFlag, 전자문서 정보설정

DocContentInfoFlag 확장필드는 증명서에 포함할 전자문서 정보들을 선택하기 위하여 사용한다.

증명서 요청자는 연관된 전자문서 정보를 증명서에 포함시켜 발급해 줄 것을 요청할 수 있으며, 본 확장필드를 사용하여 증명서에 포함시킬 전자문서 정보를 선택할 수 있다.

id-kiec-docContentInfoFlag OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 6 }

DocContentInfoFlag ::= BIT STRING {

title (0),

keyword (1),

description (2) }

각 bit 값의 의미는 다음과 같다.

- □ title : 증명서의 증명대상 필드에서 사용되는 docContentInfo 필드 하위에 title 필드를 생성한다.
- □ keyword : 증명서의 증명대상 필드에서 사용되는 docContentInfo 필드 하위에 keyword 필드를 생성한다.
- □ description : 증명서의 증명대상 필드에서 사용되는 docContentInfo 필드 하위에 description 필드를 생성한다.

본 확장필드는 이용자의 요청에 의하여 생성되어야 하며, 만약 증명요청서에 본 확장필드가 생성되지 않았다면, 증명서의 증명대상 필드 하위에 docContentInfo 필드가 생성될 수 없다. 즉, 공인전자문서센터는 이용자의 요청없이 증명서에 전자문서 정보를 포함시킬 수 없다.

시점확인증명서의 경우는 본 필드를 설정하지 않아야 한다.

본 필드 생성 시 critical의 값은 TRUE로 설정해야 한다.

4.2.2.7 CertVersion, 증명서 버전

CertVersion 확장필드는 증명서의 버전을 나타낸다.

증명서 요청자는 CertVersion 필드에 증명서의 버전을 명시할 수 있다.

id-kiec-CertVersion OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 7 }

CertVersion ::= INTEGER

CertVersion 필드는 이용자가 직접 기재한 값을 INTEGER 형식으로 설정하도록 한다.

본 필드 생성 시 critical의 값은 TRUE 또는 FALSE가 가능하다.

5. 증명서 응답 메시지

증명서 요청에 대하여 공인전자문서센터는 증명서 발급을 하거나 거부 또는 실패에 대한 정보를 증명서 요청자에게 전달한다.

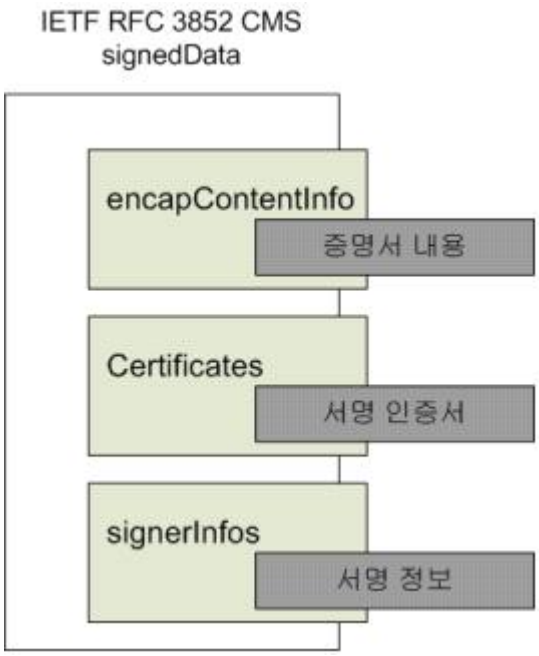
이 장에서는 증명서 요청에 대하여 증명서 요청자에게 전달하는 응답 메시지의 구조와 증명서 발급 실패에 대한 원인 분류를 한다.

5.1 응답 메시지 구조

공인전자문서센터가 보내는 응답메시지도 증명서 요청 메시지와 같이, 전자서명 구조인 IETF RFC 3852 CMS (Cryptographic Message Syntax)에서 제시하는 ContentInfo 구조체로 표현된 signedData를 사용한다.

인코딩 방식도 증명서 요청 메시지와 같은 ASN.1 Basic Encoding Rules(BER)을 따르며, 일부 정보에 대하여는 Distinguished Encoding Rules(DER)을 요구할 수도 있다. 또한, 증명서 요청 메시지와 같이 모든 영역에서 JSON Encoding Rules(JER), XML Encoding Rules(XER) 인코딩 방식을 사용할 수 있다.

CMS의 signedData는 다음과 같은 구조를 가지고 있다.



```
ContentInfo ::= SEQUENCE {
        contentType ContentType,
        content [0] EXPLICIT ANY DEFINED BY contentType }

ContentType ::= OBJECT IDENTIFIER

id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
rsadsi(113549) pkcs(1) pkcs7(7) 2 }

SignedData ::= SEQUENCE {
        version             CMSVersion,
        digestAlgorithms    DigestAlgorithmIdentifiers,
        encapContentInfo    EncapsulatedContentInfo,
        certificates        [0] IMPLICIT CertificateSet OPTIONAL,
        crls                [1] IMPLICIT RevocationInfoChoices OPTIONAL,
        signerInfos         SignerInfos }
```

응답 메시지의 내용은 encapContentInfo에 포함되며, 응답 메시지를 생성한 공인전자문서센터의 전자서명은 signerInfos에 그리고 공인전자문서센터의 인증서는 certificates에 포함된다.

5.1.1 encapContentInfo

응답 메시지의 실제 데이터인 ARCCertResponse 구조체를 포함하는 부분으로 무결성의 제공을 위하여 전자서명 되는 부분이다.

응답 메시지 콘텐츠의 구별을 위하여 식별자는 id-kiec-arcCertReseponse 를 사용한다.

```
id-kiec-arcCertReseponse OBJECT IDENTIFIER ::=  { iso(1) member-body(2)
korea(410) kiec(200032) certificate(2) 2 }

ARCCertResponse ::= CHOICE {
```

```
arcCertInfo          [0] EXPLICIT ARCCertInfo ,
arcErrorNotice       [1] EXPLICIT ARCErrorNotice }
```

ARCCertResponse는 증명 정보인 arcCertInfo 또는 에러 정보인 arcErrorNotice를 포함하며 각각에 대한 정보는 5.2절 및 5.3절에서 자세히 기술한다.

5.1.2 certificates

certificates 필드는 certificate의 집합으로 공인전자문서센터가 응답 메시지 서명에 사용한 인증서를 비롯하여 인증기관의 인증서와 최상위 인증기관 인증서를 포함할 수 있다.

certificate의 형식은 공인인증체계에서 발급하는 형식인 X.509 version 3 인증서를 의미한다.

certificates 필드는 선택적으로 사용 가능하지만, 본 규격에서는 공인전자문서센터의 인증서를 포함하여 응답메시지를 생성하도록 한다.

5.1.3 signerInfos

signerInfos 필드는 signerInfo의 집합으로 서명자에 대한 정보를 나타내는 필드이다.

공인전자문서센터는 응답 메시지의 무결성을 위하여 signerInfo에 공인전자문서센터의 전자서명을 포함한다.

signerInfo는 전자서명, 전자서명 알고리즘 및 속성들을 포함하고 있으며, 공인전자문서센터는 CMS 표준에서 언급하고 있는 속성들을 추가적으로 사용 할 수 있다.

다음은 일반적으로 사용하는 서명 알고리즘의 객체식별자 이다.

iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-1(1) SHA1WithRSAEncryption(5)

5.2 증명서

증명서는 ARCCertResponse의 ‘arcCertInfo’를 포함한 signedData의 구조를 갖는다.

5.2.1 증명서 기본 필드

증명서 기본 필드는 증명서의 확장 필드(extensions)를 제외한 부분으로 버전, 일련번호, 발급자, 발급시간, 만료시간, 정책, 요청자 정보, 증적 등 증명서 생성에 필요한 필수 정보를 나타낸다.

증명서 기본 필드는 증명서에 반드시 포함되어야 하는 정보이다.

```
ARCCertInfo ::= SEQUENCE {
        version                 [0] EXPLICIT ARCVersion DEFAULT v1,
        serialNumber            SerialNumber,
        issuer                  GeneralNames,
        dateOfIssue             GeneralizedTime,
        dateOfExpiration        CertDateOfExpiration,
        policy                  ARCCertificatePolicies,
        requestInfo             RequestInfo,
        target                  TargetToCertify,
        extentions              [1] EXPLICIT Extensions OPTIONAL }
```

ARCCertInfo는 DER 인코딩 또는 JER, XER 인코딩 방식을 따르며, 각각의 항목들은 아래에서 설명한다.

5.2.1.1 version, 버전

version 필드는 증명서의 버전을 표시한다.

target 필드에 dataHash 가 사용되면 버전 2를 사용하여야 하며, 그 외의 경우에는 버전 1을 사용하여야 한다.

5.2.1.2 serialNumber, 일련 번호

serialNumber 필드는 공인전자문서센터가 발급하는 증명서의 식별번호를 나타낸다. 일련번호는 동일한 번호로 발급되면 안 되며, 양의 정수 값만을 사용하여야 한다.

공인전자문서센터는 증명서의 일련번호를 최대 20byte까지 생성할 수 있어야 하고, 이용자 소프트웨어는 최대 20byte 길이의 일련번호를 처리할 수 있어야 한다.

SerialNumber ::= INTEGER

5.2.1.3 issuer, 증명서 발급자

issuer 필드는 증명서를 발급하는 주체인 공인전자문서센터를 의미하며, 작은 의미로는 증명서를 발급하는 증명시스템을 의미한다.

증명서 발급자 필드를 이용하여 검증하려는 증명서를 발급한 공인전자문서센터를 식별할 수 있다.

issuer 필드는 공인전자문서센터의 실명 및 식별번호를 사용하여 증명요청서의 requester 필드와 동일한 방법으로 생성한다.

5.2.1.4 dateOfissue, 증명서 발급일

dateOfIssue 필드는 공인전자문서센터가 사용자의 증명서 발급 요청에 대하여 증명서를 생성한 시점을 표현한다. 증명서 발급일은 GeneralizedTime 형식을 사용한다.

5.2.1.5 dateOfExpiration, 증명서 효력 만기일

dateOfExpiration 필드는 증명서의 효력 만기일을 의미하며, 일반적으로 증명서 정책을 기반으로 증명서의 효력 만기일이 정해진다.

시점확인증명서 발급의 경우를 제외하고 모든 증명서 발급 시, 본 필드 하위의 dateOfExpiration 필드를 사용하여 증명서 효력 만기일을 설정하도록 하며, 시점확인증명서 발급 시에는 NULL을 설정하여야 한다. 즉, 시점확인증명서의 효력 만기일은 없으며, 다만 증명서 유효성 검증 과정 중의 하나인 서명 인증서 검증 단계에서 증명서에 전자서명을 수행한 서명 인증서의 유효기간을 검증함으로써 해당 증명서의 효력기간을 판단하게 된다.

```
CertDateOfExpiration ::= CHOICE {
        dateOfExpiration        DateOfExpiration,
        null                    NULL    }
```

증명요청서에 dateOfExpiration 확장 필드가 존재하고 critical 값이 TRUE인 경우, 공인전자문서센터는 증명요청서의 dateOfExpiration 필드의 값을 사용하여 dateOfExpiration 필드의 값을 설정해야 한다.

증명요청서의 dateOfExpiration 필드의 값이 TRUE이나, 어떤 사유로 증명요청서

의 dateOfExpiration 필드값과 동일하게 설정하지 못하는 경우는, 증명서 발급에 실패한 것으로 처리하고, 증명서 요청자에게 에러 응답 메시지를 전송해야 한다.

증명요청서의 dateOfExpiration 필드의 critical 값이 FALSE로 설정되었고, 필드값이 증명서 정책상 효력 만기일을 초과하는 경우, 공인전자문서센터는 정책상의 유효기간을 사용하여 증명서 효력 만기일을 설정한다.

증명요청서의 dateOfExpiration 필드의 critical 값이 FALSE로 설정되었고, 필드값이 증명서 정책상 효력 만기일 내에 있다면, 공인전자문서센터는 해당 값을 사용하거나 또는 정책상 유효기간을 사용하여 증명서 효력 만기일을 설정하는 것이 모두 가능하다.

전자문서 등록 시에 발급되어지는 최초등록증명서의 효력 만기일은 대응되는 전자문서에 설정된 보존 만료일(RetentionExpiredDate)과 동일해야 함에 주의한다.

증명서의 사용에 있어서 증명서의 효력 기간은 증명서 발급일로부터 증명서 효력 만기일 사이이다.

단, 증명서 효력 만기일 내에 있으나, 증명서 서명 인증서의 유효기간이 곧 만료되거나 이미 만료되어 증명서의 유효성을 보증 하지 못하는 경우에, 이용자는 기존 증명서를 갱신된 인증서로 재서명하여 증명서를 갱신해 줄 것을 공인전자문서센터에 요청 할 수 있고, 공인전자문서센터는 이에 대하여 갱신 요청받은 증명서가 해당 공인전자문서센터가 발급한 증명서임을 반드시 확인한 후, 동일한 효력 만기일의 증명서로 갱신해 주어야 한다.

증명서 갱신의 의미는 증명서에 첨부된 서명 및 서명에 사용된 공인전자문서센터 인증서의 정보를 갱신하여 증명서의 유효성을 연장하는 것으로서, 갱신 전 증명서의 내용과 갱신 후 증명서의 내용은 변경되지 않는다.

따라서 공인전자문서센터에 갱신을 요청하기 위하여 이용자가 별도의 증명요청서를 생성할 필요는 없으며, 갱신된 증명서의 arcCertInfo 필드의 내용은 이전 증명서의 arcCertInfo의 내용과 동일해야 한다.

증명서 갱신을 위한 재서명 시, 공인전자문서센터는 증명서에 첨부된 이전 서명정보를 제거하고 갱신된 공인전자문서센터 인증서를 사용하여 생성된 새로운 서명정보를 추가하도록 한다.

5.2.1.6 policy, 증명서 정책

policy 필드는 증명서 정책을 나타낸다.

증명서의 정책은 증명서를 발급하는 공인전자문서센터의 해당 증명서에 대한 운영 정책 및 이용자의 증명서 이용 범위를 포함한다.

공인전자문서센터는 기본 증명서 정책으로 등록증명서 정책, 발급증명서 정책, 이

관증명서 정책, 폐기증명서 정책, 원본증명서 정책, 불변경증명서 정책, 시점확인증명서 정책을 정의하고 각 정책에 OID를 부여하여야 하며, 필요에 따라 각 기본 증명서 정책의 하위에 세부적인 증명서 정책을 정의하여 사용할 수 있다.

또한 공인전자문서센터는 부가적으로 시점확인증명서를 발급하기 위한 시점확인증명서 정책을 정의하고 OID를 부여하여 사용할 수 있다.

증명서의 정책에 따라서 증명서의 용도 및 효력이 결정되므로 증명서를 이용하는 모든 주체는 증명서 정책에 기술된 정책 내용에 대하여 인지하여야 한다.

ARCCertificatePolicies는 PolicyInformation 구조의 집합으로 이루어져 있으며, 다시 PolicyInformation은 증명서 정책을 기술하는 policyIdentifier와 이를 보다 자세하게 기술하는 policyQualifiers로 구성된다.

policyQualifiers에는 업무준칙에 대한 공시를 위한 URI 주소를 나타내는 cPSuri와 사용자에게 간단한 정책 관련 정보를 제공하는 userNotice가 포함된다.

본 버전의 규격에서는 cPSuri를 사용하여 표현된 PolicyQualifierInfo 구조를 반드시 포함하여야 하며, userNotice를 사용하여 표현된 PolicyQualifierInfo 구조는 선택적으로 포함 가능하다. userNotice를 사용할 경우, noticeRef는 생성하지 않으며, explicitText의 형식은 BMPString을 사용하여야 한다.

```
ARCCertificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
        policyIdentifier        CertPolicyId,
        policyQualifiers        SEQUENCE SIZE (1..MAX) OF
                                        PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
        policyQualifierId       PolicyQualifierId,
        qualifier               ANY DEFINED BY policyQualifierId }

PolicyQualifierId ::=  OBJECT IDENTIFIER ( id-qt-cps | id-qt-unotice )
```

```
Qualifier ::= CHOICE {
        cPSuri          CPSuri,
        userNotice      UserNotice }

CPSuri ::= IA5String

UserNotice ::= SEQUENCE {
        noticeRef       NoticeReference OPTIONAL,
        explicitText    DisplayText OPTIONAL}

NoticeReference ::= SEQUENCE {
        organization    DisplayText,
        noticeNumbers   SEQUENCE OF INTEGER }

DisplayText ::= CHOICE {
        ia5String       IA5String       (SIZE (1..200)),
        visibleString   VisibleString   (SIZE (1..200)),
        bmpString       BMPString       (SIZE (1..200)),
        utf8String      UTF8String      (SIZE (1..200)) }
```

5.2.1.7 requestInfo, 증명서 요청 메시지 정보

requestInfo 필드는 증명서 요청자가 생성한 증명서 요청 메시지를 나타낸다.

증명서 요청자가 증명서를 요청하는 일반적인 증명서 요청은 반드시 증명요청서의 ARCCertRequest를 사용하고, 최초등록증명서와 같이 증명요청서가 존재하지 않는 경우는 NULL을 사용한다.

```
RequestInfo  ::= CHOICE {
    arcCertRequest      ARCCertRequest,
```

```
        null                    NULL    }
```

5.2.1.8 target, 증명대상

target 필드는 공인전자문서센터가 제공한 서비스에 대한 증적이나 공인전자문서센터가 증명서를 통하여 보증하고자 하는 사실을 나타내는 필드이다.

공인전자문서센터는 이용자들이 요청한 등록, 발급, 이관, 폐기 서비스에 대하여 처리가 완료된 시점을 기준으로 서비스의 내역을 OperationRecord 구조체의 형식으로 생성하여 증적으로서 기록하고, 증명서 발급 시 증명의 대상으로서 증명서에 포함한다.

또한 공인전자문서센터는 전자문서 발급 또는 열람 서비스 제공 시, 발급된 원본 전자문서가 보관중인 원본 전자문서와 동일함을 보증하거나, 열람되는 변환본 전자문서의 내용이 원본 전자문서의 내용과 동일함을 보증하기 위하여 원본 전자문서 및 변환본 전자문서의 정보를 증명대상으로 증명서에 포함하여 발급한다.

그리고 공인전자문서센터는 증명서에 증명서 요청자가 보내온 해시정보를 포함하여 발급할 수 있는데, 이는 어떤 데이터가 특정 시각에 존재하였음을 증명하는 기능을 제공한다.

```
TargetToCertify   ::= CHOICE {
        opRecord            [0] EXPLICIT OperationRecord,
        orgAndIssued        [1] EXPLICIT OriginalAndIssuedDocumentInfo,
        dataHash            [2] EXPLICIT HashedDataInfo }
```

본 필드를 구성하는 TargetToCertify 구조체의 하위 각 필드는 증명요청서의 target 필드를 구성하는 Target 구조체에 사용된 하위 각 필드에 상응하여 생성되어야 한다.

즉, 증명요청서의 target 필드에 targetRecord 필드가 사용되었다면, 본 필드에는 opRecord 필드를 설정하여야 하고, targetHash 필드인 경우는 dataHash 필드를, targetDocInfo 필드인 경우는 orgAndIssued 필드를 각각 설정하여야 한다.

opRecord 필드는 공인전자문서센터가 서비스를 수행하고 기록한 증적 정보로서 등록증명서, 발급증명서, 이관증명서, 폐기증명서 발급 시에 설정하도록 하며, orgAndIssued 필드는 공인전자문서센터에 보관중인 원본 전자문서 정보와 발급 또는 열람되는 전자문서 정보로 구성된 필드로서 원본증명서 및 불변경증명서 발급 시에 설정하도록 한다.

dataHash 필드는 시점확인증명서 발급 시 증명요청서에 기재된 데이터 해시정보

를 사용하여 설정하도록 한다.

OperationRecord 구조체는 증적 정보를 기술하기 위한 구조체로서 하기의 구조를 갖는다.

```
OperationRecord  ::= SEQUENCE {
        serialNo            INTEGER,
        opRequesterInfo     OperationRequesterInfo,
        opRequestTime       GeneralizedTime,
        opTime              GeneralizedTime,
        opType              OperationType,
        orgDocInfo          PackageDocumentInfo,
        issuedDocInfo       [0] EXPLICIT PackageDocumentInfo OPTIONAL,
        peerARCInfo         [1] EXPLICIT PeerARCInfo OPTIONAL,
        reason              [2] EXPLICIT Reason OPTIONAL }
```

각 필드의 의미는 다음과 같다.

- □ serialNo : 증적의 식별번호
- □ opRequesterInfo : 전자문서 서비스 요청자 정보
- □ opRequestTime : 전자문서 서비스 요청 시간.
- □ opTime : 공인전자문서센터의 서비스 제공 시간, 즉 작업 시간
- □ opType : 공인전자문서센터가 제공한 서비스 종류, 즉 작업 종류
- □ orgDocInfo : 공인전자문서센터에 보관 중인 원본 문서 관련 정보
- □ issuedDocInfo : 발급 또는 열람되는 문서 관련 정보
- □ peerARCInfo : 이·수관 시 상대 공인전자문서센터 관련 정보
- □ reason : 작업 사유

serialNo 필드와 opType 필드의 값은 증명요청서의 targetRecord 필드내의

serialNo 필드와 opType 필드의 값과 동일해야 한다.

```
OperationRequesterInfo ::= CHOICE {
        opRequester             GeneralNames ,
        null                    NULL }
```

opRequester 필드는 전자문서 관련 서비스 요청자의 신원정보로서, 등록, 발급, 이관, 폐기 등의 전자문서 서비스를 요청한 이용자의 실명 및 식별번호를 사용하여 증명요청서의 requester 필드와 동일한 방법으로 생성하며, 만약 서비스 요청자가 개인일 경우에는 민감정보(CI/DI 포함)의 도용을 방지하기 위하여, 공인전자문서센터가 직접 160 비트의 안전한 난수를 생성하여 requester 필드 생성 시에 사용하도록 한다. 단, 주민번호를 센터에서 수집·보관·활용이 불가능하기 때문에 주민번호 사용에 유의해야한다. 민감정보가 아닌 것을 식별번호로 사용할 경우에는 사업자번호와 동일하게 처리가 가능하다.

공인전자문서센터는 생성한 난수를 증적 구조체와 함께 보관하여 서비스 요청자에 대한 신원을 검증할 필요가 있을 때 검증 중에 사용하도록 한다.

서비스 요청자는 전자문서에 대한 등록, 발급, 이관, 폐기 서비스를 직접 요청한 이용자의 정보이므로 권한 위임 등을 통하여 해당 서비스가 행해진 경우, 전자문서 소유자가 아닌 실제 서비스 요청자의 정보가 설정되어야 함에 주의한다. 예를 들어 등록증명서의 자격 부여 필드를 이용하여 문서 발급 권한을 위임받은 수임자가 타인 소유의 원본 전자문서를 발급받은 경우, 해당 발급 증적의 서비스 요청자 정보에 실제 문서를 발급받은 수임자의 정보가 설정되어야 한다. 수임자가 공인전자문서센터 이용자가 아닌 경우 공인전자문서센터는 해당 수임자의 실명 및 식별번호를 획득하여 서비스 요청자 정보를 생성하도록 한다. 이 때 획득한 식별번호는 서비스 요청자 정보를 생성한 후 즉시 시스템에서 삭제되어야 하며, 시스템 외부로 노출 및 저장될 수 없다.

전자문서 이·수관 시 수관 공인전자문서센터의 등록 작업, 공인전자문서센터 간의 관계에서 비롯되어 이용자의 요청 없이 발생한 전자문서 이관 작업, 유효기간 만료 시의 전자문서 자동 폐기 작업 등의 경우처럼, 이용자가 전자문서 서비스를 요청하지 않은 경우에는 opRequester대신 null을 사용한다.

opRequestTime 필드는 공인전자문서센터가 서비스를 요청받은 시각을 설정하도록 하며, 만약 opRequesterInfo 필드가 null로 설정된 경우에는 opTime 필드와 동일한 값을 설정하도록 한다.

opTime 필드는 공인전자문서센터가 서비스를 제공한 시각으로서, 서비스 수행이

완료된 시각을 설정하도록 한다. 특별히 등록서비스의 경우에는 패키지 규격상 RegisterDateTime 필드에 설정된 시각값과 동일한 값이 설정되어야 한다. 이관서비스의 경우, 이관 공인전자문서센터가 이관작업을 시작한 시각이 아닌, 수관 공인전자문서센터의 응답메시지를 수신하여 정상적으로 이관작업을 완료하였음을 확인한 시각이 설정되어야 함에 주의한다.

orgDocInfo 필드는 공인전자문서센터에 보관중인 원본 전자문서의 정보를 포함하며, issuedDocInfo 필드는 원본 전자문서들 중에서 발급된 전자문서의 정보를 포함하는데, 전자문서 정보를 사용하여 PackageDocumentInfo 구조체의 형식으로 표현된다.

orgDocInfo 필드는 공인전자문서센터가 수행한 모든 서비스(등록, 발급, 이관, 폐기)에 대하여 첨부된 모든 원본 전자문서 파일의 정보를 이용하여 설정되어야 한다.

issuedDocInfo 필드는, 공인전자문서센터가 수행한 전자문서 발급 서비스에 대하여 첨부된 원본 전자문서 파일 중 발급된 전자문서 파일의 정보를 이용하여 설정된다.

예를 들어, 원본 전자문서가 10개의 첨부파일로 구성되어 있다면, 발급 서비스 증적을 포함한 모든 서비스 증적에서 orgDocInfo 필드의 값을 계산할 때 10개의 첨부파일을 모두 연접한 후 해시하여 생성하여야 하며, 그 가운데 5개의 첨부파일이 발급된 발급 서비스 증적에서의 issuedDocInfo 필드의 값은 발급된 5개의 첨부파일만을 연접한 후 해시하여 생성하여야 한다.

즉, 첨부된 모든 전자문서가 발급되었다면, 발급증명서의 orgDocInfo 필드 및 issuedDocInfo 필드의 하위 필드인 fileIDs 필드와 docHash 필드값은 완전히 동일하며, 일부만 발급되었다면 orgDocInfo 필드 및 issuedDocInfo 필드의 하위 필드인 fileIDs 필드와 docHash 필드값이 달라진다.

```
PackageDocumentInfo   ::= SEQUENCE {
        packageID             PackageIdentifier,
        docInfo               DocumentInfo }
```

packageID 필드는 관련 패키지 식별자를 UTF8String 형식으로 변환한 값을 설정하도록 하며, orgDocInfo 필드 및 issuedDocInfo 필드 모두 동일하다.

docInfo 필드 역시 orgDocInfo 필드와 issuedDocInfo 필드 모두에서 원본 전자문서의 정보를 참조하여 생성하도록 한다.

```
DocumentInfo   ::= SEQUENCE {
        docID                 DocumentIdentifier,
```

```
    fileIDs             [0] EXPLICIT FileIDs OPTIONAL,
    docHash             DocumentHash }
```

docID 필드는 관련 원본 전자문서 식별자를 UTF8String 형식으로 변환한 값을 설정한다.

fileIDs 필드는 전자문서 내의 첨부파일 중에서 관련된 첨부파일의 ID를 나타내며, 패키지 규격에 정의된 첨부파일 식별자를 UTF8String 형식으로 변환한 값의 목록으로 표현한다. 만약 전자문서 내의 모든 첨부파일에 해당된다면 fileIDs 필드 자체를 생략한다. 즉, 원본 전자문서 내의 모든 첨부파일에 해당되는 모든 증명서의 orgDocInfo 필드에서는 항상 fileIDs 필드를 생략하도록 하고 발급증명서의 issuedDocInfo 필드에서도 모든 원본 전자문서가 발급되었다면 fileIDs 필드를 생략한다.

docHash 필드는 전자문서의 해시값을 포함하는 DocumentHash 구조체 형식으로 표현된다.

```
DocumentHash ::= SEQUENCE {
    hashAlg             HashAlgorithm,
    hashedDocument      BIT STRING      }
```

hashedDocument 필드는 해시 알고리즘인 hashAlg를 사용하여 전자문서의 내용을 한 번 해시한 값이다. 전자문서 내에 포함된 실제 문서파일이 2개 이상인 경우는 두가지 방법으로 해시 할 수 있다.

첫 번째 방법은 각 문서파일의 내용을 연접하여 해시한다. 이때, 문서파일의 연접 순서는 관련된 전자문서 내에서의 순서와 동일하다. 패키지 규격 상 전자문서 보관이나 발급 시에 문서파일이 암호화되어 패키지에 첨부될 수 있는데, 이 경우에는 반드시 암호화 이전의 문서파일을 해시하여 hashedDocument에 설정하여야 한다.

두 번째 방법은 각 문서파일의 해시값(패키지에 포함된)을 연접하여 해시한다. 이때, 해시의 연접 순서는 첫 번째 방법과 동일하게 전자문서 포함된 순서와 동일하다. 암호화 여부는 전자문서의 패키지 내에 포함된 해시값이 어떻게 생성되었는지에 따라서 다르다. 반드시 패키지 내 포함된 해시의 암호화 여부를 확인하여 해시하여 hashedDocument에 설정하여야 한다.

두 번째 방법을 사용하게 되면 기존 기술규격 버전과의 호환성을 유지하기 위해 확장필드의 버전정보를 추가하여 구분이 되도록 해야 한다.

```
PeerARCInfo  ::= SEQUENCE {
```

```
        peerARC             GeneralNames,
        peerARCPackageID    PackageIdentifier }
```

peerARCInfo 필드는 전자문서 이·수관 발생 시, 수관 공인전자문서센터의 등록증명서 및 이관 공인전자문서센터의 이관증명서 생성 시에만 생성하는 필드이다.

peerARC 필드와 peerARCPackageID 필드는 전자문서의 이·수관 서비스를 수행한 경우, 이관을 수행한 공인전자문서센터가 발급한 이관증명서에는 수관을 수행한 공인전자문서센터의 신원정보 및 수관 공인전자문서센터에서 부여한 식별자를 포함하며, 수관을 수행한 공인전자문서센터가 발급한 등록증명서에는 이관을 수행한 공인전자문서센터의 신원정보와 이관 공인전자문서센터에서 부여한 식별자를 포함한다.

만약 opType의 값이 0(register)이며 peerARCInfo 항목이 존재한다면, 타 공인전자문서센터로부터 이관된 전자문서에 대한 등록증명서임을 알 수 있다.

peerARC 필드는 상대 공인전자문서센터의 실명 및 식별번호를 사용하여 증명요청서의 requester 필드와 동일한 방법으로 생성한다.

```
Reason  ::=  BIT STRING  {
        userRequest         (0),
        arcRequest          (1),
        expired             (2)       }
```

Reason 필드는 선택적 생성 필드로서, 작업이 발생한 사유를 설정하며 값의 의미는 아래와 같다.

- □ userRequest : 이용자 요청에 의한 작업
- □ arcRequest : 공인전자문서센터 내 프로세스 상 또는 공인전자문서센터 간에 발생한 작업
- □ expired : 보존기간 만료

OriginalAndIssuedDocumentInfo 구조체는 발급된 원본 전자문서 또는 열람되는 변환본 전자문서 정보를 기술하기 위한 구조체로서 하기의 구조를 갖는다.

```
OriginalAndIssuedDocumentInfo ::= SEQUENCE {
```

```
orgDocInfo          PackageDocumentInfo,
issuedDocInfo       PackageDocumentInfo,
issuedDocOriginal   BOOLEAN }
```

orgDocInfo 필드에 대한 설명은 OperationRecord 구조체에서의 설명과 동일하다. 즉, 공인전자문서센터에 보관중인 모든 원본 전자문서의 정보를 기술하며, 하위 필드인 fileIDs 필드는 생략된다.

원본 전자문서 발급 시 발급되는 원본증명서의 issuedDocInfo 필드에 대한 설명 역시 OperationRecord 구조체에서의 설명과 동일하다. 즉, 원본 전자문서들 중에서 발급된 전자문서의 정보를 기술하며, 만약 모든 원본 전자문서가 발급되었다면 하위 필드인 fileIDs 필드는 생략된다.

변환본 전자문서 열람 시 발급되는 불변경증명서의 issuedDocInfo 필드는 열람되는 변환본 전자문서들의 정보를 기술하도록 한다. 불변경증명서는 변환본 전자문서에 대한 불변경 여부를 증명하기 위한 증명서이므로, 열람하고자 하는 변환본 영역 내에 변환되지 않은 원본 전자문서가 포함되어 있는 경우, 해당 전자문서 파일의 식별자는 제외되어야 하며, docHash 필드에 포함되는 변환본 전자문서의 해시값 계산 시에도 제외되어야 한다. 만약 변환본 전자문서 영역에 포함된 모든 전자문서들이 변환작업으로 생성된 변환본 전자문서들이며, 이용자가 해당 변환본 전자문서 영역 전체에 대하여 열람요청을 하였다면, 하위 필드인 fileIDs 필드가 생략되나, 변환본 전자문서 영역에 하나 이상의 변환되지 않은 원본 전자문서가 포함된 상태에서 이용자가 변환본 전자문서 영역 전체에 대하여 열람요청을 하였다면, filesIDs에 원본 전자문서의 파일 식별자가 포함되지 않았음을 표시하기 위하여 각 변환본 전자문서에 대한 파일 식별자들을 기술하여야 한다.

공인전자문서센터의 정책상 전자문서를 등록하는 과정이 아닌 열람서비스를 제공하는 과정 중에 변환작업을 수행한다면 공인전자문서센터의 메타데이터에는 변환본 전자문서 영역이 존재하지 않기 때문에, issuedDocInfo 필드 하위의 docID 필드에는 공인전자문서센터 정책적으로 열람 과정 중에 변환작업이 수행되었음을 알 수 있는 유니크한 ID값을 정의하여 설정하도록 한다. fileIDs 필드 내의 첨부파일 ID를 기술할 필요가 있는 경우는 생성된 변환본 전자문서에 대응하는 원본 전자문서의 첨부파일 ID를 그대로 사용한다.

issuedDocOriginal 필드는 원본증명서인지 불변경증명서인지를 구분하기 위한 정보로서, 만약 원본문서 발급 시 함께 발급되는 원본증명서인 경우는 issuedDocOriginal 필드의 값을 TRUE로 설정하여 원본 전자문서에 대한 원본증명서임을 표시하고, 변환본 전자문서 열람 시 이용자의 요청에 의하여 발급되는 불변경증명서인 경우는 issuedDocOriginal 필드의 값을 FALSE로 설정하여 변환본 전자

문서에 대한 불변경증명서임을 표시하도록 한다.

참고로, 발급되거나 열람되는 전자문서의 무결성 검증을 위하여 원본증명서 또는 불변경증명서 내의 docHash 값과 비교 시에는, 오류를 방지하기 위하여 반드시 orgDocInfo 필드가 아닌 issuedDocInfo 필드 내의 docHash 값과 비교하도록 한다.

시점확인증명서를 발급받기 위한 증명요청서인 경우는 target 필드가 targetHash 필드를 사용하여 설정되어 있으며, 공인전자문서센터는 증명서 발급 시 해당 필드 정보를 사용하여 증명서의 target 필드를 설정하여야 한다.

즉, 증명서의 target 필드는 dataHash 필드를 사용하여 설정하여야 하며, 이는 증명요청서의 targetHash 필드에 설정된 값을 사용하여 동일하게 설정하도록 한다.

시점확인증명서를 발급하기 전에 공인전자문서센터는 증명서 요청자로부터 수신한 증명요청서를 검증하는 과정에서, targetHash 필드의 하위 필드인 hashAlg 필드에 기재된 해시 알고리즘 OID와 해당 해시 알고리즘을 사용하여 생성된 해시값인 hashedData 필드값의 길이가 합리적인가를 확인하여야 한다.

5.2.2 증명서 확장 필드

증명서에는 기본 필드 이외에 다음과 같은 확장 필드를 사용할 수 있다.

확장 필드는 다음과 같은 구조로 되어 있다.

```
Extensions   ::=   SEQUENCE SIZE (1..MAX) OF Extension

Extension   ::=   SEQUENCE   {
        extnID              OBJECT  IDENTIFIER,
        critical            BOOLEAN  DEFAULT  FALSE,
        extnValue           OCTET  STRING   }
```

Extension은 확장 필드의 종류를 나타내는 extnID와 중요도를 의미하는 critical 그리고 확장 필드의 값을 포함하는 extnValue로 구성된다.

확장 필드의 critical이 TRUE로 설정되었으면 증명서 이용자 또는 검증자는 이 확장 필드를 반드시 이해하고 처리할 수 있어야 한다.

5.2.2.1 Qualifications, 자격 부여

Qualifications 필드는 증명서를 전달받아 이용하는 주체와 위임될 역할에 대한 정보를 나타낸다.

본 필드는 증명요청서의 Qualifications 필드와 동일한 내용을 포함해야 하며, 증명요청서 없이 발급되는 최초등록증명서에는 본 필드가 생성될 수 없다.

증명요청서에 Qualifications 확장 필드가 존재하고 critical 값이 TRUE인 경우, 공인전자문서센터는 반드시 본 필드를 생성해야 한다.

본 필드를 생성하지 못하는 경우는 증명서 발급에 실패한 것으로 처리하고, 증명서 요청자에게 에러 응답 메시지를 전송해야 한다.

본 필드 생성 시 critical의 값은 TRUE로 설정해야 한다.

5.2.2.2 UsageType, 증명서 이용환경

UsageType 필드는 증명서의 이용 환경에 따른 분류 방식을 나타낸다. 본 필드는 증명요청서의 UsageType 필드와 동일한 내용을 포함한다. 본 필드 생성 시 critical의 값은 FALSE로 설정해야 한다.

5.2.2.3 CertifiedTime, 보증 일시

CertifiedTime 필드는, 전자문서 등록시점부터 본 필드에 기재된 시점까지 전자문서가 공인전자문서센터에 보관되어 있었음을 보증하는 필드로서, 등록증명서에만 포함될 수 있다.

본 버전의 규격을 준수하여 생성된 등록증명서는 CertifiedTime 필드를 필수적으로 생성하여야 한다. 단, 최초등록증명서에는 본 필드를 생성하지 않도록 한다.

증명요청서에 CertifiedTime 확장 필드가 존재하고 critical 값이 TRUE로 설정되었다면, 공인전자문서센터는 증명요청서의 CertifiedTime 필드에 기재된 보증 일시까지 전자문서를 보관 중이었음을 확인한 후, 해당 값을 증명서의 CertifiedTime 필드에 기재하여 증명서를 생성하도록 한다.

전자문서의 등록 시점부터 증명요청서에 기재된 보증 일시까지 전자문서가 공인전자문서센터에 보관되어 있지 않았다면, 공인전자문서센터는 증명서 요청자에게 에러 응답 메시지를 전송하여야 한다.

증명요청서에 CertifiedTime 필드값이 존재하지 않으면, 공인전자문서센터는 하기와 같은 절차로 CertifiedTime 확장 필드의 값을 설정한다.

□ 전자문서가 해당 전자문서의 보존 만료일 전에 폐기 또는 타 공인전자문서센터로 이관 후 폐기되었다면 해당 일시를 설정하도록 한다.

□ 전자문서가 해당 전자문서의 보존 만료일까지 정상적으로 보관되었다면 해당 일시를 설정하도록 한다.

□ 전자문서가 현재 공인전자문서센터에 보관중이라면 증명서 발급일 필드와 동일한 값을 설정하도록 한다.

본 버전의 규격에서는, 증명요청서에 CertifiedTime 확장 필드가 존재한다면, 반드시 critical 값을 TRUE로 설정해야 하므로, FALSE 인 경우는 다루지 않는다.

기타 사유로 본 필드를 생성하지 못하는 경우는 증명서 발급에 실패한 것으로 처리하고, 증명서 요청자에게 에러 응답 메시지를 전송해야 한다.

본 필드 생성 시 critical의 값은 TRUE로 설정해야 한다.

5.2.2.4 CertUsage, 증명서 용도

CertUsage 확장필드는 증명서의 용도를 나타낸다.

본 필드는 증명요청서의 CertUsage 확장필드를 반영하여 동일한 값으로 생성하도록 하며, 만약 증명요청서에 CertUsage 필드가 존재하고 critical 값이 TRUE인 경우에는 반드시 본 필드를 생성하도록 하며, critical 값이 FALSE라면 공인전자문서센터의 증명서 생성정책에 따라 생성하거나 생성하지 않도록 한다.

본 필드 생성 시 critical의 값은 증명요청서 CertUsage 확장필드의 critical 값과 동일하게 설정하도록 한다.

5.2.2.5 CertVersion, 증명서 버전

CertVersion 확장필드는 증명서의 버전을 나타낸다.

본 필드는 증명요청서의 CertVersion 확장필드를 반영하여 동일한 값으로 생성하도록 하며, 만약 증명요청서에 CertVersion 필드가 존재하고 critical 값이 TRUE인 경우에는 반드시 본 필드를 생성하도록 하며, critical 값이 FALSE라면 공인전자문서센터의 증명서 생성정책에 따라 생성하거나 생성하지 않도록 한다.

본 필드 생성 시 critical의 값은 증명요청서 CertVersion 확장필드의 critical 값과 동일하게 설정하도록 한다.

5.2.2.6 docContentInfo, 증명서 추가 정보

docContentInfo 확장필드는 증명서의 추가 정보를 나타낸다.

docContentInfo 필드는 증명서와 연관된 전자문서의 추가 정보를 기술하기 위한 필드로서, 반드시 이용자의 요청에 의하여 증명요청서의 DocContentInfoFlag 확장

필드에 설정된 값에 따라, 패키지 규격에 정의된 본제목, 키워드, 패키지 내용설명 필드의 정보를 설정하여야 하며, 이용자의 요청 없이 공인전자문서센터에서 임의로 생성하거나 임의의 내용을 기재할 수 없다.

```
DocContentInfo  ::= SEQUENCE {
    title                [0] EXPLICIT BMPString (SIZE (1..128)) OPTIONAL,
    keyword              [1] EXPLICIT BMPString (SIZE (1..35)) OPTIONAL,
    description          [2] EXPLICIT BMPString (SIZE (1..1000)) OPTIONAL }
```

title 필드는 본제목(MainTitle) 필드의 값을 BMPString 형식으로 변환하여 설정하도록 한다.

keyword 필드는 키워드(keyword) 필드의 값을 BMPString 형식으로 변환하여 설정하도록 한다.

description 필드는 패키지 내용설명(description) 필드의 값을 BMPString 형식으로 변환하여 설정하도록 한다.

5.3 에러 메시지

본 규격을 준용하지 않은 증명요청서를 수신하거나, 본 규격을 준용한 증명서를 생성하지 못하는 경우에 공인전자문서센터는 증명서 대신 에러 메시지를 생성하여 증명서 요청자에게 송신하여야 한다.

에러 정보를 포함하는 메시지는 ARCErrorNotice를 이용하여 증명서 요청자에게 전달된다.

```
ARCErrorNotice  ::=  SEQUENCE  {
    transactionStatus          PKIStatusInfo ,
    transactionIdentifier      GeneralName OPTIONAL      }
```

transactionIdentifier 필드는 사용하지 않는다.

PKIStatusInfo는 IETF RFC 4210에서 정의하고 있는 구조를 사용한다.

```
PKIStatusInfo ::= SEQUENCE {
        status          PKIStatus,
        statusString    PKIFreeText     OPTIONAL,
        failInfo        PKIFailureInfo  OPTIONAL      }

PKIStatus ::= INTEGER {
        accepted                        (0),
        grantedWithMods                 (1),
        rejection                       (2),
        waiting                         (3),
        revocationWarning               (4),
        revocationNotification          (5),
        keyUpdateWarning                (6)       }

PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
```

PKIStatus의 값은 항상 rejection만을 사용한다.

PKIFreeText에는 상세한 오류 메시지를 포함할 수 있다.

failInfo필드는 역시 RFC 4210에서 사용하는 PKIFailureInfo를 사용하며, 본 규격에서 사용하는 값들은 다음과 같다.

```
PKIFailureInfo ::= BIT STRING {
        badAlg              (0),
        badMessageCheck     (1),
        badRequest          (2),
        badTime             (3),
        badDataFormat       (5),
        wrongAuthority      (6),
```

incorrectData (7),
unacceptedPolicy (15),
unacceptedExtension (16),
badSenderNonce (18),
signerNotTrusted (20),
unsupportedVersion (22),
unAuthorized (23),
systemUnavail (24),
systemFailure (25) }

번호	메시지	내용
0	badAlg	정의되지 않았거나 지원되지 않는 알고리즘
1	badMessageCheck	요청메시지 무결성 손상 (전자서명 검증 실패 등)
2	badRequest	인가되지 않았거나 지원되지 않는 작업 요청
3	badTime	요청메시지의 시간이 시스템 시간과 다름 (정책상 허용범위 초과)
5	badDataFormat	잘못된 요청메시지 포맷
6	wrongAuthority	잘못된 기관(요청메시지의 기관명이 응답 메시지 생성 기관과 다름)
7	incorrectData	요청메시지의 내용이 잘못됨
15	unacceptedPolicy	요청된 정책 미 지원
16	unacceptedExtension	요청된 확장필드 미 지원
18	badSenderNonce	잘못된 송신자 nonce
20	signerNotTrusted	요청메시지 서명자의 신원확인 불가 또는 신뢰할 수 없음
22	unsupportedVersion	지원되지 않는 요청메시지 버전
23	notAuthorized	인가되지 않은 송신자
24	systemUnavail	현재 시스템 이용 불가
25	systemFailure	시스템에서 처리 중 실패

6. 증명서 검증

증명서 요청자(requester), 증명서 수임자(nominee), 그리고 증명서 수임자가 명시되어 있지 않은 증명서를 소유한 모든 이용자는 증명서 검증자(verifier)로서 증명서를 검증할 수 있다.

증명서 검증 과정은 크게 증명서 유효성 검증과 증명서 내용 검증으로 구분된다.

증명서 유효성 검증은 증명서로서의 효력을 가지기 위한 조건들의 만족 여부를 확인하는 과정이고, 증명서 내용 검증은 증명서를 활용하기 위한 조건들을 만족하는가를 확인하는 과정이다.

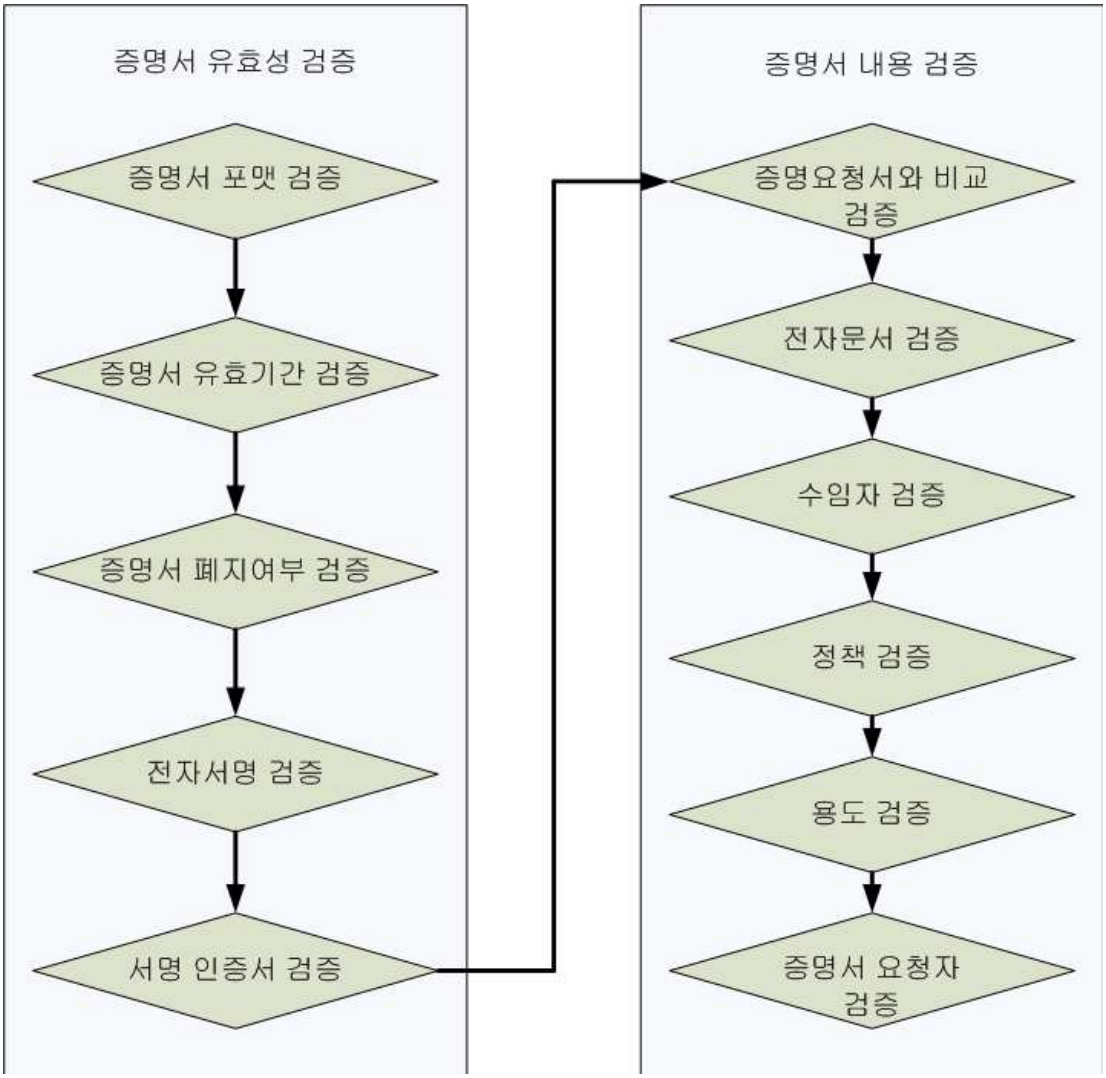


그림 4. 증명서 검증

6.1 증명서 유효성 검증

증명서 유효성 검증은 증명서로서의 효력을 가지기 위한 조건들의 만족 여부를 확인하는 과정으로서, 유효성 검증에 실패하면 증명서에 포함된 내용을 신뢰할 수 없게 되므로, 더 이상 증명서는 신뢰받는 기관이 발급한 보증서로의 역할을 수행할 수 없다.

증명서 유효성 검증은 증명서 포맷 검증, 증명서 유효기간 검증, 증명서 폐지여부 검증, 전자서명 검증, 서명 인증서 검증으로 구분되며, 한 단계에서 실패하면 다음 과정을 진행하지 않고 검증 실패로 처리한다.

6.1.1 증명서 포맷 검증

증명서 포맷 검증 단계에서는 검증 대상 증명서가 본 규격에서 제시한 증명서 포맷을 준수하여 생성되었는가의 여부를 확인한다.

본 규격에서 정의한 증명서의 구조 및 강제하거나 금하는 필드 생성 규칙의 준수 여부를 검증한 후, 증명서 내 각 필드 간의 모순이 없음을 확인한다.

필드 간 모순에 대한 주요한 몇 가지 예를 들면 다음과 같다.

- □ 증명서의 서명 인증서의 정보와 issuer 필드의 정보가 불일치
- □ 증명서 정책 OID와 target 필드의 구조 및 내용이 의미상 불일치
- □ requestInfo의 내용과 다른 필드의 내용의 불일치
- □ 등록증명서가 아닌데 nomineeRole 필드에 readDocument나 downloadDocument 값이 설정된 경우
- □ Qualifications 필드 내의 복수개의 qualification 필드들 중에 일부만이 onlyForNominee로 설정된 경우
- □ 최초등록증명서에 Qualifications 필드가 생성된 경우
- □ 등록증명서인데 CertifiedTime 필드가 생성되지 않은 경우
- □ 등록증명서가 아닌데 CertifiedTime 필드가 생성된 경우
- □ 시점확인증명서인데 Version 필드의 값이 2가 아닌 경우
- □ 시점확인증명서가 아닌데 Version 필드의 값이 1이 아닌 경우

이외에도 규격에서 정의한 필드 생성 규칙을 준수하지 않았거나, 각 필드의 내용

에 상호 모순이 있는 경우는 검증 실패로 처리하도록 한다.

검증 대상 증명서가 잘못된 포맷으로 생성되었다면, 이후의 과정을 더 진행하지 않고 검증 프로세스는 여기서 종료하도록 한다.

이용자 시스템은 공인전자문서센터로부터 발급받은 증명서의 유효성 여부를 검증하기 위하여 증명서를 발급받은 이후 이용자의 요청에 따라 증명서 포맷 검증을 수행하여야 한다.

6.1.2 증명서 유효기간 검증

증명서 유효기간이란 증명서 발급일(dateOfIssue)로부터 증명서 효력 만기일(dateOfExpiration)까지를 의미하며 본 단계는 증명서 검증 시점이 증명서 유효기간 내에 있음을 확인하는 단계이다.

증명서는 증명서의 유효기간 내에서만 유효성이 인정되므로 만약 검증 시점이 증명서 발급일보다 과거이거나 또는 증명서 효력 만기일 이후라면 증명서의 효력은 상실된다.

검증시점(T)과 증명서 발급일 및 증명서 효력 만기일을 다음과 같이 비교한다.

i) dateOfIssue ≤ T < dateOfExpiration 일 경우 : 검증 성공

ii) T < dateOfIssue 또는 dateOfExpiration ≤ T 일 경우 : 검증 실패

단, dateOfExpiration 필드에 NULL이 설정된 시점확인증명서에 대해서는 다음과 같이 비교한다.

i) dateOfIssue ≤ T : 검증 성공

ii) T < dateOfIssue : 검증 실패

즉, 효력 만기일이 명시되지 않은 시점확인증명서의 경우는 본 증명서 유효기간 검증 단계에서 증명서 발급일에 대해서만 검증하도록 한다. 이 경우, 해당 시점확인증명서의 효력기간은 해당 증명서에 서명한 전자서명 인증서의 유효기간과 동일한 것으로 보며, 이에 대한 검증은 다른 증명서와 마찬가지로 “6.1.5 서명 인증서 검증“ 단계에서 수행된다.

검증시점이 증명서의 유효기간 내에 있지 않으면 검증 실패로 처리하고 검증 프로세스를 종료한다.

이용자 시스템은 공인전자문서센터로부터 발급받은 증명서의 유효성 여부를 검증하기 위하여 증명서를 발급받은 이후 이용자의 요청에 따라 증명서 유효기간 검증을 수행하여야 한다.

6.1.3 증명서 폐지여부 검증

본 단계에서 증명서 검증자는 검증 대상 증명서에 대한 폐지여부를 공인전자문서센터에 요청하여 확인하게 된다.

신뢰기관으로서 공인전자문서센터가 발급한 증명서는 유효기간 동안 폐지되지 않는 것이 원칙이나, 중대한 사유로 증명서를 폐지해야만 되는 경우가 발생하였다면, 예외적으로 공인전자문서센터는 해당 증명서를 폐지한 후, 증명서 폐지 여부 확인 서비스를 통하여 해당 증명서의 폐지 정보를 증명서 검증자에게 제공하여야 한다.

증명서 검증자는 검증 대상 증명서를 공인전자문서센터에 송신하여 폐지여부에 대한 확인을 요청하고, 공인전자문서센터는 해당 증명서에 대한 폐지여부를 확인한 후, 응답메시지에 증명서의 폐지여부 및 폐지된 증명서인 경우에 폐지사유 등을 증명서 검증자에게 전송한다.

증명서 검증자는 “6.2.6. 증명서 요청자 검증”을 수행할 필요가 있다면, 폐지여부를 확인하는 절차 중에 증명서 요청자가 생성하여 공인전자문서센터에 보관 중인 난수값을 요청할 수 있으며, 이에 대한 응답메시지에는 폐지 여부에 대한 확인 결과와 함께 해당 난수가 포함되어 증명서 검증자에게 전달된다.

증명서 검증자가 난수를 공인전자문서센터로부터 획득하는 방법은 연계 인터페이스 규격의 ‘증명서 검증’ 항목을 참조하도록 한다.

증명서가 폐지되었다는 공인전자문서센터의 응답메시지를 수신했다면, 검증 실패로 처리하고 검증 프로세스를 종료한다.

6.1.4 전자서명 검증

공인전자문서센터가 발급하는 증명서는 무결성의 제공 및 부인방지를 위하여 전자서명을 포함하며, 전자서명 검증 단계에서 증명서에 포함된 전자서명을 검증한다.

전자서명의 검증은 일반적인 CMS의 signedData에 대한 전자서명 검증 방법을 따른다.

전자서명의 검증이 실패하였다면, 증명서의 무결성이 훼손되었음을 의미하며 더 이상 증명서의 내용을 신뢰할 수 없어 이후의 과정이 무의미해지므로 여기서 검증 프로세스를 종료하도록 한다.

이용자 시스템은 공인전자문서센터로부터 발급받은 증명서의 유효성 여부를 검증하기 위하여 증명서를 발급받은 이후 이용자의 요청에 따라 전자서명 검증을 수행하여야 한다.

6.1.5 서명 인증서 검증

서명 인증서 검증 단계는 증명서에 서명한 인증서가 공인전자문서센터의 인증서임을 확인하는 단계와 해당 인증서의 유효성을 검증하는 단계로 이루어진다. 단, 시점확인증명서의 경우 서명 인증서 검증은 예외로 한다.

증명서 검증자는 소유하고 있는 공인전자문서센터의 인증서 또는 인증서 식별정보와 증명서 서명 인증서를 비교하여 증명서 서명 인증서가 공인전자문서센터의 인증서임을 확인한다. 증명서 검증자가 공인전자문서센터의 인증서 또는 인증서 식별정보를 획득하는 과정은 본 규격에서는 다루지 않는다.

증명서 서명 인증서의 경로 구축 및 인증서의 유효성 검증은 공인인증체계의 “공인인증서 경로검증 기술규격 [KCAC.TS.CERTVAL]”을 준용하여 검증한다.

증명서 서명 인증서의 유효기간 만료로 인하여 유효성 검증에 실패한 경우, 증명서의 효력도 상실된다. 단, 증명서 서명 인증서의 유효기간이 만료되었지만, 증명서의 유효기간 검증에 성공하였고 전자서명 장기검증 기술이 적용된 상태라면, 해당 검증 기술에 따라 검증하여 성공하였을 경우, 서명 인증서의 유효기간 만료와는 관계없이 증명서의 유효성 검증에 성공한 것으로 처리한다.

전자서명 장기검증 기술은 본 버전의 규격에서는 다루지 않으며, 한국인터넷진흥원 또는 유관 기관이 제정한 기술규격을 준용하도록 한다.

이용자 시스템은 공인전자문서센터로부터 발급받은 증명서의 유효성 여부를 검증하기 위하여 증명서를 발급받은 이후 이용자의 요청에 따라 서명 인증서 검증을 수행하여야 한다.

6.2 증명서 내용 검증

증명서의 내용 검증 단계는 증명서의 활용 주체나 외부 환경 등에 따른 효용성을 검증하는 과정으로서 내용 검증에 실패하였다 하더라도 증명서 자체의 유효성을 상실하지는 않는다.

단, 증명요청서와 비교검증, 전자문서 검증에 실패하였다면, 검증 실패 메시지를 공인전자문서센터에 통보하여, 공인전자문서센터가 이에 대한 원인을 분석한 후 필요 시 증명서를 폐지 처리하여 증명서의 유효성을 상실하도록 한다. 검증 실패 메시지를 공인전자문서센터에 통보하는 방법은 본 규격에서는 다루지 않는다.

내용 검증에 실패한 증명서의 이용자는 해당 증명서를 활용할 수 없다.

증명서 내용 검증은 증명요청서와 비교검증, 전자문서 검증, 수임자 검증, 정책 검증, 용도 검증, 증명서 요청자 검증으로 구분되며, 한 단계에서 실패하면 다음 과정을 진행하지 않고 검증 실패로 처리한다.

각 검증의 순서는 필요에 따라 바뀔 수 있다.

6.2.1 증명요청서와 비교 검증

증명요청서와의 비교검증은 증명요청서를 생성한 증명서 요청자만이 수행할 수 있는 검증 단계로서, 증명서 요청자는 공인전자문서센터로부터 증명서를 발급받은 즉시 반드시 발급된 증명서가 공인전자문서센터에 송신했던 증명요청서에 대한 증명서인가의 여부를 확인하여야 하며, 추후에도 증명요청서가 존재한다면 본 검증을 수행할 수 있다.

증명서 요청자는 공인전자문서센터가 발급한 증명서의 증명서 요청 메시지 정보인 requestInfo 필드의 arcCertRequest가 자신이 생성하여 송신했던 증명요청서의 encapContentInfo 필드의 ARCCertRequest 구조체의 값과 동일함을 확인한다.

증명요청서와의 비교검증에 실패하였다면, 증명서 요청자는 공인전자문서센터에 증명요청서와의 비교검증 실패 사실을 통보하여 폐지 처리하도록 하고, 증명서를 다시 발급받아야 한다.

이용자 시스템은 증명서 발급요청에 의하여 발급받은 증명서의 내용 이상 여부를 검증하기 위하여 증명서를 발급받은 이후 이용자의 요청에 따라 증명요청서와 비교검증을 수행하여야 한다.

6.2.2 전자문서 검증

증명서 검증자가 증명서와 관련된 전자문서를 가지고 있다면, 반드시 증명서의 증명대상 필드에 포함된 전자문서의 해시값과의 비교 검증을 수행해야 한다.

이용자 시스템은 공인전자문서센터로부터 발급받은 원본증명서의 이상 여부를 검증하는 한편 발급된 원본 전자문서의 무결성을 검증하기 위하여, 원본증명서와 발급된 전자문서와의 비교검증을 수행하여야 한다.

변환본 전자문서 열람 시 불변경증명서가 발급되었다면, 마찬가지로 이용자 시스템은 불변경증명서의 이상 여부를 검증하는 한편 열람되는 변환본 전자문서의 무결성을 검증하기 위하여, 변환본 전자문서 및 불변경증명서를 수신한 이후 이용자의 요청에 따라 비교검증을 수행하여야 한다.

어떤 데이터가 특정 시각에 존재하였음을 증명하는 시점확인증명서의 경우에는 본 전자문서 검증 과정을 수행하지 않는다.

6.2.2.1 증적 증명에 대한 검증

증명서의 증명대상 필드로서 증적 증명을 의미하는 opRecord 필드에는 원본 전자문서의 해시값을 포함하는 orgDocInfo 필드와 발급된 전자문서의 해시값을 포함

하는 issuedDocInfo 필드가 있는데, orgDocInfo 필드는 증적 필드에 항상 포함되는 필드로서 공인전자문서센터가 보관 중인 원본 전자문서 전체의 해시값을 포함하고, issuedDocInfo 필드는 문서 발급 서비스를 포함되는 필드로서 발급된 전자문서의 해시값을 포함한다. 만약 발급된 전자문서가 원본 전자문서 전체에 대해서 발급되었다면, orgDocInfo 필드에 포함된 전자문서 해시값과 issuedDocInfo 필드에 포함된 전자문서 해시값은 동일해야 한다.

이용자가 증명서 검증자가 되어 orgDocInfo 필드 내의 전자문서의 해시값을 검증하기 위해서는 공인전자문서센터에 등록된 전자문서와 동일한 원본 전자문서가 필요하고, issuedDocInfo 필드 내의 전자문서의 해시값을 검증하기 위해서는 발급된 전자문서가 필요하다.

증명서 검증자는 전자문서 검증을 수행할 필요가 있다면, 해당 전자문서를 획득하여야 하며, 증명서 검증자가 해당 전자문서를 획득하는 과정은 본 규격에서는 다루지 않는다.

전자문서 검증을 수행하는 과정에서 주의하여야 할 사항은, 패키지에 다수의 전자문서파일들이 포함되어 있을 때 각 문서파일들을 해시하는 과정에서 공인전자문서센터가 수행한 과정과 증명서 검증자가 수행한 과정이 동일해야 한다는 것이다.

즉, 공인전자문서센터나 증명서 검증자는 패키지 내 다수의 전자문서파일에 대하여 패키지에 포함된 순서로 연접한 후 정해진 해시 알고리즘에 따라 한 번 해시하여 전자문서의 해시값을 생성하도록 한다. 만약, 추출된 전자문서파일들이 패키지 규격상 암호화된 상태라면 연접하기 전에 복호화를 먼저 수행하도록 한다.

만약 전자문서의 해시값과 증명대상 정보에 생성된 전자문서의 해시값이 다르다면, 증명서 검증자는 해당 사실을 공인전자문서센터에 통보하여 원인 확인 후 필요시 폐지처리 하도록 한다.

6.2.2.2 원본 및 불변경 증명에 대한 검증

증명서의 증명대상 필드로서 원본 증명 또는 불변경 증명을 의미하는 orgAndIssued 필드는 원본 전자문서 정보를 포함하는 orgDocInfo 필드와 발급 또는 열람되는 전자문서 정보를 포함하는 issuedDocInfo 필드로 구성된다. orgAndIssued 필드 내의 orgDocInfo 필드는 공인전자문서센터가 보관 중인 원본 전자문서의 전체의 해시값을 포함하고, issuedDocInfo 필드는 발급된 원본 전자문서 또는 열람되는 변환본 전자문서의 해시값을 포함한다.

원본증명서의 경우, 만약 발급된 전자문서가 원본 전자문서 전체에 대해서 발급되었다면 orgDocInfo 필드의 전자문서 해시값과 issuedDocInfo 필드의 전자문서 해시값은 동일하며, 일부 전자문서만 발급되었다면 orgDocInfo 필드의 전자문서 해시값과 issuedDocInfo 필드의 전자문서 해시값은 다르다.

변환본 열람 시 발급되는 불변경증명서의 issuedDocInfo 필드는 이용자에게 열람되는 변환본 전자문서의 해시값이므로 orgDocInfo 필드의 원본 전자문서 해시값과는 항상 다르다.

원본증명서는 원본증명서의 issuedDocInfo 필드에 설정된 원본 전자문서의 해시값과 발급된 전자문서를 해시한 값의 동일성 여부를 검증하여야 한다.

불변경증명서 역시 변환본 전자문서 열람 시 함께 발급되기 때문에, 불변경증명서의 issuedDocInfo 필드에 설정된 변환본 전자문서의 해시값과 이용자가 열람하는 변환본 전자문서를 해시한 값의 동일성 여부를 검증하여야 한다.

원본증명서 검증을 위한 원본 전자문서의 해시 절차는 증적 증명에서와 마찬가지 방법으로 생성하도록 한다. 불변경증명서 검증을 위하여 변환본 전자문서를 해시할 때는, 이용자에게 열람되는 문서들 중에서 변환되지 않은 원본 전자문서가 있다면 이를 제외한 변환본 전자문서들만을 순서대로 연접하여 해시한 후 비교 검증을 수행하여야 한다. 공인전자문서센터의 전자문서 열람 서비스 구현방식에 따라 다양한 열람 서비스가 제공될 수 있으며, 이용자 시스템은 이에 따른 불변경증명서와 변환본 전자문서와의 비교검증 기능을 제공하여야 한다.

만약 실제 전자문서를 해시한 값과 증명대상 정보에 생성된 전자문서의 해시값이 다르다면, 증명서 검증자는 해당 사실을 공인전자문서센터에 통보하여 원인 확인 후 필요 시 폐지처리 하도록 한다. 또한 검증에 실패한 원본증명서 또는 불변경증명서가 첨부된 전자문서에 대해서도 공인전자문서센터의 원인 확인 후 별도의 조치가 있을 때까지는 신뢰할 수 없는 것으로 판단하도록 한다.

6.2.3 수임자 검증

증명서에 수임자를 지정한 Qualifications 확장필드가 존재하고 critical의 값이 TRUE인 경우, 증명서 검증자는 Qualifications 필드의 값을 확인하여야 한다.

수임자 검증이란, 수임자의 정보가 올바르게 생성되었음을 검증하는 것이 아니라, 증명서 검증자 또는 수임자임을 주장하는 자가 정당한 수임자인가를 확인하는 절차이다.

증명서의 Qualifications 필드의 모든 Qualification 항목들에 대하여 nomineeRole 필드의 값이 onlyForNominee를 포함하고 있다면, 해당 증명서는 각 qualification 항목의 nomineeInfo에 지정된 증명서 수임자들을 위해서만 발급된 것이므로, 증명서 검증자 또는 수임자임을 주장하는 자가 정당한 증명서 수임자인 경우에만 증명서를 활용할 수 있다.

수임자 검증에 실패하였더라도, 증명서 검증자나 수임자임을 주장하는 자가 증명서를 활용하지 못할 뿐, 증명서의 유효성을 상실하지는 않는다.

만약 Qualifications 확장필드가 없거나, critical의 값이 FALSE이거나, 모든 qualification 필드 내 nomineeRole 필드의 값이 onlyForNominee를 포함하지 않는

다면, 해당 증명서는 모든 사람에 의해서 활용될 수 있다.

수임자 검증의 방법은 “부록 3.2. 수임자 검증”을 참조한다.

6.2.4 정책 검증

증명서 검증자는 검증하고자 하는 증명서에 포함된 증명서 정책이 현재 활용을 위한 목적에 부합되는가를 검증해야 한다.

증명서에 포함된 증명서의 정책은 증명서의 활용, 용도, 목적, 보증 범위 등에 대한 내용을 담고 있으며, 공인전자문서센터는 이에 대한 내용을 증명서를 사용하는 모든 이용자를 위하여 공지 하고 있다.

증명서 검증자는 증명서 정책 필드의 증명서 정책이 자신의 증명서 활용 목적을 위해서 적합한 정책인가를 확인한다.

정책 검증에 실패하였더라도, 증명서 검증자의 활용 목적을 위하여 증명서를 활용하지 못할 뿐, 증명서의 유효성을 상실하지는 않는다.

6.2.5 용도 검증

증명서에 증명서 용도를 지정한 UsageType 확장필드가 존재하고 critical의 값이 TRUE인 경우, 증명서 검증자는 UsageType 필드의 값을 확인하여야 한다.

증명서의 사용환경과 증명서 용도 필드의 값이 일치하지 않으면 검증에 실패하게 된다.

본 버전의 규격에서는 UsageType 필드의 critical의 값을 FALSE로 설정하도록 하므로 증명서의 용도 검증을 하지 않아도 된다.

6.2.6 증명서 요청자 검증

증명서에 증명서 요청자 정보가 포함되어 있는 경우, 즉 requestInfo 필드의 정보로 null이 아닌 arcCertRequest 필드가 사용되었다면, arcCertRequest 필드 내의 requester 필드에 대하여 필요한 경우에 검증을 수행할 수 있다.

본 증명서 요청자 검증 과정은 증명서 요청자라고 주장하는 자의 신원정보와 증명서 내의 증명서 요청자 정보와의 동일성 여부에 대한 확인을 주목적으로 한다.

증명서 요청자 검증의 방법은 “6.1.3 증명서 폐지여부 검증” 및 “부록 2.2. IdentifyData 구조체의 검증”의 절차를 참조한다.

증명서 요청자 검증에 실패하였더라도, 증명서 요청자라고 주장하는 자가 정당한 증명서 요청자가 아님을 확인한 것일 뿐, 증명서의 유효성을 상실하지는 않는다.

7. 증명서 출력

다음은 증명서의 전자 및 종이 출력에 대한 정의로써 증명서의 출력 기능을 제공하는 공인전자문서센터의 경우 증명서의 디자인을 제외한 증명서의 내용을 준수하여야 한다.

7.1. 종이 증명서 포맷

종이 증명서는 크게 구분하여, 증명서 종류, 증명서 내용, 공인전자문서센터 로고 이미지, 온라인 확인 안내문의 네 부분으로 구성된다.

[종이증명서 포맷]

7.1.1. 종이 증명서 종류

증명서의 가장 상위에는 증명서의 종류가 기술되어져야 하는데, 이는 전자파일 증명서의 일반정보 탭을 선택하였을 때 나타나는 정보 중 증명서 종류를 그대로 사용하도록 한다.

7.1.2. 종이 증명서 내용

종이 증명서의 내용 부분은 증명서 일반정보, 증명 대상, 수임자 정보의 세 부분으로 구성되어 있으며, 증명서 내용의 대부분을 포함하고 있다.

종이 증명서 일반정보에는 증명서의 기본필드 중 증명대상 필드를 제외한 전부와 확장필드 중의 증명서용도 필드를 포함한다.

1. 증명서 요청자와 발급자 항목은 한글실명만을 기술한다.
2. 증명서 정책 항목은 증명서 정책 필드의 내용 중 정책 ID와 정책이 게시된 URL만을 기술하도록 한다.
3. 최초등록증명서에서 증명서 요청자 항목과 증명서 요청시간 항목은 '-'로 표시한다.
4. 전자문서 보관 보증일시 항목은 등록증명서에서만 생성한다.

증명 대상에는 증명서의 기본필드 중 증명 대상 필드를 상세히 기술한다.

1. 종류에는 증명 대상이 증적 정보인지, 원본 및 발급본 정보인지, 데이터 해시 정보인지를 기술하도록 한다.
2. 원본문서 정보와 발급문서 정보에는 패키지 ID와 문서 ID까지만 기술하도록 한다.

수임자 정보는 증명서에 권한부여 필드가 있을 경우에만 출력되어야 함에 주의한다. 즉, 실제 증명서에 권한부여 필드가 생성되지 않았다면, 종이 증명서에서도 수임자 정보를 출력하지 않도록 한다.

수임자 정보는 수임자 실명, 수임자 인증서 정보, 수임자 권한으로 구성되어 있으며, 수임자 실명과 수임자 인증서 정보 중에 실제 증명서에 생성되지 않은 항목은 '-'로 표시한다.

증명서에 복수의 수임자 정보가 포함되어 있다면, 포함된 수임자 정보를 모두 기술해 주도록 하며, 한 페이지를 초과할 경우는 다음 페이지에 이어서 출력하도록 한다.

증명서가 두 페이지 이상일 경우에 주의할 점은 다음과 같다.

1. 온라인 확인 안내문은 모든 페이지에 출력하도록 한다.
2. 페이지의 오른쪽 하단에 발급자 및 증명서의 일련번호와 함께 현재페이지/전체페이지를 기술하도록 한다.

종이증명서의 각 항목 값을 기술할 때 주의할 점은 다음과 같다.

1. 확장필드이거나 선택적 생성필드로서 실제 증명서에 생성되지 않은 항목인 경우, 종이 증명서 포맷의 해당 항목을 '-'로 표시한다. 단, 수임자 정보는 실제 증명서에 권한부여 필드가 생성되지 않았다면 종이증명서에서 항목자체를 생략한다.

2. 실제 증명서에 "NULL" 형식으로 생성된 필드인 경우에도, 종이 증명서 포맷의 해당 항목은 '-'로 표시한다.

3. 특정한 의미를 지닌 숫자나 비트값에 대하여, 해당 값의 의미를 한글로 표현하도록 하며 각 항목에서 사용할 용어는 다음과 같다.

   가. 증명서 용도 : 온라인, 모바일, 종이출력

   나. 사유 : 이용자 요청, 공인전자문서센터 내부업무, 보존기간 만료

   다. 수임자 권한 : 정당한 수신자, 문서 열람, 문서 발급

6. 기타 항목들의 표현 방법은 각 증명서에 대한 종이증명서 화면을 참고하도록 한다.

7.1.3. 한국인터넷진흥원 로고 이미지

종이 증명서에는 한국인터넷진흥원 로고를 사용할 수 있다. 한국인터넷진흥원 로고 이미지는 한국인터넷진흥원 홈페이지에서 받을 수 있는 CI 이미지를 활용하여 공인전자문서센터로 지정된 사업자가 자유롭게 선택하여 디자인할 수 있다. 다만, 제공된 CI 이미지를 임의로 수정해선 안 되며, CI의 활용 사실을 한국인터넷진흥원에 알려야 한다.

7.1.4. 온라인 확인 안내문

종이 증명서에는 증명서의 내용을 온라인으로 확인할 수 있다는 안내문과 함께 실제 증명서의 내용을 확인할 수 있는 공인전자문서센터 웹페이지의 URL이나 QR 코드 등이 들어가야 한다.

온라인 확인 안내문과 함께, 부가적으로 종이 증명서에 2D 바코드 등의 위·변조 방지 기술을 적용할 수 있으며, 만약 출력 페이지가 두 페이지 이상인 경우에는 매 페이지마다 적용하도록 한다.

추가로 종이 증명서에 복사방지 코드를 적용할 수 있으며, 이용자에게 증명서 내용의 전달을 방해하지 않는 범위 내에서 각 공인전자문서센터가 지원하는 기술을 적용하여 구현하도록 한다.

7.1.5. 원본증명서와 대상문서의 묶음 출력

보관 중인 전자문서와 원본증명서를 제출용 등으로 묶음으로 출력할 경우 원본증명서가 해당 전자문서에 대한 증명서인지 쉽고 명확하게 확인할 수 있도록 증명서와 대상문서의 모든 페이지에 동일한 증빙 정보(증명서 일련번호 등)를 출력하여야 한다. 그리고 출력한 문서에 페이지를 임의로 추가/제거할 수 없도록 증명서 첫 페이지부터 대상문서의 마지막 페이지까지 순차적으로 증가하는 쪽 번호를 출력해야 한다. 원본증명서와 첨부된 전자문서의 신뢰성을 높이기 위하여 공인전자문서센터 지정 사실

과 명칭 사용, 증명서 효력 등의 법령을 증명서 또는 전자문서에 표기할 수 있다.

최초등록증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자		-
	증명서 요청시간		-
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2007년 4월 5일 목요일 오후 4:45:28
	증명서 효력만기일		2008년 4월 5일 토요일 오후 4:45:28
	증명서 정책	정책 ID	1.2.410.200032.2.9.1
		정책 URL	http://www.koreda.or.kr/ceda/cps.htm

증명대상	종류		증적 정보
	서비스 증적 식별번호		01ab
	서비스 요청자	실명	김길동
		식별정보	[redacted]
	서비스 요청시간		2007년 4월 2일 월요일 오전 9:30:21
	서비스 응답시간		2007년 4월 2일 월요일 오전 9:30:40
	서비스 종류		전자문서 등록
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a456dbcd1a32955decab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
	사유		이용자 요청

본 증명서는 한국공인전자문서센터에서 발급한 최초등록증명서입니다.

한국공인전자문서센터

[redacted]

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.koreda.or.kr/ceda/service/certificate/verify.htm)

[종이증명서-최초등록증명서]

등록증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	[illegible]
	증명서 요청시간		2007년 4월 5일 목요일 오후 4:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2007년 4월 5일 목요일 오후 4:45:29
	증명서 효력만기일		2008년 4월 5일 토요일 오후 4:45:29
	증명서 정책	정책 ID	1.2.410.200032.2.9.2
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		온라인, 종이출력
	전자문서 보관 보증일시		2007년 4월 5일 목요일 오후 4:45:29
	증명서 용도		은행제출용

증명대상	종류		증적 정보
	서비스 증적 식별번호		01ab
	서비스 요청자	실명	김길동
		식별정보	[illegible]
	서비스 요청시간		2007년 4월 2일 월요일 오전 9:30:21
	서비스 응답시간		2007년 4월 2일 월요일 오전 9:30:40
	서비스 종류		전자문서 등록
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955decab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		문서 제목	패키지 메타데이터 설명서
		키워드	패키지, 메타데이터, 설명서
		문서 설명	공인전자문서센터에 관리되는 전자문서 패키지 마다 필요한 메타데이터 요소에 대해 정의하는...
	사유		이용자 요청

수입자 정보 1	수입자 실명		(주)신라은행
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8a98
	수입자 권한		정당한 수신자, 문서 열람

본 증명서는 한국공인전자문서센터에서 발급한 등록증명서입니다.

한국공인전자문서센터

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

[종이증명서-등록증명서]

발급증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	55ab2980aee15fa58850daabe955ab2960aee15fa58850daabe5
	증명서 요청시간		2007년 4월 5일 목요일 오후 4:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2007년 4월 5일 목요일 오후 4:45:28
	증명서 효력만기일		2008년 4월 5일 토요일 오후 4:45:28
	증명서 정책	정책 ID	1.2.410.200032.2.9.3
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		온라인, 모바일, 종이출력
	증명서 용도		은행계좌용

증명대상	종류		증적 정보
	서비스 증적 식별번호		01ab
	서비스 요청자	실명	김길동
		식별정보	2d3d3834bcf6de56f49e2c36bce32d3d3834bcf6de56f49e2c36bce8
	서비스 요청시간		2007년 4월 2일 월요일 오전 9:30:21
	서비스 응답시간		2007년 4월 2일 월요일 오전 9:30:40
	서비스 종류		전자문서 발급
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955deca1b2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		문서 제목	패키지 메타데이터 설명서
		키워드	패키지, 메타데이터, 설명서
		문서 설명	공인전자문서센터에 관리되는 전자문서 패키지마다 필요한 메타데이터 요소에 대해 정의하는...
	발급문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955deca1b2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
	사유		이용자 요청

수임자 정보 1	수임자 실명		(주)신라은행
	수임자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8a98
	수임자 권한		정당한 수신자

본 증명서는 한국공인전자문서센터에서 발급한 발급증명서입니다.

한국공인전자문서센터

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

[종이증명서-발급증명서]

이관증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	56abc2960acee15cfa598b50dcaabe56abc2960acee15cfa598b50dcaabe5
	증명서 요청시간		2007년 4월 5일 목요일 오후 4:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2007년 4월 5일 목요일 오후 4:45:28
	증명서 효력만기일		2008년 4월 5일 토요일 오후 4:45:28
	증명서 정책	정책 ID	1.2.410.200032.2.9.4
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		온라인, 모바일, 종이출력
	증명서 용도		은행제출용

증명대상	종류		증적 정보
	서비스 증적 식별번호		01ab
	서비스 요청자	실명	김길동
		식별정보	2a3b3634bc56dae56f6b9e2bc36baca32a3b3634bc5fdea56f6b9e2bc36baca3
	서비스 요청시간		2007년 4월 2일 월요일 오전 9:30:21
	서비스 응답시간		2007년 4월 2일 월요일 오전 9:30:40
	서비스 종류		전자문서 이관
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955decab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		문서 제목	패키지 메타데이터 설명서
		키워드	패키지, 메타데이터, 설명서
		문서 설명	공인전자문서센터에 관리되는 전자문서 패키지마다 필요한 메타데이터 요소에 대해 정의하는...
	이·수관 상대 보관소	보관소명	한국문서보관소
		문서정보	1.2.410.200032.1.10.1bbbc557aef3947acde5
	사유		이용자 요청

수입자 정보 1	수입자 실명		(주)신라은행
	수입자 인증서 정보	발급자	-
		일련번호	-
	수입자 권한		정당한 수신자

본 증명서는 한국공인전자문서센터에서 발급한 이관증명서입니다

한국공인전자문서센터

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

[종이증명서-이관증명서]

폐기증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	55abc2960cce15fa59830dcabe355abc2960cce15fa59830dcabe5
	증명서 요청시간		2007년 4월 5일 목요일 오후 4:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2007년 4월 5일 목요일 오후 4:45:28
	증명서 효력만기일		2008년 4월 5일 토요일 오후 4:45:28
	증명서 정책	정책 ID	1.2.410.200032.2.9.5
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		온라인, 모바일, 종이출력
	증명서 용도		은행계좌용

증명대상	종류		증적 정보
	서비스 증적 식별번호		01ab
	서비스 요청자		-
	서비스 요청시간		2007년 4월 2일 월요일 오전 9:30:38
	서비스 응답시간		2007년 4월 2일 월요일 오전 9:30:40
	서비스 종류		전자문서 폐기
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955decab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		문서 제목	패키지 메타데이터 설명서
		키워드	패키지, 메타데이터, 설명서
		문서 설명	공인전자문서센터에 관리되는 전자문서 패키지마다 필요한 메타데이터 요소에 대해 정의하는...
	사유		보관기간 만료

수임자 정보 1	수임자 실명		-
	수임자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8a98
	수임자 권한		정당한 수신자

본 증명서는 한국공인전자문서센터에서 발급한 폐기증명서입니다.

한국공인전자문서센터

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

[종이증명서-폐기증명서]

원본증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	56abc25960acee15cfa59850daabe56abc25960acee15cfa59850daabe5
	증명서 요청시간		2007년 4월 5일 목요일 오후 4:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2007년 4월 5일 목요일 오후 4:45:28
	증명서 효력만기일		2008년 4월 5일 토요일 오후 4:45:28
	증명서 정책	정책 ID	1.2.410.200032.2.9.6
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		온라인, 모바일, 종이출력
	증명서 용도		은행제출용

증명대상	종류		원본 및 발급본 정보
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955dacab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		문서 제목	패키지 메타데이터 설명서
		키워드	패키지, 메타데이터, 설명서
		문서 설명	공인전자문서센터에 관리되는 전자문서 패키지마다 필요한 메타데이터 요소에 대해 정의하는...
	발급문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955dacab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
	원본발급여부		원본

수입자 정보 1	수입자 실명		(주)신라은행
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8a98
	수입자 권한		정당한 수신자, 문서 열람

본 증명서는 한국공인전자문서센터에서 발급한 원본증명서입니다.

한국공인전자문서센터

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

[종이증명서-원본증명서]

시점확인증명서

증명서 일반정보	증명서 버전		2
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	55abc2980acce15fa59830dcabe355abc2960acce15fa59830dcabe5
	증명서 요청시간		2007년 4월 5일 목요일 오후 4:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2007년 4월 5일 목요일 오후 4:45:29
	증명서 효력만기일		2008년 4월 5일 토요일 오후 4:45:29
	증명서 정책	정책 ID	1.2.410.200032.2.9.7
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		온라인, 종이출력
	증명서 용도		은행계좌용

증명대상	종류	데이터 해쉬 정보
	해쉬 알고리즘	sha256
	데이터 해쉬값	dd5b6eff390aefae3ef590872a363acd5c252233dd5b6eff390aefae3ef59087

수임자 정보 1	수임자 실명		(주)신라은행
	수임자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8a98
	수임자 권한		정당한 수신자

본 증명서는 한국공인전자문서센터에서 발급한 시점확인증명서입니다.

한국공인전자문서센터

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

[종이증명서-시점확인증명서]

등록증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	55abc25980acee15cfa598850dcaabe55abc25980acee15cfa598850dcaabe5
	증명서 요청시간		2007년 4월 5일 목요일 오후 4:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2007년 4월 5일 목요일 오후 4:45:29
	증명서 효력만기일		2008년 4월 5일 토요일 오후 4:45:29
	증명서 정책	정책 ID	1.2.410.200032.2.9.2
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		온라인, 종이출력
	전자문서 보관 보증일시		2007년 4월 5일 목요일 오후 4:45:29
	증명서 용도		은행제출용

증명대상	종류		중계 정보
	서비스 중계 식별번호		01ab
	서비스 요청자	실명	김길동
		식별정보	2a3b85234bc5fdae56b9e2bc35bacd32a3b85234bc5fdae56b9e2bc35bacd3
	서비스 요청시간		2007년 4월 2일 월요일 오전 9:30:21
	서비스 응답시간		2007년 4월 2일 월요일 오전 9:30:40
	서비스 종류		전자문서 등록
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955decab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		문서 제목	패키지 메타데이터 설명서
		키워드	패키지, 메타데이터, 설명서
		문서 설명	공인전자문서센터에 관리되는 전자문서 패키지마다 필요한 메타데이터 요소에 대해 정의하는...
	사유		이용자 요청

수입자 정보 1	수입자 실명		(주)신라은행
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8a98
	수입자 권한		정당한 수신자, 문서 열람
수입자 정보 2	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bf9
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

한국공인전자문서센터-0a45c3-1/3

[종이증명서-복수페이지1]

수입자 정보 3	수입자 실명		(주)한양은행
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8aaa
	수입자 권한		정당한 수신자, 문서 열람
수입자 정보 4	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bda
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급
수입자 정보 5	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bdd
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급
수입자 정보 6	수입자 실명		(주)신라은행
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8ac5
	수입자 권한		정당한 수신자, 문서 열람
수입자 정보 7	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bcd
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급
수입자 정보 8	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bce
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급
수입자 정보 9	수입자 실명		(주)한양은행
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8abb
	수입자 권한		정당한 수신자, 문서 열람
수입자 정보 10	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bba
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.koreda.or.kr/coda/service/certificate/verify.htm)

한국공인전자문서센터-0a45c3-2/3

[종이증명서-복수페이지2]

수입자 정보 11	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bef
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급
수입자 정보 12	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bfe
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급
수입자 정보 13	수입자 실명		(주)고려은행
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c2f8aff
	수입자 권한		정당한 수신자, 문서 열람
수입자 정보 14	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bea
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급
수입자 정보 15	수입자 실명		-
	수입자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0bae7bae
	수입자 권한		정당한 수신자, 문서 열람, 문서 발급

본 증명서는 한국공인전자문서센터에서 발급한 등록증명서입니다.

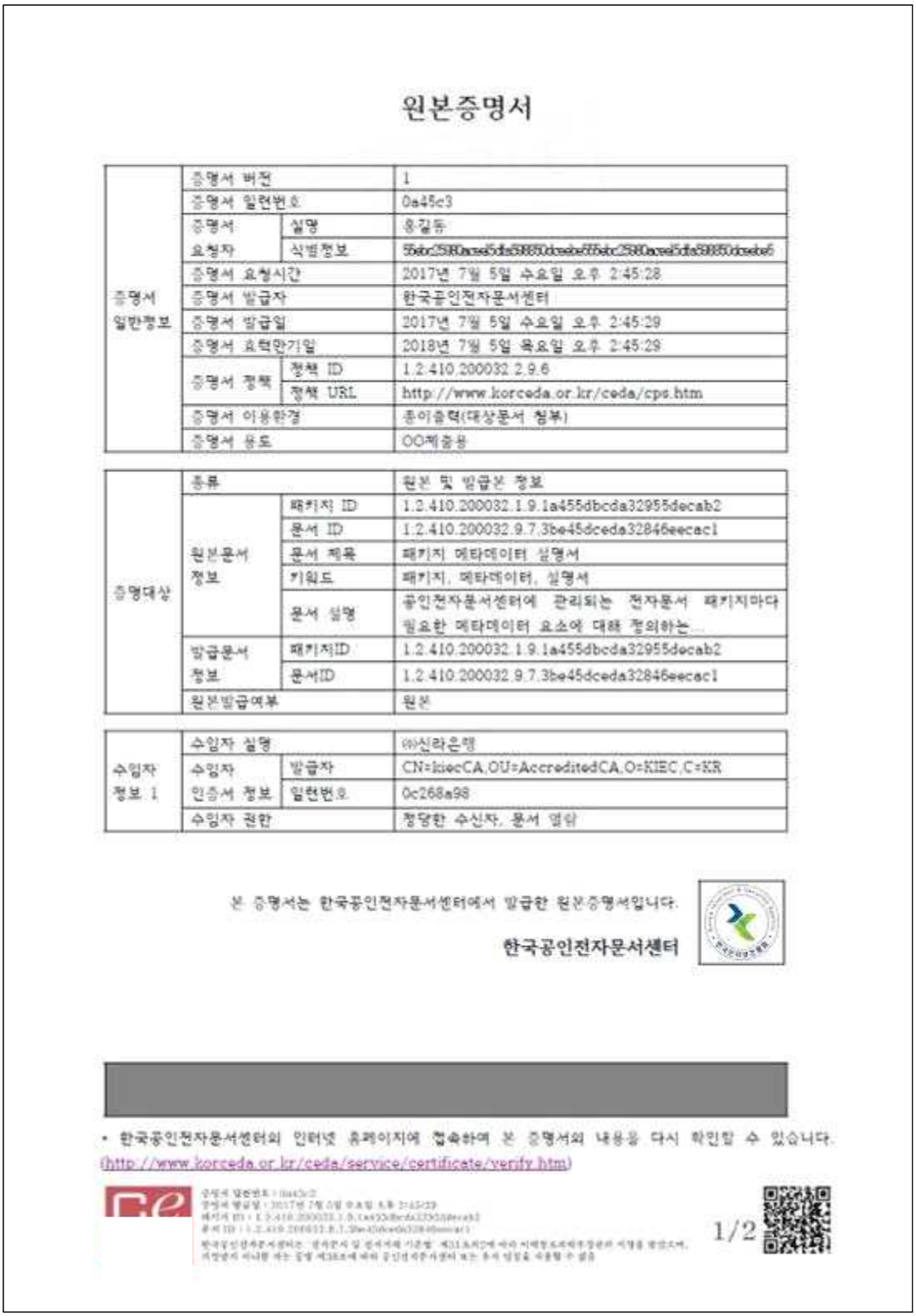
한국공인전자문서센터

[redacted]

* 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다. (http://www.koreda.or.kr/ceda/service/certificate/verify.htm)

한국공인전자문서센터-0a45c3-3/3

[종이증명서-복수페이지3]



원본증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	[redacted]
	증명서 요청시간		2017년 7월 5일 수요일 오후 2:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2017년 7월 5일 수요일 오후 2:45:29
	증명서 효력만기일		2018년 7월 5일 목요일 오후 2:45:29
	증명서 정책	정책 ID	1.2.410.200032.2.9.6
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		종이출력(대상문서 첨부)
	증명서 용도		OO제출용

증명대상	종류		원본 및 발급본 정보
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955decab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		문서 제목	패키지 메타데이터 설명서
		키워드	패키지, 메타데이터, 설명서
		문서 설명	공인전자문서센터에 관리되는 전자문서 패키지마다 필요한 메타데이터 요소에 대해 정의하는...
	발급문서 정보	패키지ID	1.2.410.200032.1.9.1a455dbcda32955decab2
		문서ID	1.2.410.200032.9.7.3be45dceda32846eecac1
	원본발급여부		원본

수임자 정보 1	수임자 실명		㈜신라은행
	수임자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c268a98
	수임자 권한		정당한 수신자, 문서 열람

본 증명서는 한국공인전자문서센터에서 발급한 원본증명서입니다.

한국공인전자문서센터

[redacted]

• 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있습니다.
(http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

1/2

[종이증명서-대상문서 묶음 출력 - 증명서]
(쪽번호, 진위확인코드, 기타 증빙정보의 내용 및 출력 위치는
공인전자문서센터가 임의로 정할 수 있음)

공인전자문서센터 지정기준 중 인력 · 기술능력

[시행 2015.10.16] [미래창조과학부고시 제2015-77호, 2015.10.16, 일부개정]

미래창조과학부(인터넷제도혁신과), 02-2110-2868

1. 정보통신기사 · 정보처리기사 및 전자계산기조직응용기사 이상의 국가기술자격과 동등한 자격

① 전자 · 통신관련학과, 정보처리기술관련학과, 암호 · 정보보호기술관련학과, 기록관리관련학과의 4년제 대학졸업자 또는 이와 동등 이상의 자격이 있다고 인정되는 자로서 동일 직무분야에서 3년 이상 실무에 종사한 자

② 정보통신 · 정보처리 및 전자계산기조직응용 분야의 산업기사로서 2년 이상 실무에 종사한 자

③ 정보통신 · 정보처리 및 전자계산기조직응용 분야의 기능사로서 5년 이상 실무에 종사한 자

④ 정보통신 · 정보처리 및 전자계산기조직응용 분야의 기사수준에 해당하는 교육훈련을 실시하는 기관에서 노동부령이 정하는 교육훈련기관의 기술훈련과정을 이수한 자로서 전자 · 통신관련, 정보처리기술관련, 암호 · 정보보호기술관련, 기록관리관련 직무분야에서 3년 이상 실무에 종사한 자

⑤ 전자 · 통신관련학과, 정보처리기술관련학과, 암호 · 정보보호기술관련학과, 기록관리관련학과의 전문대학 졸업자 또는 이와 동등 이상의 학력이 있다고 인정되는 자로서 동일 직무분야에서 5년 이상 실무에 종사한 자

⑥ 정보통신 · 정보처리 및 전자계산기조직응용 분야의 기능사수준에 해당하는 교육훈련을 실시하는 기관에서 노동부령이 정하는 교육훈련기관의 기술훈련과정을 이수한 자로서 전자 · 통신관련, 정보처리기술관련, 암호 · 정보보호기술관련, 기록관리관련 직무분야에서 5년 이상 실무에 종사한 자

2. 경력 분야

① 정보처리 · 관리 분야

가. 디지털 콘텐트 기술 분야

나. 전자문서관리 기술 분야

다. 기록관리 기술 분야

② 정보보호 운영 · 관리 분야

가. 정보보호기술 개발 분야

나. 정보보호시스템 개발 및 운영 · 관리 분야

③ 정보통신 운영 · 관리 분야

가. 정보통신기술 개발 분야

나. 정보통신시스템 개발 및 운영 · 관리 분야

다. 정보통신망 구축 및 운영 · 관리 분야

3. 행정사항

① (재검토기한) 미래창조과학부 장관은 「행정규제기본법」 및 「훈령?예규 등의 발령 및 관리에 관한 규정」에 따라 이 고시에 대하여 2016년 1월 1일 기준으로 매3년이 되는 시점(매 3년째의 12월 31일까지를 말한다)마다 그 타당성을 검토하여 개선 등의 조치를 하여야 한다.

부칙 <제2015-77호, 2015.10.16>

이 고시는 공포한 날부터 시행한다.

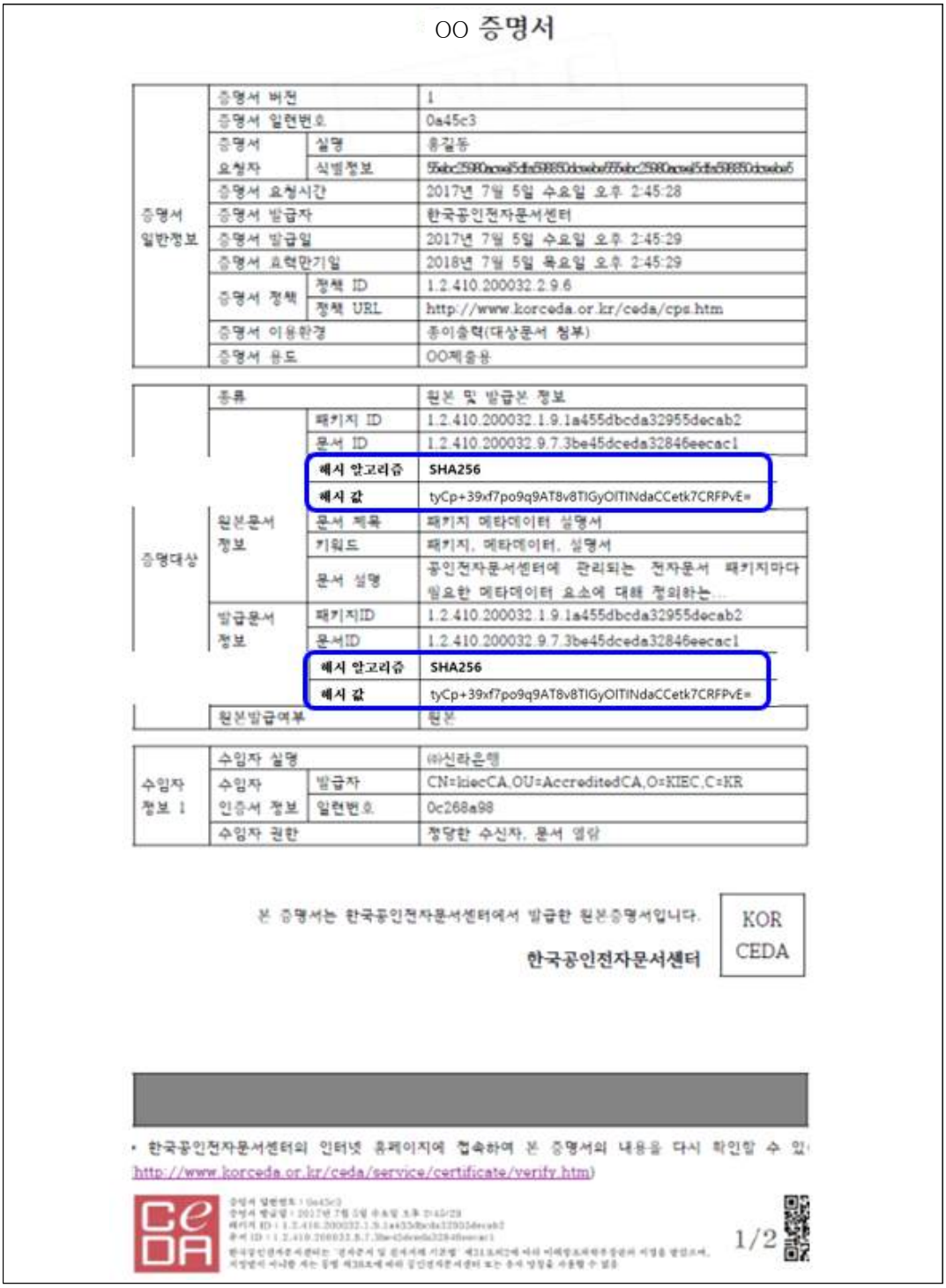
증명서 일련번호 : 0a45c3
증명서 발급일 : 2017년 7월 5일 수요일 오후 2:45:29

2/2

[종이증명서-대상문서 묶음 출력 – 대상문서(예)]
(증빙정보의 내용 및 출력 위치는 공인전자문서센터가 임의로 정할 수 있음)

7.2. PDF 증명서 포맷

PDF 증명서는 증명서 종류, 증명서 내용, 공인전자문서센터 로고 이미지, 온라인 확인 안내문의 네 부분으로 구성된다.



OO 증명서

증명서 일반정보	증명서 버전		1
	증명서 일련번호		0a45c3
	증명서 요청자	실명	홍길동
		식별정보	9abc2580acea5da68850dceabe7[illegible]
	증명서 요청시간		2017년 7월 5일 수요일 오후 2:45:28
	증명서 발급자		한국공인전자문서센터
	증명서 발급일		2017년 7월 5일 수요일 오후 2:45:29
	증명서 효력만기일		2018년 7월 5일 목요일 오후 2:45:29
	증명서 정책	정책 ID	1.2.410.200032.2.9.6
		정책 URL	http://www.korceda.or.kr/ceda/cps.htm
	증명서 이용환경		종이출력(대상문서 첨부)
	증명서 용도		OO제출용

증명대상	종류		원본 및 발급본 정보
	원본문서 정보	패키지 ID	1.2.410.200032.1.9.1a455dbcda32955decab2
		문서 ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		해시 알고리즘	SHA256
		해시 값	tyCp+39xf7po9q9AT8v8TlGyOlTINdaCCetk7CRFPvE=
		문서 제목	패키지 메타데이터 설명서
		키워드	패키지, 메타데이터, 설명서
		문서 설명	공인전자문서센터에 관리되는 전자문서 패키지마다 필요한 메타데이터 요소에 대해 정의하는...
	발급문서 정보	패키지ID	1.2.410.200032.1.9.1a455dbcda32955decab2
		문서ID	1.2.410.200032.9.7.3be45dceda32846eecac1
		해시 알고리즘	SHA256
		해시 값	tyCp+39xf7po9q9AT8v8TlGyOlTINdaCCetk7CRFPvE=
	원본발급여부		원본

수임자 정보 1	수임자 실명		㈜신라은행
	수임자 인증서 정보	발급자	CN=kiecCA,OU=AccreditedCA,O=KIEC,C=KR
		일련번호	0c268a98
	수임자 권한		정당한 수신자, 문서 열람

본 증명서는 한국공인전자문서센터에서 발급한 원본증명서입니다.

KOR CEDA

한국공인전자문서센터

[illegible]

• 한국공인전자문서센터의 인터넷 홈페이지에 접속하여 본 증명서의 내용을 다시 확인할 수 있
http://www.korceda.or.kr/ceda/service/certificate/verify.htm)

1/2

[PDF 증명서 예시]
(쪽번호, 진위확인코드, 기타 증빙정보의 내용 및 출력 위치는 공인전자문서센터가 임의로 정할 수 있음)

7.2.1. PDF 증명서 종류

증명서의 가장 상위에는 증명서의 종류가 기술돼야 하는데, 이는 전자파일 증명서의 일반정보 탭을 선택하였을 때 나타나는 정보 중 증명서 종류를 그대로 사용하도록 한다.

7.2.2. PDF 증명서 내용

PDF 증명서는 기본적으로 종이 증명서의 구성과 항목을 따른다. 증명서 일반정보, 증명 대상, 수임자 정보의 세 부분으로 구성되어 있으며, 증명서 내용의 대부분을 포함하고 있다. 그리고 PDF 증명서에 발급된 전자문서의 해시값을 포함해야 하며 해시값이 포함된 PDF 증명서를 전자서명 해야 한다.

PDF 증명서 일반정보에는 증명서의 기본필드 중 증명대상 필드를 제외한 전부와 확장필드 중의 증명서 용도 필드를 포함한다.

1. 증명서 요청자와 발급자 항목은 한글실명만을 기술한다.
2. 증명서 정책 항목은 증명서 정책 필드의 내용 중 정책 ID와 정책이 게시된 URL만을 기술하도록 한다.
3. 최초등록증명서에서 증명서 요청자 항목과 증명서 요청시간 항목은 '-'로 표시한다.
4. 전자문서 보관 보증일시 항목은 등록증명서에서만 생성한다.

증명 대상에는 증명서의 기본필드 중 증명 대상 필드를 상세히 기술한다.

1. 종류에는 증명 대상이 증적 정보인지, 원본 및 발급본 정보인지, 데이터 해시 정보인지를 기술하도록 한다.
2. 원본문서 정보와 발급문서 정보에는 패키지 ID와 문서 ID를 기술하고 해당 전자문서에 대한 해시값과 해시 알고리즘을 기술하도록 한다.
3. 발급 전자문서가 변환본인 경우 발급문서 정보에 변환본의 해시값과 해시 알고리즘을 기술하도록 한다.

수임자 정보는 증명서에 권한 부여 필드가 있을 경우에만 출력되어야 함에 주의한다. 즉, 실제 증명서에 권한 부여 필드가 생성되지 않았다면, 종이 증명서에서도 수임자정보를 출력하지 않도록 한다.

수임자 정보는 수임자 실명, 수임자 인증서 정보, 수임자 권한으로 구성되어 있으며, 수임자 실명과 수임자 인증서 정보 중에 실제 증명서에 생성되지 않은 항목은 '-'로 표시한다.

증명서에 복수의 수임자 정보가 포함되어 있다면, 포함된 수임자 정보를 모두 기술해 주도록 하며, 한 페이지를 초과할 경우는 다음 페이지에 이어서 출력하도록 한다.

발급된 전자문서의 해시값은 PDF 증명서 예시 그림과 같이 이용자가 볼 수 있는 영역에 기록하고 추가로 이용자가 볼 수 없는 영역에 기록하여야 한다. 이는, PDF 증명서와 발급된 전자문서의 무결성 검증을 위해서이다. 이렇게 작성된 PDF 증명서를 PDF의 표준 전자서명 방식으로 전자서명 한다.

1. PDF의 Catalog Dictionary 영역에 사용자 영역(CEDCProof)을 정의하고 값을 쓴다.

번호	필드명	TYPE	생성	처리	비고
1	CEDCProof	Dictionary	m	m	사용자 영역의 상위 키
2	OID	String	m	m	발급 센터의 OID
	DocNum	Integer	m	m	발급된 문서의 수
	DocHash	String Array	m	m	해시값의 배열
	DocAlgo	String	m	m	해시 알고리즘
	TimeStamp	Date	m	m	값을 넣은 날짜와 시간(UTC)

```
1 0 obj
<<
 /Type /Catalog
 /Pages 2 0 R
 /PageLabels 3 0 R
 /Outlines 4 0 R
 /CEDCProof 5 0 R
>>
endobj
..... 중략
5 0 obj
<<
 /OID
 /DocNum 3
 /DocHash [(A1234...) (B5678...) (C9012...)]
 /DocAlgo /SHA256
 /TimeStamp (20260801090000Z)
>>
endobj
```

[PDF 증명서 구조 예시]

2. 데이터 삽입이 끝난 PDF 증명서의 전체 영역에 대해 발급 기업의 공인전자문서센터 인증서로 전자서명 한다.

증명서가 두 페이지 이상일 경우에 주의할 점은 다음과 같다.

1. 공인전자문서센터 로고 이미지는 마지막 페이지에 출력한다.
2. 온라인 확인 안내문은 모든 페이지에 출력하도록 한다.
3. 페이지의 오른쪽 하단에 발급자 및 증명서의 일련번호와 함께 현재페이지/전체페이지를 기술하도록 한다.

PDF 증명서의 각 항목 값을 기술할 때 주의할 점은 다음과 같다.

1. 확장필드이거나 선택적 생성필드로서 실제 증명서에 생성되지 않은 항목인 경우, PDF 증명서 포맷의 해당 항목을 '-'로 표시한다. 단, 수임자 정보는 실제 증명서에 권한 부여 필드가 생성되지 않았다면 PDF 증명서에서 항목자체를 생략한다.
2. 실제 증명서에 "NULL" 형식으로 생성된 필드인 경우에도, PDF 증명서 포맷의 해당 항목은 '-'로 표시한다.
3. 특정한 의미를 지닌 숫자나 비트값에 대하여, 해당 값의 의미를 한글로 표현하도록 하며 각 항목에서 사용할 용어는 다음과 같다.

   가. 증명서 용도 : 온라인, 모바일, 종이출력

   나. 사유 : 이용자 요청, 공인전자문서센터 내부업무, 보존기간 만료

   다. 수임자 권한 : 정당한 수신자, 문서 열람, 문서 발급

6. 기타 항목들의 표현 방법은 각 증명서에 대한 종이증명서 화면을 참고하도록 한다.

※ PDF 증명서는 추가되는 값(해시 알고리즘, 해시값, 전자서명 등)을 제외하고 종이 증명서와 같은 항목을 갖는다.

7.2.3. PDF 증명서 전자서명

공인전자문서센터 PDF 증명서의 전자서명은 ETSI EN 319 122의 CAdES 표준의 PDF 서명 방식인 PAdES로 서명한다. 해당 서명 방식에 관한 자세한 사항은 ISO 32000-2:2020의 "12.8.3.4 CAdES signatures as used in PDF"와 ETSI EN 319 122 PAdES에 기술되어 있다. 다만, PDF 증명서 전자서명에는 다음 세 가지 사항에 주의해야 한다.

해시 알고리즘은 SHA-256(256bits) 이상을 사용해야 하며 전자서명 알고리즘은 RSA(2048bits 이상) 혹은 ECDSA(256bits 이상)를 사용해야 한다. 관련 자세한 내용은 KISA "암호 알고리즘 및 키 길이 이용 안내서"를 참조한다. 또한, CMS 서명 데이터에는 서명자 인증서와 발급자 인증서를 포함해야 한다. 마지막으로 사용자 영역인 CEDCProof를 포함하여 전자서명 해야 한다.

7.2.4. 한국인터넷진흥원 로고 이미지

PDF 증명서에는 한국인터넷진흥원 로고를 사용할 수 있다. 한국인터넷진흥원 로고 이미지는 한국인터넷진흥원 홈페이지에서 받을 수 있는 CI 이미지를 활용하여 공인전자문서센터로 지정된 사업자가 자유롭게 선택하여 디자인할 수 있다. 다만, 제공된 CI 이미지를 임의로 수정해선 안 되며, CI의 활용 사실을 한국인터넷진흥원에 알려야 한다.

7.2.5. 온라인 확인 안내문

PDF 증명서에는 증명서의 내용을 온라인으로 확인할 수 있다는 안내문과 함께 실제 증명서의 내용을 확인할 수 있는 공인전자문서센터 웹페이지의 URL이나 QR 코드 등이 들어가야 한다.

7.2.6. PDF 증명서 검증

PDF 증명서에는 증명서 내용의 무결성과 부인방지가 포함되어 있다. PDF 증명서의 검증은 전자서명의 검증과 첨부파일의 해시값 비교검증 두 단계로 진행된다.

1. PDF 증명서의 전자서명 검증 방법은 크게 “문서(PDF 증명서)의 무결성 확인”과 “인증서 신뢰성 확인” 두 단계로 진행한다.

※ 자세한 검증 방법은 PDF 공식 표준문서(ISO 32000-2:2020) 참고

2. “문서(PDF 증명서)의 무결성 확인”은 서명 데이터 추출 및 무결성 검증이며, “인증서 신뢰성 확인”은 인증서 경로 및 유효성 검증이다.

3. PDF 증명서와 함께 발급된 전자문서의 비교검증은 PDF 증명서 발급시 포함된 DocHash 내 해시값을 이용해서 각 전자문서의 해시값을 비교해서 검증한다.

1, 2번 단계를 통해 PDF 증명서의 무결성을 검증하고, 3단계를 통해 함께 발급된 전자문서의 무결성을 검증한다.

부 록

1. ASN.1 구조

증명서 요청 메시지 및 응답 메시지는 전자서명 구조인 IETF RFC 3852 CMS (Cryptographic Message Syntax)에서 제시하는 ContentInfo 구조체로 표현된 signedData의 구조를 사용한다.

```
ContentInfo ::= SEQUENCE {
        contentType ContentType,
        content [0] EXPLICIT ANY DEFINED BY contentType }

ContentType ::= OBJECT IDENTIFIER

id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
rsadsi(113549) pkcs(1) pkcs7(7) 2 }

SignedData ::= SEQUENCE {
        version             CMSVersion,
        digestAlgorithms    DigestAlgorithmIdentifiers,
        encapContentInfo    EncapsulatedContentInfo,
        certificates        [0] IMPLICIT CertificateSet OPTIONAL,
        crls                [1] IMPLICIT RevocationInfoChoices OPTIONAL,
        signerInfos         SignerInfos }
```

본 규격 외부의 표준 규격으로부터 import 된 구조체에 대하여, context specific tag에 대한 tag mode의 정의를 비롯한 구조체의 생성 규칙은, 본 규격에서 재정의 되지 않는 한, 해당 표준 규격을 준용한다.

증명서 요청 메시지 및 응답 메시지 생성 시의 인코딩 방식은 DER 인코딩 또는 JER, XER 인코딩 방식을 사용해야 한다.

1.1. 증명서 요청 및 응답 메시지 정의

ARCCertificate { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) }

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

-- EXPORTS ALL --

IMPORTS

Extensions, AlgorithmIdentifier, CertificateSerialNumber FROM PKIX1Explicit88 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-pkix1-explicit-88(1)}

GeneralName, PolicyInformation FROM PKIX1Implicit88 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-pkix1-implicit-88(2)}

PKIStatusInfo FROM PKIXCMP {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-cmp(9)}

ContentInfo, IssuerAndSerialNumber, SubjectKeyIdentifier FROM CryptographicMessageSyntax2004 {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) modules(0) cms-2004(24)}

IdentifyData FROM IDNumber88 { iso(1) member-body(2) korea(410) kisa(200004) npki(10) attributes(1) identifyData(1) } ;

-- 증명서 요청 메시지

id-kiec-arcCertRequest OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) 1 }

ARCCertRequest ::= SEQUENCE {

version ARCVersion DEFAULT v1,

```
        requester               Requester,
        requestTime             RequestTime,
        policy                  ARCCertificatePolicies,
        target                  Target,
        nonce                   INTEGER,
        extentions              [0] EXPLICIT Extensions OPTIONAL }

  ARCVerison ::= INTEGER { v1(1), v2(2)}

  Requester     ::= CHOICE {
        generalNames            GeneralNames,
        null                    NULL    }

  GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

  id-kiec-HashedIDNInfo OBJECT IDENTIFIER ::=   { iso(1) member-body(2)
korea(410) kiec(200032) certificate(2) attributes(4) 1 }

  HashedIDNInfo::= SEQUENCE  {
        hashAlg         HashAlgorithm,
        hashedIDN       OCTET STRING   }

  HashAlgorithm ::= AlgorithmIdentifier

  RequestTime ::= CHOICE {
        generalizedTime         GeneralizedTime,
        null                    NULL    }

  ARCCertificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation
```

```
Target ::= CHOICE {
        targetRecord            TargetRecord,
        targetHash              [0] EXPLICIT HashedDataInfo,
        targetDocInfo           [1] EXPLICIT TargetDocInfo }

TargetRecord ::= SEQUENCE {
        serialNo        INTEGER,
        opType          OperationType }

OperationType ::= ENUMERATED {
        register                (0),
        issue                   (1),
        transfer                (2),
        delete                  (3)}

HashedDataInfo ::= SEQUENCE {
        hashAlg                 HashAlgorithm,
        hashedData              BIT STRING      }

TargetDocInfo ::= SEQUENCE {
        packageID               PackageIdentifier,
        docID                   [0] EXPLICIT DocumentIdentifier OPTIONAL,
        fileIDs                 [1] EXPLICIT FileIDs OPTIONAL,
        issuedDocOriginal       BOOLEAN }

PackageIdentifier ::= UTF8String

DocumentIdentifier ::= UTF8String
```

FileIDs::= SEQUENCE SIZE (1..MAX) OF FileIdentifier

FileIdentifier ::= UTF8String

id-kiec-qualifications OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 1 }

Qualifications ::= SEQUENCE SIZE (1..MAX) OF Qualification

Qualification ::= SEQUENCE {
nomineeInfo NomineeInfo,
nomineeRole NomineeRole }

NomineeInfo ::= SEQUENCE {
nominee [0] EXPLICIT GeneralNames OPTIONAL,
nomineeCert [1] EXPLICIT CertIdentifier OPTIONAL }

CertIdentifier ::= CHOICE {
issuerAndSerialNumber [0] EXPLICIT IssuerAndSerialNumber,
subjectKeyIdentifier [1] EXPLICIT SubjectKeyIdentifier }

NomineeRole ::= BIT STRING {
onlyForNominee (0),
readDocument (1),
downloadDocument (2) }

id-kiec-usageType OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 2 }

```
UsageType ::= BIT STRING {
        online              (0),
        mobile              (1),
        paperEnable         (2) }
```

id-kiec-dateOfExpiration OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 3 }

DateOfExpiration ::= GeneralizedTime

id-kiec-certifiedTime OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 4 }

CertifiedTime ::= GeneralizedTime

id-kiec-certUsage OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 5 }

CertUsage ::= BMPString (SIZE (1..128))

id-kiec-docContentInfoFlag OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) aRCCertificateExtensions(3) 6 }

```
DocContentInfoFlag ::= BIT STRING {
        title               (0),
        keyword             (1),
        description         (2) }
```

-- 증명서 응답 메시지

id-kiec-arcCertReseponse OBJECT IDENTIFIER ::= { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) 2 }

```
ARCCertResponse ::= CHOICE {
    arcCertInfo             [0] EXPLICIT ARCCertInfo ,
    arcErrorNotice          [1] EXPLICIT ARCErrorNotice }

ARCCertInfo ::= SEQUENCE {
        version             [0] EXPLICIT ARCVersion DEFAULT v1,
        serialNumber        SerialNumber,
        issuer              GeneralNames,
        dateOfIssue         GeneralizedTime,
        dateOfExpiration    CertDateOfExpiration,
        policy              ARCCertificatePolicies,
        requestInfo         RequestInfo,
        target              TargetToCertify,
        extentions          [1] EXPLICIT Extensions OPTIONAL }

SerialNumber  ::=  INTEGER

CertDateOfExpiration ::= CHOICE {
        dateOfExpiration    DateOfExpiration,
        null                NULL    }

ARCCertificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

RequestInfo  ::= CHOICE {
        arcCertRequest      ARCCertRequest,
        null                NULL    }
```

```
TargetToCertify  ::= CHOICE {
        opRecord                [0] EXPLICIT OperationRecord,
        orgAndIssued            [1] EXPLICIT OriginalAndIssuedDocumentInfo,
        dataHash                [2] EXPLICIT HashedDataInfo }

OperationRecord  ::= SEQUENCE {
        serialNo                INTEGER,
        opRequesterInfo         OperationRequesterInfo,
        opRequestTime           GeneralizedTime,
        opTime                  GeneralizedTime,
        opType                  OperationType,
        orgDocInfo              PackageDocumentInfo,
        issuedDocInfo           [0] EXPLICIT PackageDocumentInfo OPTIONAL,
        peerARCInfo             [1] EXPLICIT PeerARCInfo OPTIONAL,
        reason                  [2] EXPLICIT Reason OPTIONAL }

OperationRequesterInfo ::= CHOICE {
        opRequester             GeneralNames,
        null                    NULL }

PackageDocumentInfo  ::= SEQUENCE {
        packageID               PackageIdentifier,
        docInfo                 DocumentInfo }

DocumentInfo  ::= SEQUENCE {
        docID                   DocumentIdentifier,
        fileIDs                 [0] EXPLICIT FileIDs OPTIONAL,
        docHash                 DocumentHash }
```

```
DocumentHash ::= SEQUENCE {
        hashAlg                 HashAlgorithm,
        hashedDocument          BIT STRING      }

PeerARCInfo   ::= SEQUENCE {
        peerARC                 GeneralNames,
        peerARCPackageID        PackageIdentifier }

Reason   ::=   BIT STRING   {
        userRequest             (0),
        arcRequest              (1),
        expired                 (2)       }

OriginalAndIssuedDocumentInfo ::= SEQUENCE {
        orgDocInfo              PackageDocumentInfo,
        issuedDocInfo           PackageDocumentInfo,
        issuedDocOriginal       BOOLEAN }

ARCErrorNotice   ::=   SEQUENCE   {
      transactionStatus         PKIStatusInfo ,
      transactionIdentifier     GeneralName OPTIONAL       }

DocContentInfo   ::= SEQUENCE {
        title                   [0] EXPLICIT BMPString (SIZE (1..128)) OPTIONAL,
        keyword                 [1] EXPLICIT BMPString (SIZE (1..35)) OPTIONAL,
        description             [2] EXPLICIT BMPString (SIZE (1..1000)) OPTIONAL }

END
```

1.2. 에러 메시지

번호	메시지	내용
0	badAlg	정의되지 않은 알고리즘
1	badMessageCheck	메시지 무결성 손상 (전자서명 검증 실패 등)
2	badRequest	요청메시지의 불허가 또는 미지원
3	badTime	시스템과의 많은 시간
5	badDataFormat	잘못된 데이터 포맷
6	wrongAuthority	잘못된 기관(요청메시지의 기관명이 응답 메시지 생성 기관과 다름)
7	incorrectData	요청자의 데이터가 다름
8	missingTimeStamp	타임스탬프가 필요하나 생성되지 않음
13	badRecipientNonce	잘못된 사용자 nonce
15	unacceptedPolicy	요청된 정책 미 지원
16	unacceptedExtension	요청된 확장 미 지원
18	badSenderNonce	잘못된 송신자 nonce

2. IdentifyData 구조체의 생성 및 검증

IdentifyData는 각 주체의 실명 및 식별번호를 증명서에 포함하기 위하여 사용되는 구조체로서, 증명서 요청자, 증명서 발급자, 증명서 수임자 등을 증명요청서와 증명서에 포함할 때는 주체의 실명 및 식별번호로 구성된 IdentifyData를 GeneralName의 otherName 필드를 사용하여 설정한다.

2.1. 생성

IdentifyData의 구조는 아래와 같다.

```
IdentifyData ::= SEQUENCE  {
        realName        UTF8String,
        userInfo        SEQUENCE SIZE (1..MAX) OF
                        AttributeTypeAndValue OPTIONAL  }
```

먼저 주체의 실명을 UTF8String으로 인코딩한 후 realName 필드에 설정한다.

userInfo 필드는 HashedIDNInfo 구조체를 생성하여 설정하게 되며, HashedIDNInfo의 구조는 아래와 같다.

```
id-kiec-HashedIDNInfo OBJECT IDENTIFIER ::=  { iso(1) member-body(2) korea(410) kiec(200032) certificate(2) attributes(4) 1 }

HashedIDNInfo::= SEQUENCE  {
        hashAlg         HashAlgorithm,
        hashedIDN       OCTET STRING  }

HashAlgorithm ::= AlgorithmIdentifier
```

HashedIDNInfo는 식별번호를 가공하여 생성한 구조체로서, 사업자의 식별번호로는 사업자번호를 사용하도록 한다.

증명서 요청 시 요청의 주체, 즉 증명서 요청자는 사업자가 아닌 개인일 경우도 있는데, 이때는 민감정보(CI/DI 등)의 도용을 방지하기 위하여 160비트의 안전한 난수를 생성하여 주민번호와 연접하여 사용한다. 단, 주민번호를 센터에서 수집·보관·활용이 불가능하기 때문에 주민번호 사용에 유의해야 한다. 민감정보가 아닌 것을 식별번호로 사용할 경우에는 사업자번호와 동일하게 처리가 가능하다.

이때 증명서 요청자가 생성한 난수는 증명요청서와 함께 공인전자문서센터에 전달되어, 공인전자문서센터가 생성한 증명서와 함께 보관되어야 하며, 이후 증명서 검증자가 공인전자문서센터에 증명서의 폐지 여부에 대한 확인 요청 시에 검증 결과와 함께 전달되어 증명서 요청자의 신원 확인을 위하여 사용된다.

증명서 요청자가 난수를 공인전자문서센터에 전달하는 방법은 연계 인터페이스 규격의 '증명서 발급' 항목을 참조하도록 한다.

식별번호를 사용하여 HashedIDNInfo 구조를 만드는 과정은 다음과 같다.

먼저 string 상태의 식별번호에서 '-' 등의 구분자 및 빈 공간이 있으면 이를 제거한 후, 민감정보인 경우는 민감정보 뒤에 160 비트의 난수를 연접한다. 그리고 안전한 해시 알고리즘을 사용하여 두 번 해시하여, HashedIDNInfo에 해시 알고리즘과 함께 설정한다.

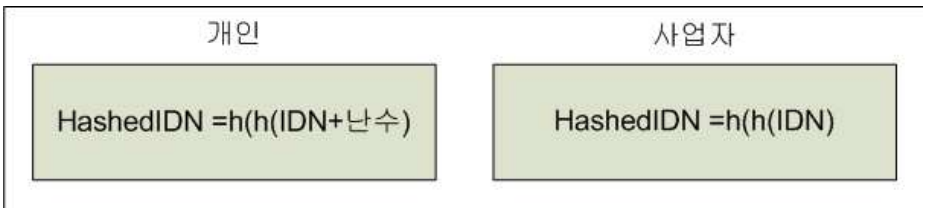


그림 5. HashedIDN 생성

2.2. 검증

IdentifyData의 검증은 증명서 요청자, 증명서 발급자, 증명서 수임자 등 주체의 실명과 식별번호를 IdentifyData를 생성할 때와 동일한 방법으로 가공하여 값을 비교하도록 한다.

즉, 실명을 UTF8String으로 인코딩한 후, realName 필드에 설정된 값과 비교하고, string 상태의 식별번호를 hashAlg에 설정된 해시 알고리즘을 사용하여 두 번 해시한 후, hashedIDN에 설정된 값과 비교한다.

증명서 요청자가 개인일 경우, 증명서 요청자에 대한 검증을 수행하기 위해서는, 공인전자문서센터에 증명서와 함께 보관 중인 난수가 필요하며, 증명서 검증자는 증명서의 유효성 검증의 단계에서 증명서의 폐지 여부를 공인전자문서센터에 요청하여 확인하는 절차 중에 응답메시지에 포함된 난수를 획득하여야 한다.

증명서 검증자가 난수를 공인전자문서센터로부터 획득하는 방법은 연계 인터페이스 규격의 '증명서 검증' 항목을 참조하도록 한다.

증명서 검증자는 공인전자문서센터로부터 획득한 난수를 증명서 요청자의 식별번호인 민감정보 뒤에 연접하여 해시한 후, hashedIDN에 설정된 값과 비교한다.

주의할 점은 증명서 요청자와 증명서 검증자 간에 합의된 경우를 제외하고, 개인의 식별번호인 민감정보의 노출을 방지하기 위하여, 이 모든 과정은 증명서 검증자의 내부 시스템에서 이루어져야 하며, 증명서 검증자가 직접 민감정보나 난수를 인식할 수 없어야 한다.

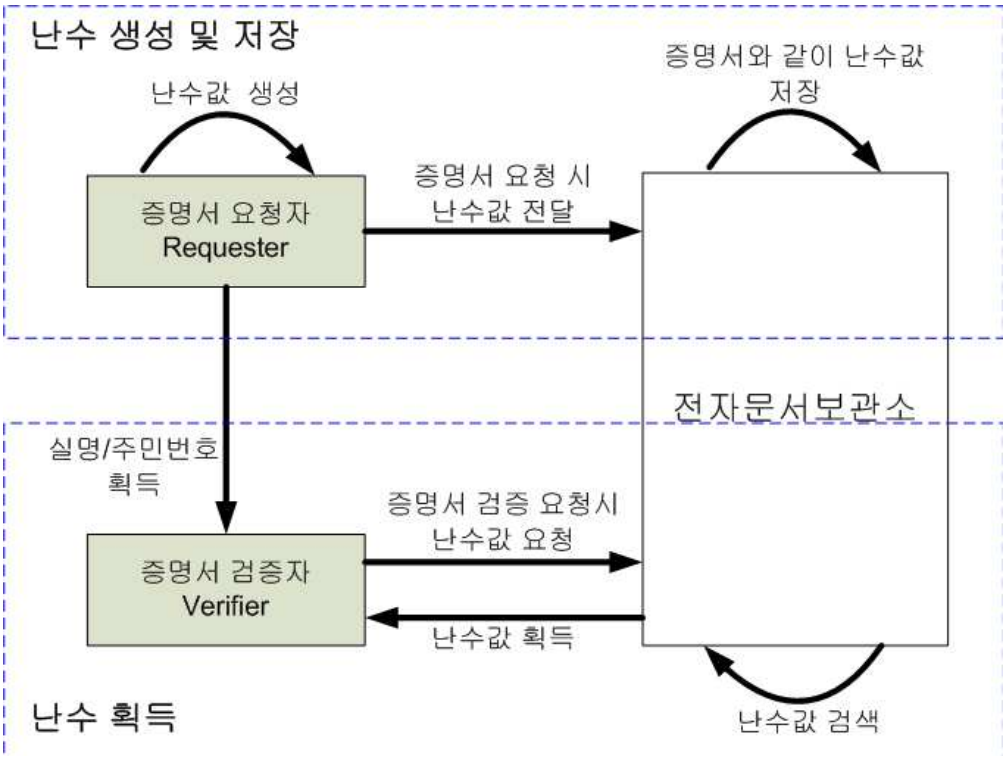


그림 6. 난수 생성 및 획득 프로세스

3. Qualificatiions (권한부여) 필드의 생성 및 검증

3.1. 생성

Qualifications 필드는 증명 요청서 및 증명서에 포함되는 확장필드로서, 수임자 정보를 나타내기 위하여 사용된다.

증명서 요청자는 수임자의 실명 및 식별번호, 또는 수임자의 공인 인증서를 획득하여 Qualifications 필드를 생성한 후, 증명 요청서에 포함하여 공인전자문서센터에 송신한다.

공인전자문서센터는 증명 요청서에 포함된 Qualifications 필드를 추출하여 그대로 증명서에 포함하여 발급한다. 이때 공인전자문서센터는 Qualifications 필드의 구조만을 검증할 뿐, 내용을 검증할 필요는 없다.

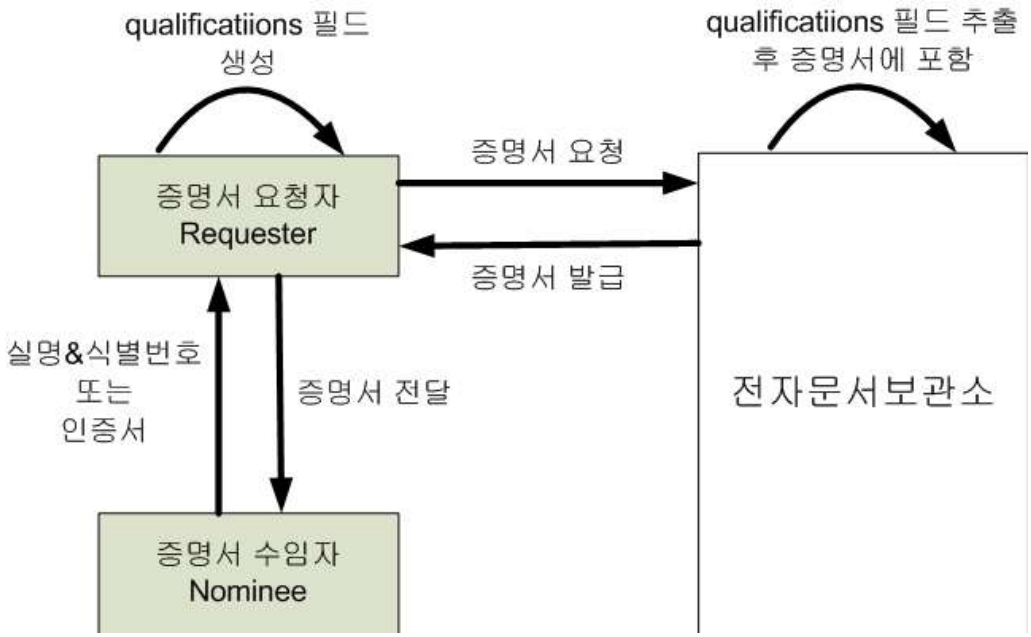


그림 7. Qualifications 필드 생성

Qualifications 필드의 구조는 아래와 같다.

Qualifications ::= SEQUENCE SIZE (1..MAX) OF Qualification

```
Qualification ::= SEQUENCE {
        nomineeInfo     NomineeInfo,
        nomineeRole     NomineeRole   }
```

3.1.1. nomineeInfo 필드

nomineeInfo 필드는 수임자의 정보를 나타내는 필드로서 구조는 아래와 같다.

```
NomineeInfo   ::= SEQUENCE {
        nominee                 [0] EXPLICIT GeneralNames OPTIONAL,
        nomineeCert             [1] EXPLICIT CertIdentifier OPTIONAL }
```

nominee 필드는 수임자의 실명 및 식별번호로 구성되며, nomineeCert 필드는 수임자의 인증서 식별정보로 구성되는데, 이때 수임자의 실명 및 식별번호나 인증서 식별정보를 획득하는 과정은 본 규격에서는 다루지 않는다.

3.1.1.1. nominee 필드

nominee 필드는 수임자의 실명 및 식별번호로 구성된 IdentifyData를 GeneralName의 otherName 필드를 사용하여 설정한다.

수임자가 개인일 경우에는 nominee 필드를 사용할 수 없으며 반드시 nomineeCert 필드만을 사용해야 한다.

IdentifyData의 생성 방법은 “부록 2.1 IdentifyData 구조체의 생성”의 절차와 동일하다.

3.1.1.2. nomineeCert 필드

nomineeCert 필드는 수임자의 인증서 식별정보를 나타내는 필드로서 구조는 아래와 같다.

```
CertIdentifier   ::= CHOICE {
        issuerAndSerialNumber       [0] EXPLICIT IssuerAndSerialNumber,
        subjectKeyIdentifier        [1] EXPLICIT SubjectKeyIdentifier }

IssuerAndSerialNumber ::= SEQUENCE {
        issuer          Name,
        serialNumber    CertificateSerialNumber }

CertificateSerialNumber ::= INTEGER

SubjectKeyIdentifier ::= OCTET STRING
```

issuer 필드는 수임자 인증서의 issuer 필드의 값을 설정하며, serialNumber 필드는 수임자 인증서의 serialNumber 필드의 값을 설정한다.

subjectKeyIdentifier 필드는 수임자 인증서의 SubjectKeyIdentifier 확장필드의 값을 설정한다.

3.1.2. nomineeRole 필드

nomineeRole 필드는 수임자의 역할 또는 권한을 나타내는 필드로서 구조는 아래와 같다.

```
NomineeRole ::= BIT STRING   {
        onlyForNominee          (0),
        readDocument            (1),
        downloadDocument        (2)   }
```

수임자에게 부여할 역할 또는 권한에 해당하는 값을 설정한다.

만약 onlyForNominee 값이 설정되어야 한다면, Qualifications 필드 내의 모든 Qualification 항목의 NomineeRole에 onlyForNominee값이 포함되어야 한다.

3.2. 검증

Qualifications 필드의 검증이란, 수임자로서의 역할을 수행하거나 권리를 주장하는 자의 신원정보와 Qualifications 필드의 각 Qualification 항목에 대하여 nomineeInfo 필드에 설정된 수임자 정보가 일치하는가를 확인하는 것이며, 검증에 성공한 경우에만 정당한 수임자로 인정받아 역할을 수행하거나, 권리를 행사할 수 있다.

단, Qulifications 필드의 검증에 실패하였다 하더라도, 수임자로서의 역할을 수행하거나 권리를 주장하는 자가 정당한 수임자가 아님을 확인한 것이므로, 증명서 자체의 유효성에 영향을 미치지는 않는다.

Qualifications 필드에 대한 검증 절차는 각 Qualification 항목에 명시된 수임자 본인이 검증자가 되어 자신에게 발급된 증명서인가의 여부를 검증하는 경우와 수임자 이외의 일반 검증자가 증명서의 Qualifications 필드에 명시된 수임자 정보 중에 수임자로서의 역할을 수행하거나 권리를 주장하는 자가 포함되어 있는가의 여부를 검증하는 경우로 구분된다.

3.2.1. 수임자 본인 검증

수임자 본인이 검증자가 되어 정당한 수임자인가의 여부를 확인하는 경우는 본문 6.2.2에서 서술한 것처럼 증명서의 내용을 검증하는 과정 중에 행해진다.

즉, 증명서의 내용을 검증하는 과정에서 Qualifications 확장필드의 critical 값이 TRUE 라면, 증명서 검증자가 증명서 수임자와 동일해야만 증명서를 활용할 수 있으므로, 증명서 검증자는 자신의 신원정보를 사용하여 자신이 증명서 수임자 임을 검증해야 한다.

수임자 정보를 나타내기 위하여 nominee 필드가 사용되었다면, 이때의 검증 과정은 “부록 2.2 IdentifyData 구조체의 검증”의 절차와 동일하다.

만약 nomineeCert 필드가 사용되었다면, 설정된 issuerAndSerialNumber 필드 또는 subjectKeyIdentifier 필드의 값과 검증자의 인증서에서 각각의 필드에 해당하는 값을 추출하여 비교 검증한다.

nominee 필드와 nomineeCert 필드가 동시에 사용었다면, 반드시 nominee 필드의 값을 비교 검증한 결과를 사용해야 한다.

그림 8. 수임자 본인 검증

3.2.2. 일반 검증자 검증

일반 검증자 검증이란 수임자임을 주장하는 자에 대하여 제3의 검증자가 검증을 수행하는 것으로서, 일반적으로 공인전자문서센터가 수임자의 전자문서 열람권한이나 전자문서 발급권한을 확인할 때 발생하지만, 검증자가 반드시 공인전자문서센터일 필요는 없으며, 일반인도 수임자로부터 수임자 검증에 필요한 데이터를 전송받아 수임자에 대한 검증을 수행할 수 있다.

증명서에 명시된 정당한 수임자임을 주장하는 자의 검증 과정에는 반드시 수임자임을 주장하는 자의 신원을 확인하기 위한 인증서가 필요하다.

검증 과정은 증명서의 nomineeInfo 필드의 수임자 정보와 인증서에 포함된 인증서 소유자의 신원정보 또는 인증서 식별정보와의 비교 검증 단계와 수임자임을 주장하는 자가 해당 인증서의 정당한 소유자인가를 검증하는 단계로 구분되며, 두 단계의 순서는 바뀌어도 무방하다.

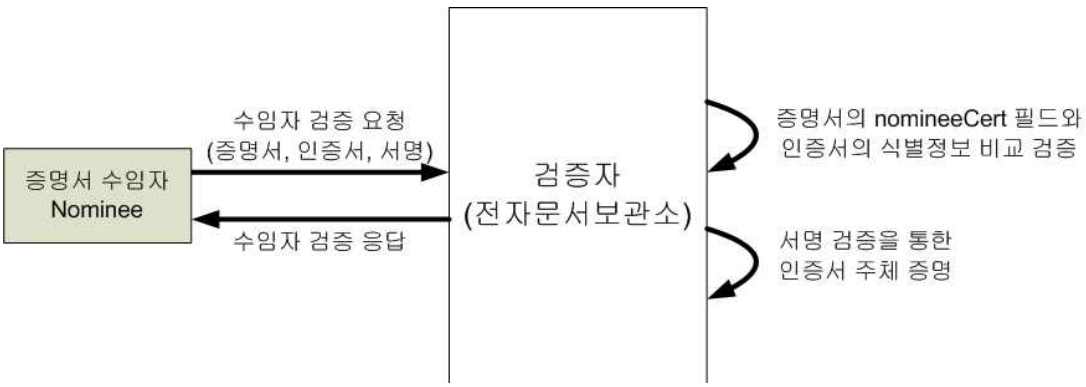


그림 9. nomineeCert 필드를 사용한 검증

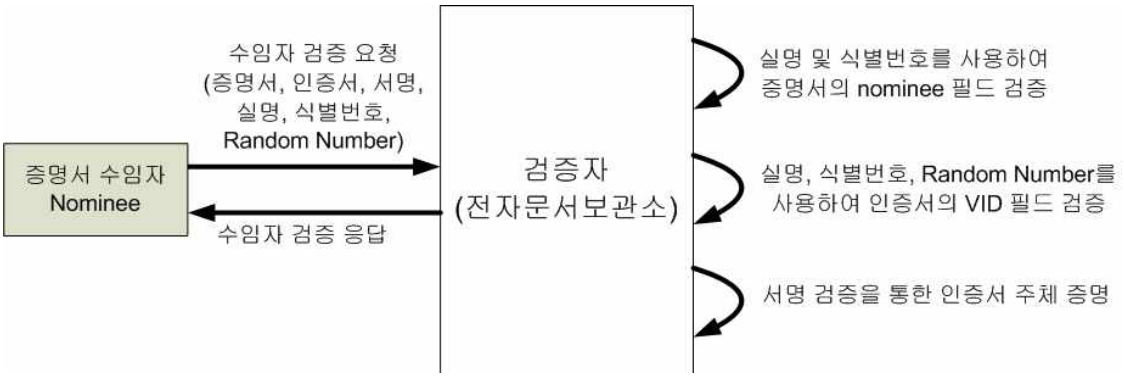


그림 10. nominee 필드를 사용한 검증

3.2.2.1. 인증서 정보와의 비교 검증

증명서의 정당한 수신자임을 주장하는 자는 검증자에게 증명서와 함께 반드시 자신의 신원을 증명하는 인증서를 전달해야 한다.

만약 nomineeInfo 필드에 수임자의 실명 및 식별번호를 가공하여 생성한 nominee 필드가 설정되었다면, 이를 검증하기 위한 자신의 실명 및 식별번호도 함께 검증자에게 전달해야 한다.

검증자는 전달받은 실명 및 식별번호를 사용하여 부록 "2.2 IdentifyData 구조체의 검증"과 동일한 절차로 검증한다. 이는 증명서가 해당 실명 및 식별번호의 주체에게 발급되었다는 것을 증명할 뿐, 함께 전달된 인증서의 주체가 해당 실명 및 식별번호의 주체와 동일인 이라는 것을 증명하는 것은 아니므로, 이를 확인하기 위하여 동일한 실명 및 식별번호를 사용하여 인증서에 포함된 신원정보인 VID 값의 검증을 수행하여야 한다.

실명 및 식별번호를 사용하여 인증서에 포함된 VID 값을 검증하는 절차는 공인인증체계의 "식별번호를 이용한 본인확인 기술규격"의 "부록 B. 식별번호 검증예제" 중 '1. 응용사이트에서 식별번호를 필요로 하는 경우'에 서술된 절차를 따르며, 이 경우 개인키 저장포맷 내에 포함된 Random Number도 함께 전송하여야 한다.

만약 nomineeInfo 필드에 nomineeCert가 설정되었다면, nomineeCert 필드에 설정된 issuerAndSerialNumber 필드 또는 subjectKeyIdentifier 필드의 값과 검증자의 인증서에서 각각의 필드에 해당하는 값을 추출하여 비교 검증한다.

nominee 필드와 nomineeCert 필드가 동시에 사용되었다면, 반드시 nominee 필드의 값과 인증서의 VID 값을 검증한 결과를 사용해야 한다.

3.2.2.2. 인증서 주체 증명

인증서 정보와의 비교 검증이 성공했다면, 증명서는 함께 전달된 인증서의 주체에게 발급되었음이 증명된 것이며, 증명서의 정당한 수신자임을 주장하는 자와 인증서의 주체가 동일인임을 검증해야만 Qualifications 필드의 검증이 완료된다.

인증서의 주체 증명은 인증서에 포함된 공개키와 한 쌍인 개인키에 대한 소유 증명의 방법을 사용하여 이루어진다.

일반적으로 인증서의 주체임을 증명하고자 하는 자가 자신의 개인키를 사용하여 임의의 데이터에 서명을 한 후, 인증서, 데이터, 그리고 서명값을 검증자에게 전달하면, 검증자는 인증서에 포함된 공개키를 사용하여 데이터와 서명값을 검증하게 된다.

이때 인증서 주체 증명을 위한 서명 및 서명의 대상이 되는 임의의 데이터는 수임자 검증의 상황에 따라 다른데, 예를 들어, 증명서에 수임자로 명시된 이용자가 증명서를 사용하여 공인전자문서센터에 전자문서 열람이나 발급을 요청할 때 수행되는 수임자 검증에서는 연계 인터페이스 규격에서 정의된 요청메시지와 요청메시지의 무결성 및 부인방지를 위하여 첨부되는 서명이 본 규격의 인증서 주체 증명을 위한 임의의 데이터 및 서명이 된다.

인증서의 주체 증명 단계와 인증서 정보와의 비교 검증 단계의 순서가 바뀌거나 동시에 처리되어도 무방하다.

4. 전자문서 수관 시의 최초등록증명서 재발급

4.1. 재발급 절차

공인전자문서센터 간 전자문서 이·수관 시 수관 공인전자문서센터는 이관 공인전자문서센터로부터 전달받은 TIP에 첨부된 최초등록증명서에 대하여 수관 공인전자문서센터의 정보를 사용하여 재발급하여야 한다. 재발급되는 최초등록증명서는 수관된 전자문서에 대한 최초의 공인전자문서센터 등록 사실에 대한 보증서로서, 수관 공인전자문서센터의 전자문서 수관작업에 대한 보증서인 등록증명서와 구분된다.

최초등록증명서의 재발급 절차는 다음과 같다.

□ 이관 공인전자문서센터는 이관 대상 전자문서에 대한 최초등록증명서가 발급되지 않았다면 발급을 수행한다.

□ 이관 공인전자문서센터는 전자서명의 유효기간이 만료된 최초등록증명서에 대하여 갱신을 수행한다.

□ 이관 공인전자문서센터는 전자문서 이관 시 최초등록증명서를 TIP에 첨부하여 수관 공인전자문서센터에 전달한다.

□ 수관 공인전자문서센터는 이관 공인전자문서센터로부터 전달받은 TIP 및 최초등록증명서를 검증한다.

□ 수관 공인전자문서센터는 수관 작업을 수행한 후, 수관 사실에 대한 등록증적을 생성하여 저장한다.

□ 수관 공인전자문서센터는 이관 공인전자문서센터로부터 전달받은 최초등록증명서의 증명대상 정보를 그대로 유지하여 최초등록증명서를 재생성하여 저장한다.

□ 수관 공인전자문서센터는 재생성된 최초등록증명서에 대하여 이용자가 다운로드 받을 수 있도록 다운로드 서비스를 제공하도록 한다.

상기의 절차에 기술된, 수관 공인전자문서센터에서 생성한 등록증적은 수관 사실에 대한 근거 정보로서, 전자문서의 최초 등록 시점을 보증하는 최초등록증명서와는 관련이 없음에 주의한다.

이후, 전자문서의 수관 공인전자문서센터 등록여부에 대한 확인이 필요한 경우는

수관 사실에 대한 등록증적을 이용하여 등록증명서를 발급하도록 하고, 전자문서의 최초 등록 사실에 대한 확인이 필요한 경우는 재발급된 최초등록증명서를 다운로드 하도록 한다.

4.2. 생성

수관 공인전자문서센터는 최초등록증명서 재발급 시 하기의 방식으로 생성하도록 한다.

- □ version : 본 규격 버전에서는 버전 1을 사용
- □ serialNumber : 수관 공인전자문서센터의 정책에 따라 신규 일련 번호를 부여
- □ issuer : 수관 공인전자문서센터의 실명 및 식별번호
- □ dateOfIssue : 최초등록증명서 재발급 시점
- □ dateOfExpiration : 최초등록증명서의 기존 증명서 효력 만기일
- □ policy : 수관 공인전자문서센터의 최초등록증명서 정책
- □ requestInfo : NULL
- □ target : 최초등록증명서의 기존 증명대상
- □ extensions : 생성안함

수관 공인전자문서센터는 상기의 방식으로 증명서 기본 필드를 생성한 후 수관 공인전자문서센터의 전자서명 및 기타 정보를 추가하여 최초등록증명서를 재발급한다.

4.3. 검증

재발급된 최초등록증명서에 대한 검증 방식은 일반적인 증명서 검증 방식과 동일하다.

5. 필드 생성 및 처리 기준

5.1. 증명요청서

증명요청서 필드의 생성 주체는 증명서 요청자이며, 처리 주체는 공인전자문서센터이다. 여기에서 처리란 공인전자문서센터가 해당 필드의 내용을 이해하고, 증명서 생성 시 반영함을 의미한다.

반드시 처리해야 하는 증명요청서의 필드를 처리하지 못할 시에, 공인전자문서센터는 증명서 요청자에게 증명서 발급 대신 에러 응답 메시지를 전송해야 한다.

증명서 필드의 생성 여부가 optional이라 하더라도, 증명요청서의 해당 필드의 처리 여부가 mandatory라면 증명서 생성 시 해당 필드를 반드시 생성해야 한다.

5.1.1. 기본 필드

번호	필드명	TYPE	생성	처리	비고
1	버전(version)	ARCVersion	m	m	DEFAULT v1 (①) INTEGER { v1(1), v2(2)}
2	requester	Requester	m	m	CHOICE
	generalNames	GeneralNames (②)	o	m	Sequence Size(1..MAX) Of GeneralName, null과 choice 관계
	otherName	OtherName	m	m	
	KISA 식별값 (type-id)	OID	m	m	id-kisa-identifyData (1.2.410.200004.10.1.1)
	value	IdentifyData	m	m	
	요청자 명 (realName)	UTF8String	m	m	한글 실명
	userInfo		m	m	Sequence Size(1..MAX) Of AttributeTypeAndValue
	KIEC 식별값(고정) (type)	OID	m	m	id-kiec-HashedIDNInfo (1.2.410.200032.2.4.1)
	value	HashedIDNInfo	m	m	
	해시알고리즘 식별자 (hashAlg)	HashAlgorithm	m	m	AlgorithmIdentifier (2.16.840.1.101.3.4.2.*)
	요청자식별번호해시값 (HashedIDN)	OCTET STRING	m	m	
	null	NULL	o	m	generalNames와 choice

번호	필드명	TYPE	생성	처리	비고
					관계
3	requestTime	RequestTime	m	m	CHOICE
	요청시간 (generalizedTime)	GeneralizedTime	o	m	null과 choice 관계
	null	NULL	o	m	generalizedTime와 choice 관계
4	policy	ARCCertificatePolicies	m	m	Sequence Size(1..MAX) Of PolicyInformation
	증명서 정책 OID (policyIdentifier)	CertPolicyId	m	m	OID
	policyQualifiers	Sequence Size(1..MAX) Of PolicyQualifierInfo	x	x	사용 안함
5	target	Target	m	m	CHOICE
	targetRecord	TargetRecord	o	m	targetHash, targetDocInfo와 choice 관계
	증명서 식별번호 (serialNo)	INTEGER	m	m	
	증명서 종류 (opType)	OperationType	m	m	ENUMERATED
	targetHash	HashedDataInfo	o	m	targetRecord, targetDocInfo와 choice 관계
	해시알고리즘 식별자 (hashAlg)	HashAlgorithm	m	m	AlgorithmIdentifier (2.16.840.1.101.3.4.2.*)
	증명 데이터 해사값 (hashedData)	BIT STRING	m	m	
	targetDocInfo	TargetDocInfo	o	m	targetRecord, targetHash와 choice 관계
	정보패키지 식별자 (packageID)	PackageIdentifier	m	m	UTF8String
	전자문서 식별자 (docID)	DocumentIdentifier	o	m	UTF8String
	첨부파일 식별자 (fileIDs)	Sequence Size(1..MAX) Of FileIdentifier	o	m	UTF8String
	원본/불변경증명서 구분자 (issuedDocOriginal)	BOOLEAN	m	m	원본증명서 TRUE 불변경증명서 FALSE
6	재사용/추측 공격 방지용 임시값 (nonce)	INTEGER	m	m	임의의 20byte 값

① v1인 경우 version 필드 생략

② GeneralName의 CHOICE에서 otherName 필드를 사용

5.1.2. 확장 필드

번호	필드명	TYPE	C	생성	처리	비고
1	Qualifications	Sequence Size(1..MAX) Of Qualification	t	o	m	id-kiec-qualifications
	nomineeInfo	NominieeInfo (①)		m	m	
	nominee	GeneralNames (②)		o	m	
	(requester의 generalNames와 동일한 형식으로 중략)					
	nomineeCert	CertIdentifier		o	m	CHOICE
	issuerAnd-SerialNumber	IssuerAndSerialNumber		o	m	subjectKeyIdentifier와 choice 관계
	수임자 인증서 발급자(issuer)	Name		m	m	
	수임자 인증서 일련번호 (serialNumber)	CertificateSerialNumber		m	m	INTEGER
	수임자 인증서 공개키 식별자 (subjectKey-Identifier)	SubjectKeyIdentifier		o	m	OCTET STRING, issuerAndSerialNumber와 choice 관계
	수임자의 권한 (nomineeRole)	BIT STRING		m	m	
2	증명서 이용환경 (UsageType)	BIT STRING	f	o	o	id-kiec-usageType
3	증명서 효력만기일 (DateOfExpiration)	GeneralizedTime	t/f	o	m/o (③)	id-kiec-dateOfExpiration
4	증명서 보증일시 (CertifiedTime)	GeneralizedTime	t	o	m	id-kiec-certifiedTime
5	증명서 용도 (CertUsage)	BMPString	t/f	o	m/o (③)	id-kiec-certUsage
6	전자문서 정보 (DocContentInfoFlag)	BIT STRING	t	o	m	id-kiec-docContentInfoFlag
7	증명서 버전 (CertVersion)	INTEGER	t/f	o	m	id-kiec-certVersion

① nominee 필드와 nomineeCert 필드 중 하나는 반드시 생성되어야 함

② 증명요청서 기본필드인 requester의 generalNames와 동일한 형식임

③ critical의 값이 TRUE이면 반드시 처리되어야 함

5.2. 증명서

증명서 필드의 생성 주체는 공인전자문서센터이며, 처리 주체는 모든 이용자이다. 여기에서 처리란 이용자가 해당 필드의 내용을 이해하고, 증명서 이용 시 반영함을 의미한다.

반드시 처리해야 하는 증명서의 필드를 처리하지 못할 시에, 이용자 소프트웨어는 에러 메시지를 리턴해야 한다.

5.2.1. 기본 필드

번호	필드명	TYPE	생성	처리	비고
1	버전(version)	ARCVersion	m	m	DEFAULT v1 (①) INTEGER { v1(1), v2(2)}
2	증명서 식별번호 (serialNumber)	INTEGER	m	m	
3	issuer	GeneralNames (②)	m	m	Sequence Size(1..MAX) Of GeneralName
	(requester의 generalNames와 동일한 형식으로 중략)				
4	증명서 발급일 (dateOfIssue)	GeneralizedTime	m	m	
5	dateOfExpiration	CertDateOfExpiration	m	m	CHOICE
	증명서 효력 만기일 (dateOfExpiration)	DateOfExpiration	o	m	null과 choice 관계
	null	NULL	o	m	dateOfExpiration과 choice 관계
6	policy	ARCCertificatePolicies	m	m	Sequence Size(1..MAX) Of PolicyInformation
	증명서 정책 OID (policyIdentifier)	CertPolicyId	m	m	OID
	policyQualifiers	Sequence Size(1..MAX) Of PolicyQualifierInfo	m	m	
	증명서 상세 정책 OID (policyQualifierId)	PolicyQualifierId	m	m	OID
	qualifier	Qualifier	m	m	CHOICE
	업무준칙 URI (cPSuri)	CPSuri	o	m	IA5String userNotice와 choice 관계
	userNotice	UserNotice	o	m	

번호	필드명	TYPE	생성	처리	비고
	noticeRef	NoticeReference	x	x	사용 안함
	정책 정보 (explicitText)	DisplayText	m	m	BMPString 사용
7	requestIinfo	RequestInfo	m	m	CHOICE
	증명서요청 메시지 정보 (arcCertRequest)	ARCCertRequest	o	m	null과 choice 관계
	null	NULL	o	m	arcCertRequest와 choice 관계
8	target	TargetToCertify	m	m	CHOICE
	opRecord	OperationRecord	o	m	orgAndIssued와 choice 관계
	증적의 식별번호 (serialNo)	INTEGER	m	m	
	opRequesterInfo	OperationRequesterInfo	m	m	CHOICE
	전자문서 서비스 요청자 (requester)	GeneralNames (②)	o	m	null과 choice 관계
	(requester의 generalNames와 동일한 형식으로 중략)				
	null	NULL	o	m	
	서비스 요청 시각 (opRequestTime)	GeneralizedTime	m	m	
	서비스 제공 시각 (opTime)	GeneralizedTime	m	m	
	증명서 종류 (opType)	OperationType	m	m	ENUMERATED (증명요청서의 opType과 동일)
	orgDocInfo	PackageDocumentInfo	m	m	
	패키지 식별자 (packageID)	PackageIdentifier	m	m	UTF8String
	docInfo	DocumentInfo	m	m	
	전자문서 식별자 (docID)	DocumentIdentifier	m	m	UTF8String
	첨부파일 식별자 (fileIDs)	Sequence Size(1..MAX) Of FileIdentifier	o	m	UTF8String
	docHash	DocumentHash	o	m	
	해시알고리즘 식별자 (hashAlg)	HashAlgorithm	m	m	AlgorithmIdentifier (2.16.840.1.101.3.4.2.*)
	전자문서 해시값 (hashedDocument)	BIT STRING	m	m	
	docContentInfo	DocContentInfo (③)	o	m	

번호	필드명	TYPE	생성	처리	비고
	본제목 (title)	BMPString	o	m	
	키워드 (keyword)	BMPString	o	m	
	내용설명 (description)	BMPString	o	m	
	발급 문서 정보 (issuedDocInfo)	PackageDocumentInfo	o	m	
	(PackageDocumentInfo 구조체 형식으로 발급된 전자문서 정보로 중략)				
	peerARCInfo	PeerARCInfo	o	m	
	상대 센터 정보 (peerARC)	GeneralNames (②)	m	m	
	상대 정보 (peerARCPackageID)	PackageIdentifier	m	m	
	작업 발생 사유 (reason)	Reason	o	m	BIT STRING
	orgAndIssued	OriginalAndIssuedDocumentInfo	o	m	
	전자문서 정보 (orgDocInfo)	PackageDocumentInfo	m	m	
	(PackageDocumentInfo 구조체 형식으로 발급된 전자문서 정보로 중략)				
	발급 문서 정보 (issuedDocInfo)	PackageDocumentInfo	m	m	
	(PackageDocumentInfo 구조체 형식으로 발급된 전자문서 정보로 중략)				
	원본/불변경증명서 구분자 (issuedDocOriginal)	BOOLEAN	m	m	원본증명서 TRUE 불변경증명서 FALSE
	증명요청서 해시값 (dataHash)	HashedDataInfo (④)	m	m	

① v1인 경우 version 필드 생략

② 증명요청서 기본필드인 requester와 동일한 형식임

③ 증명요청서 확장필드인 DocContentInfoFlag 값을 반영하여 생성되어야 함

④ 증명요청서 기본필드인 target 하위의 targetHash와 동일한 형식임

5.2.2. 확장 필드

번호	필드명	TYPE	C	생성	처리	비고
1	Qualifications	Sequence Size(1..MAX) Of Qualification (①)	t	o	m	id-kiec-qualification
2	UsageType	BIT STRING (②)	f	o	o	id-kiec-usageType
3	CertifiedTime	GeneralizedTime (③)	t	m (④)	m	id-kiec-certifiedTime
4	CertUsage	BMPString	t/f (⑤)	m/o (⑥)	m/o (⑤)	
5	docContentInfo	DocContentInfo	t/f (⑦)	o	m	
	본제목 (title)	BMPString		o	m	
	키워드 (keyword)	BMPString		o	m	
	내용설명 (description)	BMPString		o	m	

① 증명요청서 확장필드인 Qualifications과 동일한 형식임

② 증명요청서 확장필드인 UsageType과 동일한 형식임

③ 증명요청서 확장필드인 CertifiedTime과 동일한 형식임

④ 등록증명서인 경우 반드시 생성하여야 함(타 증명서에서는 생성 불가)

⑤ 증명요청서 확장필드인 CertUsage의 critical 값을 동일하게 적용하며 TRUE인 경우 반드시 처리되어야 함

⑥ 증명요청서 확장필드인 CertUsage의 critical 값이 TRUE이면 반드시 생성

⑦ 증명요청서의 DocContentInfoFlag 확장필드에 설정된 값에 따라 생성

6. JSON Encoding Rules(JER)

6.1. JER이란?

JER은 JSON 문법을 준수하는 ASN.1의 인코딩 규칙을 말합니다. JER의 상세 규칙 및 사양은 ITU-T Recommendation X.697(02/21)을 따른다.

6.2. JER 인코딩 규칙

6.2.1. JSON 표기법

각 필드명은 임의로 작성한 것으로 value와 length 부분을 제외하고는 이용자가 임으로 작성해도 된다.

번호	변수	예시
1	OBJECT IDENTIFIER	"OJIF":"1.2.3.4.5“
2	BIT STRING	"BITSTR":{ "value":"ED", "length":8}
3	GeneralizedTime(UTC)	"GENTIME":"20220901091010Z“
4	GeneralizedTime(LocalTime)	"GENTIME":"20220901091010“
5	ENUMERATED	"ENUM":"select1“
6	UTF8STRING, IA5String, BMPString	"UTF8STR":"hello"

6.3. JSON 인코딩 예시

6.3.1. CMS 메시지 예시

```
{
  “SignedData“:[
```

```
    {
      “version”:4.1
      “digestAlgorithms”:"2.16.840.1.101.3.4.2.1",
      “encapContentInfo”:{ “실제 증명요청서 내용”},
      “certificates”:{“요청자의 전자서명 내용”},
    }]
  }
```

6.3.2. 증명 요청서

6.3.2.1. 등록증명서, 발급증명서, 이관증명서, 폐기증명서 요청 예시

```
  {
    "version":1,
    "requester":
    {
      "generalNames":[
      {
        "otherName":
        {
          "type-id":"1.2.410.200004.10.1.1",
          "value":[
          {
            "realName":"KISA",
            "userinfo":[
            {
              "type":"1.2.410.200032.2.4.1",
              "value":[
              {
                "hashAlg":"2.16.840.1.101.3.4.2.1",
                "HashedIDN":
```

```
                    {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
                        "length":272
                    }
                }]
            }]
        }]
    }]
  },
  "requestTime":"20220801090000Z",
  "policy":[
  {
    "policyIdentifier":"1.2.410.200032.1.16",
    "policyQualifiers":"NULL"
  }],
  "target":[
  {
    "serialNo":1,
    "opType":0
  }],
  "nonce":1234
}
```


6.3.2.2. 시점확인증명서 요청 예시


```
{
  "version":2,
```

```
"requester":
{
   "generalNames":[
   {
      "otherName":
      {
         "type-id":"1.2.410.200004.10.1.1",
         "value":[
         {
            "realName":"KISA",
            "userinfo":[
            {
               "type":"1.2.410.200032.2.4.1",
               "value":[
               {
                  "hashAlg":"2.16.840.1.101.3.4.2.1",
                  "HashedIDN":
                  {
"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
                     "length":272
                  }
               }]
            }]
         }]
      }
   },
```

```
    "requestTime":"20220801090000Z",
    "policy":[
    {
      "policyIdentifier":"1.2.410.200032.1.16",
      "policyQualifiers":"NULL"
    }],
    "target":[
    {
      "targetHash":
      {
        "hashAlg:"2.16.840.1.101.3.4.2.1",
        "hashedData":
        {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
          "length":272
        }
      }
    }],
    "nonce":1234
  }
```


6.3.2.3. 원본증명서, 불변경증명서 요청 예시


```
  {
    "version":2,
    "requester":
    {
      "generalNames":[
```

```
        {
          "otherName":
          {
            "type-id":"1.2.410.200004.10.1.1",
            "value":[
            {
              "realName":"KISA",
              "userinfo":[
              {
                "type":"1.2.410.200032.2.4.1",
                "value":[
                {
                  "hashAlg":"2.16.840.1.101.3.4.2.1",
                  "HashedIDN":
                  {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
                    "length":272
                  }
                }]
              }]
            }
          }]
        },
        "requestTime":"20220801090000Z",
        "policy":[
        {
```

```
        "policyIdentifier":"1.2.410.200032.1.16",
        "policyQualifiers":"NULL"
    }],
    "target":[
    {
        "packageID":"tempDoc1",
        "docID":"ori_TempDoc1",
        "fileIDs":["docFile1","docFile3"],
        "issuedDocOriginal":"TRUE"  -- 원본증명서 "TRUE", 불변경증명서"FALSE"
    }],
    "nonce":1234
}
```

6.3.2.4. 확장 필드

```
{
    "Extensions":[
    {
        "extnID":"1.2.410.200004.10.1.1",
        "citical":"TRUE",
        "extnValue":
        {
            "Qualifications":[
            {
                "nomineeInfo":
                {
                    "nominee":
                    {
                        "generalNames":[
                        {
```

```
                "otherName":
                {
                   "type-id":"1.2.410.200004.10.1.1",
                   "value":[
                   {
                      "realName":"KISA",
                      "userinfo":[
                      {
                         "type":"1.2.410.200032.2.4.1",
                         "value":[
                         {
                               "hashAlg":"2.16.840.1.101.3.4.2.1",
                               "HashedIDN":
                            {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
                                  "length":272
                            }
                         }]
                      }]
                   }]
                },
                "nomineeCert":
                {
                   "issuerAndSerialNumber":[
                   {
```

```
                    "issuer":"KISA",
                    "serialNumber":1234567899
                }]
            }
        },
        "nomineeRole":
        {
            "value":"00",
            "length":1
        }
    }],
    "DateofExpiration":"20251201090000Z",
    "CertifiedTime":"20220901090000Z",
    "CertUsage":"출력용",
    "DocContentInfoFlag":
    {
        "value":"E0",
        "length":3
    }
  }
 }]
}
```


6.3.3. 증명서

6.3.3.1. 등록증명서, 발급증명서, 이관증명서, 폐기증명서


```
{
  "version":2,
  "serialNumber":1,
```

```
"issuer":
{
   "generalNames":[
   {
      "otherName":
      {
         "type-id":"1.2.410.200004.10.1.1",
         "value":[
         {
            "realName":"KISA",
            "userinfo":[
            {
               "type":"1.2.410.200032.2.4.1",
               "value":[
               {
                  "hashAlg":"2.16.840.1.101.3.4.2.1",
                  "HashedIDN":
                  {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
                     "length":272
                  }
               }]
            }]
         }]
      }
   },
```

```
"dateOfIssue":"20220801090000Z",
"dateOfExpiration":
{
  "dateOfExpiration":"20271201090000Z"
},
"policy":[
{
  "policyIdentifier":"1.2.410.200032.2.3.2",
  "policyQualifiers":[
  {
    "policyQualifierId":"1.",
    "qualifier":
    {
      "cPSuri":"www.kisa.or.kr",
      "userNotice":[
      {
        "explicitText":"정책에 대한 설명"
      }]
    }
  }]
}],
"requestIinfo":
{
  -- 증명요청서가 존재하는 증명서는 증명요청서의 ARCCertRequest 부분 삽입 --
},
"target":
{
  "opRecord":[
```

```
{
  "serialNo":1,
  "opRequesterInfo":[ -- 요청자의 requester 필드와 동일한 값 --
  {
    "generalNames":[
    {
      "otherName":
      {
        "type-id":"1.2.410.200004.10.1.1",
        "value":[
        {
          "realName":"KISA",
          "userinfo":[
          {
            "type":"1.2.410.200032.2.4.1",
            "value":[
            {
              "hashAlg":"2.16.840.1.101.3.4.2.1",
              "HashedIDN":
              {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
                "length":272
              }
            }]
          }]
        }]
      }
```

```
    }]
  }],
  "opRequestTime":"20220801091000Z",
  "opTime":"20220801091100Z",
  "opType":0,
  "orgDocInfo":[
  {
    "packageID":"패키지 식별번호",
    "docInfo":[
    {
      "docID":"원본 전자문서 식별자",
      "fileIDs":["첨부파일 식별자","첨부파일 식별자2"],
      "docHash":[
      {
        "hashAlg":"2.16.840.1.101.3.4.2.1",
        "hashedDocument":
        {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
          "length":272
        }
      }]
    }],
    "docContentInfo":[
    {
      "title":"부동산 계약서",
      "keyword":"real estate",
      "description":"부동산 임대차 계약서입니다."
```

```
    },
    {
      "title":"입금 확인서",
      "keyword":"deposit",
      "description":"은행 입금 확인서입니다."
    }]
  }],
  "issuedDocInfo":[
  {
    "packageID":"패키지 식별번호",
    "docInfo":[
    {
      "docID":"원본 전자문서 식별자",
      "fileIDs":["첨부파일 식별자"],
      "docHash":[
      {
        "hashAlg":"2.16.840.1.101.3.4.2.1",
        "hashedDocument":
        {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
          "length":272
        }
      }]
    }],
    "docContentInfo":[
    {
      "title":"부동산 계약서",
```

```
            "keyword":"real estate",
            "description":"부동산 임대차 계약서입니다."
          }]
        }],
        "peerARCInfo":"NULL",  -- 이수관 시에만 사용 --
        "reason":
        {
          "value":"80", -- 1000 0000(userRequest)의 HEX 값 --
          "length":1
        }
      }],
      "orgAndIssued":[
      {
        "orgDocInfo":[
        {
          "packageID":"패키지 식별번호",
          "docInfo":[
          {
            "docID":"원본 전자문서 식별자",
            "docHash":[
            {
              "hashAlg":"2.16.840.1.101.3.4.2.1",
              "hashedDocument":
              {
"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
                "length":272
              }
```

```
        }]
      }],
      "docContentInfo":[
      {
        "title":"부동산 계약서",
        "keyword":"real estate",
        "description":"부동산 임대차 계약서입니다."
      },
      {
        "title":"입금 확인서",
        "keyword":"deposit",
        "description":"은행 입금 확인서입니다."
      }]
    }],
    "issuedDocInfo":[
    {
      "packageID":"패키지 식별번호",
      "fileIDs":["첨부파일 식별자"],
      "docInfo":[
      {
        "docID":"원본 전자문서 식별자",
        "docHash":[
        {
          "hashAlg":"2.16.840.1.101.3.4.2.1",
          "hashedDocument":
          {
```

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",

```
                    "length":272
                }
            }]
        }],
        "docContentInfo":[
        {
            "title":"부동산 계약서",
            "keyword":"real estate",
            "description":"부동산 임대차 계약서입니다."
        }]
      }],
      "issuedDocOriginal":"NULL"
    }],
    "dataHash":
    {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
        "length":272
    }
  }
}
```


6.3.3.2. 시점확인증명서


```
{
  "version":2,
  "serialNumber":1,
  "issuer":
  {
```

```
        "generalNames":[
        {
          "otherName":
          {
            "type-id":"1.2.410.200004.10.1.1",
            "value":[
            {
              "realName":"KISA",
              "userinfo":[
              {
                "type":"1.2.410.200032.2.4.1",
                "value":[
                {
                  "hashAlg":"2.16.840.1.101.3.4.2.1",
                  "HashedIDN":
                  {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
                    "length":272
                  }
                }]
              }]
            }]
          }
        },
        "dateOfIssue":"20220801090000Z",
        "dateOfExpiration":"NULL",
```

```
"policy":[
{
  "policyIdentifier":"1.2.410.200032.2.3.2",
  "policyQualifiers":[
  {
    "policyQualifierId":"1.",
    "qualifier":
    {
      "cPSuri":"www.kisa.or.kr",
      "userNotice":[
      {
        "explicitText":"정책에 대한 설명"
      }]
    }
  }]
}],
"requestIinfo":
{
 -- 증명요청서가 존재하는 증명서는 증명요청서의 ARCCertRequest 부분 삽입
--
},
"target":
{
  "opRecord":[
  {
    "serialNo":1,
    "opRequesterInfo":[ -- 요청자의 requester 필드와 동일한 값 --
    {
      "generalNames":[
```

```
{
   "otherName":
   {
      "type-id":"1.2.410.200004.10.1.1",
      "value":[
      {
         "realName":"KISA",
         "userinfo":[
         {
            "type":"1.2.410.200032.2.4.1",
            "value":[
            {
               "hashAlg":"2.16.840.1.101.3.4.2.1",
               "HashedIDN":
               {
"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
                  "length":272
               }
            }]
         }]
      }
   }]
}],
"opRequestTime":"20220801091000Z",
"opTime":"20220801091100Z",
"opType":0,
```

```
"orgDocInfo":[
{
  "packageID":"패키지 식별번호",
  "docInfo":[
  {
    "docID":"원본 전자문서 식별자",
    "fileIDs":["첨부파일 식별자","첨부파일 식별자2"],
    "docHash":[
    {
      "hashAlg":"2.16.840.1.101.3.4.2.1",
      "hashedDocument":
      {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
        "length":272
      }
    }]
  }],
  "docContentInfo":[
  {
    "title":"부동산 계약서",
    "keyword":"real estate",
    "description":"부동산 임대차 계약서입니다."
  },
  {
    "title":"입금 확인서",
    "keyword":"deposit",
    "description":"은행 입금 확인서입니다."
```

```
            }]
        }],
        "issuedDocInfo":["NULL"],
        "peerARCInfo":"NULL", -- 이수관 시에만 사용 --
        "reason":
        {
          "value":"80", -- 1000 0000(userRequest)의 HEX 값 --
          "length":1
        }
      }],
      "dataHash":
      {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
        "length":272
      }
    }
  }
```


6.3.3.3. 원본증명서, 불변경증명서


```
  {
    "version":2,
    "serialNumber":1,
    "issuer":
    {
      "generalNames":[
      {
        "otherName":
```

```
        {
          "type-id":"1.2.410.200004.10.1.1",
          "value":[
          {
            "realName":"KISA",
            "userinfo":[
            {
              "type":"1.2.410.200032.2.4.1",
              "value":[
              {
                "hashAlg":"2.16.840.1.101.3.4.2.1",
                "HashedIDN":
                {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE956F4A40F",
                  "length":272
                }
              }]
            }]
          }]
      },
      "dateOfIssue":"20220801090000Z",
      "dateOfExpiration":
      {
        "dateOfExpiration":"20271201090000Z"
      },
```

```
"policy":[
{
   "policyIdentifier":"1.2.410.200032.2.3.2",
   "policyQualifiers":[
   {
      "policyQualifierId":"1.",
      "qualifier":
      {
         "cPSuri":"www.kisa.or.kr",
         "userNotice":[
         {
            "explicitText":"정책에 대한 설명"
         }]
      }
   }]
}],
"requestIinfo":
{
-- 증명요청서가 존재하는 증명서는 증명요청서의 ARCCertRequest 부분 삽입 --
},
"target":
{
   "orgAndIssued":[
   {
      "orgDocInfo":[
      {
         "packageID":"패키지 식별번호",
         "docInfo":[
         {
```

```
            "docID":"원본 전자문서 식별자",
            "docHash":[
            {
              "hashAlg":"2.16.840.1.101.3.4.2.1",
              "hashedDocument":
              {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
                  "length":272
              }
            }]
          }],
          "docContentInfo":[
          {
            "title":"부동산 계약서",
            "keyword":"real estate",
            "description":"부동산 임대차 계약서입니다."
          },
          {
            "title":"입금 확인서",
            "keyword":"deposit",
            "description":"은행 입금 확인서입니다."
          }]
        }],
        "issuedDocInfo":[
        {
          "packageID":"패키지 식별번호",
          "fileIDs":["첨부파일 식별자"],
```

```
            "docInfo":[
            {
              "docID":"원본 전자문서 식별자",
              "docHash":[
              {
                "hashAlg":"2.16.840.1.101.3.4.2.1",
                "hashedDocument":
                {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
                    "length":272
                }
              }]
            }],
            "docContentInfo":[
            {
              "title":"부동산 계약서",
              "keyword":"real estate",
              "description":"부동산 임대차 계약서입니다."
            }]
          }],
          "issuedDocOriginal":"TRUE"
        }]
      },
      "dataHash":
      {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
```

```
        "length":272
    }
}
```

6.3.3.4. 확장필드

```
{
  “Extensions”:[
  {
    “extnID”:“1.2.410.200004.10.1.1",
    “cririca l”:“TRUE”,
    “extnValue”:[
    {
      "Qualifications":[
      {
        "nomineeInfo":
        {
          "nominee":
          {
            "generalNames":[
            {
              "otherName":
              {
                "type-id":"1.2.410.200004.10.1.1",
                "value":[
                {
                  "realName":"KISA",
                  "userinfo":[
                  {
                    "type":"1.2.410.200032.2.4.1",
```

```
                    "value":[
                    {
                        "hashAlg":"2.16.840.1.101.3.4.2.1",
                        "HashedIDN":
                      {

"value":"042088919E474A23E6FD5C4B8275A0CBF2472E5462A772C7ECDB16C67AE9
56F4A40F",
                            "length":272
                      }
                    }]
                  }]
                }]
              }
            }]
          },
          "nomineeCert":
          {
            "issuerAndSerialNumber":[
            {
              "issuer":"KISA",
              "serialNumber":1234567899
            }]
          }
        },
        "nomineeRole":
        {
          "value":"00",
          "length":1
```

```
            }
        }],
        "CertifiedTime":"20240901090000Z",
        "CertUsage":"출력용"
    }
  }]
}
```


7. XML Encoding Rules(XER)

7.1. XER이란?

XER은 XML 문법을 준수하는 ASN.1의 인코딩 규칙을 말합니다. XER의 상세 규칙 및 사양은 ITU-T Recommendation X.693(02/21)을 따른다.

7.2. XER 인코딩 규칙

7.2.1. XML 표기법

번호	변수	예시
1	OBJECT IDENTIFIER	<OJIF>1.2.3.4.5</OJIF>
2	BIT STRING	<BITSTR>11101101</BITSTR>
3	GeneralizedTime(UTC)	<GENTIME>20220901091010Z</GENTIME>
4	GeneralizedTime(LocalTime)	<GENTIME>20220901091010</GENTIME>
5	ENUMERATED	<ENUM><select1/></ENUM>
6	UTF8STRING, IA5String, BMPString	<UTF8STR>hello</UTF8STR>

7.3. XER 인코딩 예시

7.3.1. CMS 예시

```
<SignedData>
  <version>4.1</version>
  <digestAlgorithms>2.16.840.1.101.3.4.2.1</digestAlgorithms>
  <encapContentInfo>“실제 증명요청서 내용”</encapContentInfo>
  <certificates>“요청자의 전자서명 내용”</certificates>
</SignedData>
```

7.3.2. 증명 요청서

7.3.2.1. 등록증명서, 발급증명서, 이관증명서, 폐기증명서 요청 예시

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCCertRequest>
  <version>1</version>
  <requester>
    <generalNames>
      <otherName>
        <type-id>1.2.410.200004.10.1.1</type-id>
        <value>
          <realName>KISA</realName>
          <userinfo>
            <type>1.2.410.200032.2.4.1</type>
            <value>
              <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>10000100000100010001001000110011110010001110100101001000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
```

```
000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000</HashedIDN>
                    </value>
                  </userinfo>
                </value>
              </otherName>
            </generalNames>
          </requester>
          <requestTime>20220801090000Z</requestTime>
          <policy>
            <policyIdentifier>1.2.410.200032.1.16</policyIdentifier>
            <policyQualifiers>NULL</policyQualifiers>
          </policy>
          <target>
            <serialNo>1</serialNo>
            <opType>0</opType>
          </target>
          <nonce>1234</nonce>
        </ARCCertRequest>
```


7.3.2.2. 시점확인증명서 요청 예시


```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCCertRequest>
  <version>2</version>
  <requester>
    <generalNames>
      <otherName>
        <type-id>1.2.410.200004.10.1.1</type-id>
        <value>
```

```
                &lt;realName&gt;KISA&lt;/realName&gt;
                &lt;userinfo&gt;
                    &lt;type&gt;1.2.410.200032.2.4.1&lt;/type&gt;
                    &lt;value&gt;
                        &lt;hashAlg&gt;2.16.840.1.101.3.4.2.1&lt;/hashAlg&gt;

&lt;HashedIDN&gt;1000010000010001000100100011001111001000111010010100100000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000&lt;/HashedIDN&gt;
                    &lt;/value&gt;
                &lt;/userinfo&gt;
            &lt;/value&gt;
        &lt;/otherName&gt;
    &lt;/generalNames&gt;
&lt;/requester&gt;
&lt;requestTime&gt;20220801090000Z&lt;/requestTime&gt;
&lt;policy&gt;
    &lt;policyIdentifier&gt;1.2.410.200032.1.16&lt;/policyIdentifier&gt;
    &lt;policyQualifiers&gt;NULL&lt;/policyQualifiers&gt;
&lt;/policy&gt;
&lt;target&gt;
    &lt;targetHash&gt;
        &lt;hashAlg&gt;2.16.840.1.101.3.4.2.1&lt;/hashAlg&gt;

&lt;hashedData&gt;1000010000010001000100100011001111001000111010010100100000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000&lt;/hashedData&gt;
    &lt;/targetHash&gt;
&lt;/target&gt;
```

```
    <nonce>1234</nonce>
  </ARCCertRequest>
```

7.3.2.3. 원본증명서, 불변경증명서 요청 예시

```
  <?xml version="1.0" encoding="UTF-8" ?>
  <ARCCertRequest>
    <version>2</version>
    <requester>
      <generalNames>
        <otherName>
          <type-id>1.2.410.200004.10.1.1</type-id>
          <value>
            <realName>KISA</realName>
            <userinfo>
              <type>1.2.410.200032.2.4.1</type>
              <value>
                <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>10000100000100010001001000110011110010001110100101001000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000</HashedIDN>
              </value>
            </userinfo>
          </value>
        </otherName>
      </generalNames>
    </requester>
    <requestTime>20220801090000Z</requestTime>
    <policy>
```

```
    <policyIdentifier>1.2.410.200032.1.16</policyIdentifier>
    <policyQualifiers>NULL</policyQualifiers>
  </policy>
  <target>
    <packageID>tempDoc1</packageID>
    <docID>ori_TempDoc1</docID>
    <fileIDs>docFile1</fileIDs>
    <fileIDs>docFile3</fileIDs>
    <issuedDocOriginal>TRUE</issuedDocOriginal>
  </target>
  <nonce>1234</nonce>
</ARCCertRequest>
```

7.3.2.4. 확장 필드

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCCertRequest>
  <Extensions>
    <extnID>1.2.410.200004.10.1.1</extnID>
    <citical>TRUE</citical>
    <extnValue>
      <Qualifications>
        <nomineeInfo>
          <nominee>
            <generalNames>
              <otherName>
                <type-id>1.2.410.200004.10.1.1</type-id>
                <value>
                  <realName>KISA</realName>
                  <userinfo>
```

<type>1.2.410.200032.2.4.1</type>
<value>
<hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>100001000000100010001001000110011110010001110100101001000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000</HashedIDN>
</value>
</userinfo>
</value>
</otherName>
</generalNames>
</nominee>
<nomineeCert>
<issuerAndSerialNumber>
<issuer>KISA</issuer>
<serialNumber>1234567899</serialNumber>
</issuerAndSerialNumber>
</nomineeCert>
</nomineeInfo>
<nomineeRole>10000000</nomineeRole>
</Qualifications>
<DateofExpiration>20251201090000Z</DateofExpiration>
<CertifiedTime>20220901090000Z</CertifiedTime>
<CertUsage>출력용</CertUsage>
<DocContentInfoFlag>11100000</DocContentInfoFlag>
</extnValue>
</Extensions>
</ARCCertRequest>

7.3.3. 증명서

7.3.3.1. 등록증명서, 발급증명서, 이관증명서, 폐기증명서


```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ARCCertRequest>
  <version>2</version>
  <serialNumber>1</serialNumber>
  <issuer>
    <generalNames>
      <otherName>
        <type-id>1.2.410.200004.10.1.1</type-id>
        <value>
          <realName>KISA</realName>
          <userinfo>
            <type>1.2.410.200032.2.4.1</type>
            <value>
              <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>1000010000001000100010010001100111100100011101001010010000000000
0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000</HashedIDN>
            </value>
          </userinfo>
        </value>
      </otherName>
    </generalNames>
  </issuer>
  <dateOfIssue>20220801090000Z</dateOfIssue>
  <dateOfExpiration>
```

<dateOfExpiration>20271201090000Z</dateOfExpiration>
</dateOfExpiration>
<policy>
<policyIdentifier>1.2.410.200032.2.3.2</policyIdentifier>
<policyQualifiers>
<policyQualifierId>1.</policyQualifierId>
<qualifier>
<cPSuri>www.kisa.or.kr</cPSuri>
<userNotice>
<explicitText>정책에 대한 설명</explicitText>
</userNotice>
</qualifier>
</policyQualifiers>
</policy>
<requestIinfo/>
<target>
<opRecord>
<serialNo>1</serialNo>
<opRequesterInfo>
<generalNames>
<otherName>
<type-id>1.2.410.200004.10.1.1</type-id>
<value>
<realName>KISA</realName>
<userinfo>
<type>1.2.410.200032.2.4.1</type>
<value>
<hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>100001000000100010001001000110011110010001110100101001000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000</HashedIDN>

</value>

</userinfo>

</value>

</otherName>

</generalNames>

</opRequesterInfo>

<opRequestTime>20220801091000Z</opRequestTime>

<opTime>20220801091100Z</opTime>

<opType>0</opType>

<orgDocInfo>

<packageID>패키지 식별번호</packageID>

<docInfo>

<docID>원본 전자문서 식별자</docID>

<fileIDs>첨부파일 식별자</fileIDs>

<fileIDs>첨부파일 식별자2</fileIDs>

<docHash>

<hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<hashedDocument>1000010000001000100010010001100111100100011101001010010000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000</hashedDocum
ent>

</docHash>

</docInfo>

<docContentInfo>

<title>부동산 계약서</title>

```
      <keyword>real estate</keyword>
      <description>부동산 임대차 계약서입니다.</description>
    </docContentInfo>
    <docContentInfo>
      <title>입금 확인서</title>
      <keyword>deposit</keyword>
      <description>은행 입금 확인서입니다.</description>
    </docContentInfo>
  </orgDocInfo>
  <issuedDocInfo>
    <packageID>패키지 식별번호</packageID>
    <docInfo>
      <docID>원본 전자문서 식별자</docID>
      <fileIDs>첨부파일 식별자</fileIDs>
      <docHash>
        <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<hashedDocument>100001000001000100010010001100111100100011101001010010000
000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000</hashedDocum
ent>
      </docHash>
    </docInfo>
    <docContentInfo>
      <title>부동산 계약서</title>
      <keyword>real estate</keyword>
      <description>부동산 임대차 계약서입니다.</description>
    </docContentInfo>
  </issuedDocInfo>
```

```
    <peerARCInfo>NULL</peerARCInfo>
    <reason>10000000</reason>
  </opRecord>
  <orgAndIssued>
    <orgDocInfo>
      <packageID>패키지 식별번호</packageID>
      <docInfo>
        <docID>원본 전자문서 식별자</docID>
        <docHash>
          <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<hashedDocument>100001000001000100010010001100111100100011101001010010000
000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000</hashedDocum
ent>
        </docHash>
      </docInfo>
      <docContentInfo>
        <title>부동산 계약서</title>
        <keyword>real estate</keyword>
        <description>부동산 임대차 계약서입니다.</description>
      </docContentInfo>
      <docContentInfo>
        <title>입금 확인서</title>
        <keyword>deposit</keyword>
        <description>은행 입금 확인서입니다.</description>
      </docContentInfo>
    </orgDocInfo>
    <issuedDocInfo>
```

<packageID>패키지 식별번호</packageID>
<fileIDs>첨부파일 식별자</fileIDs>
<docInfo>
<docID>원본 전자문서 식별자</docID>
<docHash>
<hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<hashedDocument>100001000001000100010010001100111100100011101001010010000
000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000</hashedDocum
ent>
</docHash>
</docInfo>
<docContentInfo>
<title>부동산 계약서</title>
<keyword>real estate</keyword>
<description>부동산 임대차 계약서입니다.</description>
</docContentInfo>
</issuedDocInfo>
<issuedDocOriginal>NULL</issuedDocOriginal>
</orgAndIssued>

<dataHash>1000010000010001000100100011001111001000111010010100100000000000
000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000</dataHash>
</target>
</ARCCertRequest>

7.3.3.2. 시점확인증명서

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCCertRequest>
  <version>2</version>
  <serialNumber>1</serialNumber>
  <issuer>
    <generalNames>
      <otherName>
        <type-id>1.2.410.200004.10.1.1</type-id>
        <value>
          <realName>KISA</realName>
          <userinfo>
            <type>1.2.410.200032.2.4.1</type>
            <value>
              <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>10000100000100010001001000110011110010001110100101001000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000</HashedIDN>
            </value>
          </userinfo>
        </value>
      </otherName>
    </generalNames>
  </issuer>
  <dateOfIssue>20220801090000Z</dateOfIssue>
  <dateOfExpiration>NULL</dateOfExpiration>
  <policy>
    <policyIdentifier>1.2.410.200032.2.3.2</policyIdentifier>
```

<policyQualifiers>
<policyQualifierId>1.</policyQualifierId>
<qualifier>
<cPSuri>www.kisa.or.kr</cPSuri>
<userNotice>
<explicitText>정책에 대한 설명</explicitText>
</userNotice>
</qualifier>
</policyQualifiers>
</policy>
<requestIinfo/>
<target>
<opRecord>
<serialNo>1</serialNo>
<opRequesterInfo>
<generalNames>
<otherName>
<type-id>1.2.410.200004.10.1.1</type-id>
<value>
<realName>KISA</realName>
<userinfo>
<type>1.2.410.200032.2.4.1</type>
<value>
<hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>100001000000100010001001000110011110010001110100101001000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000</HashedIDN>

</value>

```
            </userinfo>
          </value>
        </otherName>
      </generalNames>
    </opRequesterInfo>
    <opRequestTime>20220801091000Z</opRequestTime>
    <opTime>20220801091100Z</opTime>
    <opType>0</opType>
    <orgDocInfo>
      <packageID>패키지 식별번호</packageID>
      <docInfo>
        <docID>원본 전자문서 식별자</docID>
        <fileIDs>첨부파일 식별자</fileIDs>
        <fileIDs>첨부파일 식별자2</fileIDs>
        <docHash>
          <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<hashedDocument>10000100000100010001001000110011110010001110100101001000
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000</hashedDocum
ent>
        </docHash>
      </docInfo>
      <docContentInfo>
        <title>부동산 계약서</title>
        <keyword>real estate</keyword>
        <description>부동산 임대차 계약서입니다.</description>
      </docContentInfo>
      <docContentInfo>
```

```
            <title>입금 확인서</title>
            <keyword>deposit</keyword>
            <description>은행 입금 확인서입니다.</description>
          </docContentInfo>
        </orgDocInfo>
        <issuedDocInfo>NULL</issuedDocInfo>
        <peerARCInfo>NULL</peerARCInfo>
        <reason>10000000</reason>
      </opRecord>

<dataHash>10000100000100010001001000110011111001000111010010100100000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000</dataHash>
    </target>
  </ARCCertRequest>
```

7.3.3.3. 원본증명서, 불변경증명서

```
  <?xml version="1.0" encoding="UTF-8" ?>
  <ARCCertRequest>
    <version>2</version>
    <serialNumber>1</serialNumber>
    <issuer>
      <generalNames>
        <otherName>
          <type-id>1.2.410.200004.10.1.1</type-id>
          <value>
            <realName>KISA</realName>
            <userinfo>
              <type>1.2.410.200032.2.4.1</type>
```

<value>
<hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>10000100000010001000100100011001111100100011101001010010000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000</HashedIDN>
</value>
</userinfo>
</value>
</otherName>
</generalNames>
</issuer>
<dateOfIssue>20220801090000Z</dateOfIssue>
<dateOfExpiration>
<dateOfExpiration>20271201090000Z</dateOfExpiration>
</dateOfExpiration>
<policy>
<policyIdentifier>1.2.410.200032.2.3.2</policyIdentifier>
<policyQualifiers>
<policyQualifierId>1.</policyQualifierId>
<qualifier>
<cPSuri>www.kisa.or.kr</cPSuri>
<userNotice>
<explicitText>정책에 대한 설명</explicitText>
</userNotice>
</qualifier>
</policyQualifiers>
</policy>
<requestIinfo/>

```
<target>
  <orgAndIssued>
    <orgDocInfo>
      <packageID>패키지 식별번호</packageID>
      <docInfo>
        <docID>원본 전자문서 식별자</docID>
        <docHash>
          <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<hashedDocument>100001000001000100010010001100111100100011101001010010000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000</hashedDocum
ent>
        </docHash>
      </docInfo>
      <docContentInfo>
        <title>부동산 계약서</title>
        <keyword>real estate</keyword>
        <description>부동산 임대차 계약서입니다.</description>
      </docContentInfo>
      <docContentInfo>
        <title>입금 확인서</title>
        <keyword>deposit</keyword>
        <description>은행 입금 확인서입니다.</description>
      </docContentInfo>
    </orgDocInfo>
    <issuedDocInfo>
      <packageID>패키지 식별번호</packageID>
      <fileIDs>첨부파일 식별자</fileIDs>
```

<docInfo>

<docID>원본 전자문서 식별자</docID>

<docHash>

<hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<hashedDocument>1000010000010001000100100011001111001000111010010100100</hashedDocument>

</docHash>

</docInfo>

<docContentInfo>

<title>부동산 계약서</title>

<keyword>real estate</keyword>

<description>부동산 임대차 계약서입니다.</description>

</docContentInfo>

</issuedDocInfo>

<issuedDocOriginal>TRUE</issuedDocOriginal>

</orgAndIssued>

</target>

<dataHash>10000100000100010001001000110011110010001110100101001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000</dataHash>

</ARCCertRequest>

7.3.3.4. 확장필드

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCCertRequest>
  <Extensions>
    <extnID>1.2.410.200004.10.1.1</extnID>
    <citical>TRUE</citical>
    <extnValue>
      <Qualifications>
        <nomineeInfo>
          <nominee>
            <generalNames>
              <otherName>
                <type-id>1.2.410.200004.10.1.1</type-id>
                <value>
                  <realName>KISA</realName>
                  <userinfo>
                    <type>1.2.410.200032.2.4.1</type>
                    <value>
                      <hashAlg>2.16.840.1.101.3.4.2.1</hashAlg>

<HashedIDN>10000100000100010001001000110011110010001110100101001000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000</HashedIDN>
                    </value>
                  </userinfo>
                </value>
              </otherName>
            </generalNames>
          </nominee>
```

```
            <nomineeCert>
              <issuerAndSerialNumber>
                <issuer>KISA</issuer>
                <serialNumber>1234567899</serialNumber>
              </issuerAndSerialNumber>
            </nomineeCert>
          </nomineeInfo>
          <nomineeRole>“10000000”</nomineeRole>
        </Qualifications>
        <CertifiedTime>20240901090000Z</CertifiedTime>
        <CertUsage>출력용</CertUsage>
      </extnValue>
    </Extensions>
  </ARCCertRequest>
```

8. JER XER 인코딩 차이

	JER	XER
OBJECT IDENTIFIER	"OJIF":"1.2.3.4.5"	<OJIF>1.2.3.4.5</OJIF>

BIT STRING	"BITSTR":
	{
	"value":"ED",
	"length":8
	}
		<BITSTR>11101101</BITSTR>

GeneralizedTime	- UTC "GENTIME":"20220901091010Z“	- UTC <GENTIME>20220901091010Z</GENTIME>
	- LocalTime "GENTIME":"20220901091010“	- LocalTime <GENTIME>20220901091010</GENTIME>
ENUMERATED	"ENUM":"select1"	<ENUM><select1/></ENUM>
UTF8STRING	"UTF8STR":"hello"	<UTF8STR>hello</UTF8STR>

※ 이외 인코딩 규칙은 각 설명 자료 참조

규 격 연 혁

버전	제 · 개정일	제 · 개정내역
v1.00	2006년 11월 28일	· 제정
v1.10	2007년 8월 27일	· 보증일시(CertifiedTime) 확장필드 관련 내용 추가 · 시점확인증명서 관련 내용 추가 · 모호하게 표현되거나 설명이 부족한 부분을 보완 · 증명서 포맷 검증 에러의 예를 추가 및 수정 · 증명서 검증절차 그림 수정
v1.20	2009년 11월 4일	· 증적의 시각값을 패키지 규격 상의 시각값과 일치시킴 · 증적의 전자문서 해시값 생성 및 검증 시, 암호화된 첨부파일의 경우 암호화 이전 첨부파일에 대한 해시값을 생성하거나 검증하는 내용 추가 · 증명요청서의 nonce값의 길이를 명확하게 제시함 · 기타 버그 수정
v2.00	2011년 12월 30일	· 전자문서 등록 시 최초등록증명서 발급 여부 및 이용자 전달방식 보완 · 증적의 시각값과 전자문서 패키지의 시각값을 일치시키는 내용 중, 전자문서 등록 시 AIP의 RegisterDateTime 필드값을 제외한 다른 항목은 삭제 · 패키지 규격에서 변환본 관련 내용이 삭제됨에 따라, 증적의 하위 필드 중 변환본 관련 필드 및 설명 삭제 및 보완 · 전자문서 수관 시 최초등록증명서 재발급 절차 및 생성 방법 추가
v2.10	2013년 5월 23일	· 규격 용어 현행화
v3.00	2014년 1월 1일	· 시점확인증명요청서의 전자서명을 옵션으로 변경 · 시점확인증명서 요청자 정보 및 효력만기일을 생략가능하도록 수정 · 시점확인증명서 포맷 변경 관련 발급 및 처리 절차 보완

		· 원본증명서 발급요청 시 증명요청서를 생성하여 요청하도록 수정 · 불변경증명서 추가 및 전자문서 발급 및 열람에 따른 증명서 필드 설정 내용 보완 · 증명서에 문서 연관 정보를 기재할 수 있는 필드 추가 · CertUsage 확장필드 추가
v3.01	2017년 7월 18일	· 원본증명서와 전자문서를 함께 종이로 출력하는 경우의 요건 명시
v3.10	2023년 7월 1일	· 증명요청서에 전자서명 선택으로 변경 · JER, XER 형식 증명서 추가 · 증명서의 발급 후 즉시 검증을 삭제 · 시점확인증명서에 전자서명 검증 제거 · AIP, DIP 등 패키지 내용 삭제 · CI/DI 민감정보로 포함 · 증명서 출력 규격 · 증명서 출력 형태에 KISA 마크 삽입 · 시점확인증명서와 동일한 방법으로 전자서명 제외 가능하도록 함 · 증명 요청서 및 증명응답메시지에 확장 필드->버전 정보 추가 · 증명응답메시지에 확장 필드->문서추가정보 추가 · 두 개 이상 전자문서에 대한 해시 방법 추가
3.20	2026년 1월	· 고시에서 제외된 증명서에 대한 권고 처리 · PDF 증명서 형식의 출력 방법 추가