○法務省告示第五百四十三号

商業登記規則(昭和三十九年法務省令第二十三号)第三十三条の六第五項(第三十三条の十四第二項において準用する場合を含む。)及び第六項、第三十三条の八第一項(第三十三条の十五第三項において準用する場合を含む。)及び第二項、第三十三条の十三第二項並びに第三十三条の十五第二項及び第三項の規定(これらの規定を他の省令において準用する場合を含む。)に基づき、法務大臣が指定する電子証明書の方式等を次のように定め、平成二十六年十二月十三日から施行する。なお、平成十二年九月二十九日法務省告示第三百七十二号は、平成二十六年十二月十二日限り、廃止する。

平成二十六年十二月十二日

法務大臣　上川　陽子

第1　電磁的記録への記録方式

1　使用する電磁的記録媒体

(1)　商業登記規則第33条の6第4項第1号の光ディスク(以下「申請用光ディスク」という。)のトラックフォーマットは、産業標準化法(昭和24年法律第185号)に基づく日本産業規格(以下「日本産業規格」という。)X6241又はX6281に適合する直径120ミリメートルの光ディスクの再生装置で再生することが可能なものによる。ボリューム及びファイル構成は、日本産業規格X0606又はX0610による。

(2)　商業登記規則第33条の6第4項第2号の不揮発性半導体記憶装置(以下「申請用メモリ」という。)の構造は、ユーエスビーインプリメンターズフォーラムが定めたUSB1.0、USB1.1、USB2.0又はUSB3.0に適合し、かつ、Standard A端子を備えたものによる。ボリューム及びファイル構成は、File Allocation Table 16、File Allocation Table 32、NT File System又はExtended File Allocation Tableによる。

(3)　1個の申請用光ディスク又は申請用メモリには、1件の申請に係る商業登記規則第33条の6第1項の電磁的記録(以下第2までにおいて「電磁的記録」という。)のみを記録する。

2　ファイル名

ファイル名は、「SHINSEI」とする。

3　ファイルへの記録の方式

(1)　ファイルに所要事項(データ)を格納する際には、次の4に定めるところにより、「データ型」欄に掲げる形式を用いて、付録1の様式に従って、「フィールド」欄に掲げる事項を記録する。この場合において、「データ型」欄の形式は、国際標準化機構が

定めた規格8824-1:1998、8824-2:1998、8824-3:1998、8824-4:1998の抽象構文記法1(以下「ASN.1」という。)及び付録1に定めるところによる。

(2) データの符号化は、国際標準化機構ISO／IEC8825-1:2015の識別符号化規則(以下「DER」という。)による。

(3) フィールド欄の「—」部分は、フィールドが定義されていないことを表す。以下同じ。

4 証明書発行申請ファイル

フィールド	データ型	設定値	必須(注1)
—	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
—	Name(RDNSequence)		○
recipient	GeneralName([4])		
—	Name(RDNSequence)		○
body	PKIBody([0])		
—	CertReqMessages		
—	CertReqMsg		
certReq	CertRequest		
certReqId	INTEGER	0	◎
certTemplate	CertTemplate		

subject	[5]		△ (注2)
—	Name(RDNSequence)		
—	RelativeDistinguishedName		△ (注3)
(organizationName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.10	↑◎ (注2)
value	DirectoryString (UTF8String)	商業登記規則第33条の6第6項の規定により商号等の表音等をローマ字等で表示したものを記録する場合には、記録する。 (注4)	↑◎ (注2)
—	RelativeDistinguishedName		△ (注5)
(commonName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.3	↑◎ (注2)
value	DirectoryString (UTF8String)	商業登記規則第33条の6第6項の規定により氏名の表音をローマ字等で表示したものを記録する場	↑◎ (注2)

		合には、記録する。(注4)	
publicKey	[6] SubjectPublicKeyInfo		
algorithm	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1	◎
parameters	NULL		△
subjectPublicKey	BIT STRING	日本産業規格X5731-8附属書Dに定める方式に従って作成した2,048ビットの公開かぎを記録する。	◎
extensions	[9] Extensions		
(registeredCorporationInfo)	Extension		
extnId	OBJECT IDENTIFIER	1.2.392.100300.1.1.3	◎
extnValue	OCTET STRING		
—	RegisteredCorporationInfoSyntax		
corporateName	[0]		
—	DirectoryString (UTF8String)	「商号」を記録する。(注6)	◎
corporateAddress	[2]		

—	DirectoryString (UTF8String)	「本店(印鑑提出者が商号使用者又は支配人であるときは、それぞれ営業所又は支配人を置いた営業所)」を記録する。(注6)	◎
representativeDirectorName	[3]		
—	DirectoryString (UTF8String)	「印鑑提出者の氏名」を記録する。(注6)	◎
representativeDirectorTitle	[4]		
—	DirectoryString (UTF8String)	「印鑑提出者の資格」を記録する。(注6)	◎
pop	ProofOfPossession ([1] POPOSigningKey)		
algorithmIdentifier	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL		△
signature	BIT STRING	(注7)	◎
regInfo	SEQUENCE OF AttributeTypeAndValue		
(suspensionSecretCode)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	1.2.392.100300.1.2.105	◎

value	SuspensionSecretCode		
hashAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	2.16.840.1.101.3.4.2.1	◎
parameters	NULL		△
hashedSecretCode	OCTET STRING	(注8)	◎
(timeLimit)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	1.2.392.100300.1.2.104	◎
value	TimeLimit	商業登記法第12条の2第1項第2号の期間(月数)を記録する。(注9)	◎

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。「必須」欄中に○印のあるフィールドは、必ず設けなければならない。「必須」欄中に△印のあるフィールドは、記録する内容によって任意に設けることができる。「必須」欄中に↑◎印のあるフィールドには、上欄にフィールドを設けたときは、必ず値を記録しなければならない。

注2　商業登記規則第33条の6第6項の規定により電磁的記録に商号又は氏名等の表音等をローマ字等で記録する場合には、このフィールドを設定する。

注3　商業登記規則第33条の6第6項の規定により商号の表音等をローマ字等で表示したものを記録する場合には、このフィールドを設定する。

注4　商号等の表音をローマ字等で表記する場合の文字数は44文字以内とし、氏名の表音をローマ字等で表記する場合の文字数は50文字以内とする。使用する文字等の範囲は、日本産業規格X0201-1997のラテン文字用図形文字集合及びスペースとし、文字の符号化表現は、日本産業規格X0208-1997附属書1に規定する方式による。

注5　商業登記規則第33条の6第6項の規定により氏名の表音をローマ字等で表示したものを記録する場合には、このフィールドを設定する。

注6　「商号」、「本店(印鑑提出者が商号使用者又は支配人であるときは、それぞれ営業所又は支配人を置いた営業所)」及び「印鑑提出者の資格」の文字数は、各128文字

以内とし、「印鑑提出者の氏名」の文字数は、126文字以内とする。使用する文字等の範囲は、日本産業規格X0208-1997の2バイト図形文字集合とし、この範囲外の文字等は、範囲内の類似の文字等又はその表音を片仮名に置き換えて記録する。文字の符号化表現は、日本産業規格X0208-1997附属書1に規定する方式による。

なお、営業所又は支配人を置いた営業所を記録する場合においては、当該営業所等に係る表示の末尾に、それぞれ「(営業所)」又は「(支配人を置いた営業所)」と続けて記録する。

注7　「certReq」に属する部分をDERにより符号化した値にsha-256WithRSAEncryptionによる電子署名(日本産業規格X5007及びX5603に規定するオブジェクト識別子(以下「オブジェクト識別子」という。)を「1.2.840.113549.1.1.11」とするアルゴリズムに基づき変換する措置をいう(以下「sha-256WithRSAによる電子署名」という。)。)を講じた値を記録する。

注8　商業登記規則第33条の6第5項第4号に規定する申請人が定める識別符号をSHA-256(オブジェクト識別子を「2.16.840.1.101.3.4.2.1」とするアルゴリズムをいう。以下同じ。)により変換した値を記録する。同号の規定により申請人が定める識別符号の長さは、8バイト以上64バイト以下とする。使用する文字等の範囲は、日本産業規格X0201で規定されたラテン文字用図形文字集合とする。文字等の符号化表現は、日本産業規格X0208-1997附属書1に規定する方式による。

注9　使用する数字は、日本産業規格X0201で規定されたラテン文字用図形文字集合の数字とする。この場合において、1桁の数字を記録するときは、最初に0を記録して2桁にしなければならない。文字の符号化表現は、日本産業規格X0208-1997附属書1に規定する方式による。

第2　電子証明書の方式

1　通則

(1)　商業登記規則第33条の8第2項の電子証明書に所要事項(データ)を格納する際には、次の2に定めるところにより、「データ型」欄に掲げる形式を用いて、付録2の様式に従って、「フィールド」欄に掲げる事項を記録する。この場合において、「データ型」欄の形式は、ASN.1及び付録2に定めるところによる。また、電子認証登記所の登記官(以下「登記官」という。)が商業登記規則第33条の8第1項の規定により電子署名を講ずるのに用いる公開かぎを明らかにするため、次の3に定めるところにより作成する登記官の電子証明書についても同様とする。

(2) データの符号化は、DERによる。

2 印鑑提出者の電子証明書

フィールド	データ型	設定値
―	Certificate	
tbsCertificate	TBSCertificate	
version	[0]	
―	Version	2
serialNumber	CertificateSerialNumber	(注1)
signature	AlgorithmIdentifier	
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11
parameters	NULL	(注2)
issuer	Name(RDNSequence)	
―	RelativeDistinguishedName	
(countryName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.6
value	PrintableString	JP
―	RelativeDistinguishedName	
(organizationName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.10

value	DirectoryString (UTF8String)	Japanese Government
—	RelativeDistinguishedName	
(organizationalUnitName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.11
value	DirectoryString (UTF8String)	Ministry of Justice
—	RelativeDistinguishedName	
(commonName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.3
value	DirectoryString (UTF8String)	Registrar of Tokyo Legal Affairs Bureau
validity	Validity	
notBefore	Time(UTCTime)	(注3)
notAfter	Time(UTCTime)	(注4)
subject	Name(RDNSequence)	
—	RelativeDistinguishedName	
(countryName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.6

value	PrintableString	JP
—	RelativeDistinguishedName	
(organizationName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.10
value	DirectoryString (UTF8String)	(注5)
—	RelativeDistinguishedName	
(commonName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.3
value	DirectoryString (UTF8String)	(注6)
subjectPublicKeyInfo	SubjectPublicKeyInfo	
algorithm	AlgorithmIdentifier	
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1
parameters	NULL	(注2)
subjectPublicKey	BIT STRING	(注7)
extensions	[3]	
—	Extensions	
(authorityKeyIdentifier)	Extension	

extnId	OBJECT IDENTIFIER	2.5.29.35
extnValue	OCTET STRING	
—	AuthorityKeyIdentifier	
keyIdentifier	[0] KeyIdentifier	(注8)
authorityCertIssuer	[1] GeneralNames	
—	GeneralName([4])	
—	Name(RDNSequence)	
—	RelativeDistinguishedName	
(countryName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.6
value	PrintableString	JP
—	RelativeDistinguishedName	
(organizationName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.10
value	DirectoryString (UTF8String)	Japanese Government
—	RelativeDistinguishedName	

(organizationalUnitName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.11
value	DirectoryString (UTF8String)	Ministry of Justice
—	RelativeDistinguishedName	
(commonName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.3
value	DirectoryString (UTF8String)	Registrar of Tokyo Legal Affairs Bureau
authorityCertSerialNumber	[2] CertificateSerialNumber	(注9)
(subjectKeyIdentifier)	Extension	
extnId	OBJECT IDENTIFIER	2.5.29.14
extnValue	OCTET STRING	
—	SubjectKeyIdentifier	(注10)
(certificatePolicies)	Extension	
extnId	OBJECT IDENTIFIER	2.5.29.32
extnValue	OCTET STRING	
—	CertificatePoliciesSyntax	

—	PolicyInformation	
policyIdentifier	CertPolicyId	1.2.392.100300.1.3.3(注11)
policyQualifiers	SEQUENCE OF PolicyQualifierInfo	
—	PolicyQualifierInfo	
policyQualifierId	OBJECT IDENTIFIER	1.3.6.1.5.5.7.2.2
qualifier	UserNotice	
noticeRef	NoticeReference	
organization	DisplayText(VisibleString)	Ministry of Justice
noticeNumbers	SEQUENCE OF INTEGER	
—	INTEGER	1
explicitText	DisplayText(VisibleString)	(注12)
(authorityInfoAccess)	Extension	
extnId	OBJECT IDENTIFIER	1.3.6.1.5.5.7.1.1
extnValue	OCTET STRING	
—	AuthorityInfoAccessSyntax	
—	AccessDescription	

accessMethod	OBJECT IDENTIFIER	1.3.6.1.5.5.7.48.1
accessLocation	GeneralName([6] IA5String)	http://crca.moj.go.jp/bin/dcwcgi/DC_HUSR/cert/cert
(jCertificatePolicies)	Extension	
extnId	OBJECT IDENTIFIER	1.2.392.100300.1.1.1
extnValue	OCTET STRING	
—	JCertificatePoliciesSyntax	
—	PolicyInformation	
policyIdentifier	CertPolicyId	1.2.392.100300.1.3.4(注11)
policyQualifiers	SEQUENCE OF PolicyQualifierInfo	
—	PolicyQualifierInfo	
policyQualifierId	OBJECT IDENTIFIER	1.3.6.1.5.5.7.2.2
qualifier	UserNotice	
noticeRef	NoticeReference	
organization	DisplayText(UTF8String)	法務省
noticeNumbers	SEQUENCE OF INTEGER	
—	INTEGER	1
explicitText	DisplayText(UTF8String)	(注13)
(registrar)	Extension	

extnId	OBJECT IDENTIFIER	1.2.392.100300.1.1.2
extnValue	OCTET STRING	
—	RegistrarSyntax(UTF8String)	東京法務局登記官
(registeredCorporationInfo)	Extension	
extnId	OBJECT IDENTIFIER	1.2.392.100300.1.1.3
extnValue	OCTET STRING	
—	RegisteredCorporationInfoSyntax	
corporateName	[0]	
—	DirectoryString(UTF8String)	(注14)
registeredNumber	[1]	
—	PrintableString	(注15)
corporateAddress	[2]	
—	DirectoryString(UTF8String)	(注16)
representativeDirectorName	[3]	
—	DirectoryString(UTF8Stri	(注17)

	ng)	
representativeDirectorTitle	[4]	
—	DirectoryString(UTF8String)	(注18)
registryOffice	[6]	
—	DirectoryString(UTF8String)	(注19)
algorithm	AlgorithmIdentifier	
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11
parameters	NULL	(注2)
signature	BIT STRING	(注20)

注1　商業登記規則第33条の8第2項第2号の電子証明書の番号を記録する。

注2　長さオクテットに「0」を記録する。

注3　商業登記規則第33条の8第2項第3号の電子証明書の作成日時をグリニッジ標準時により記録する。

注4　電子証明書を作成した日の翌日から起算して、法第12条の2第1項第2号の期間の満了する日の日本時間23時59分59秒をグリニッジ標準時により記録する。

注5　「MOJ No.「会社法人等番号」-「商号等の表音をローマ字等で表示したもの」」の形式で記録する。なお、「-「商号等の表音をローマ字等で表示したもの」」は、電磁的記録に記録がある場合に限り、これを記録する。

注6　「「役員番号」-「氏名の表音をローマ字等で表示したもの」」の形式で記録する。なお、「-「氏名の表音をローマ字等で表示したもの」」は、電磁的記録に記録がある場合に限り、これを記録する。

注7　電磁的記録の「subjectPublicKey」に記録された事項を記録する。日本産業規格X5731-8附属書Dに定める方式に従って作成した2,048ビットの公開かぎが記録される。

注8　登記官の電子証明書の「subjectPublicKey」の値(識別子オクテット、長さオクテ

ット及び未使用ビットを除く。)をSHA-1により変換した値を記録する。

注9　登記官の電子証明書の番号を記録する。

注10　電子証明書の「subjectPublicKey」の値(識別子オクテット、長さオクテット及び未使用ビットを除く。)をSHA-1により変換した値を記録する。

注11　法務省ホームページに掲示される電子証明書に関する注意事項等を識別するオブジェクト識別子を記録する。

注12　電子証明書を利用する際の注意事項を英字により記録する。

注13　電子証明書を利用する際の注意事項を記録する。

注14　電磁的記録の「corporateName」に記録された事項を記録する。

注15　会社法人等番号を記録する。

注16　電磁的記録の「corporateAddress」に記録された事項を記録する。

注17　電磁的記録の「representativeDirectorName」に記録された事項を記録する。

注18　電磁的記録の「representativeDirectorTitle」に記録された事項を記録する。

注19　商業登記規則第33条の8第2項第4号の登記所を記録する。

注20　「tbsCertificate」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

3　登記官の電子証明書

フィールド	データ型	設定値
―	Certificate	
tbsCertificate	TBSCertificate	
version	[0]	
―	Version	2
serialNumber	CertificateSerialNumber	(注1)
signature	AlgorithmIdentifier	
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11
parameters	NULL	(注2)

issuer	Name(RDNSequence)	
—	RelativeDistinguishedName	
(countryName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.6
value	PrintableString	JP
—	RelativeDistinguishedName	
(organizationName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.10
value	DirectoryString (UTF8String)	Japanese Government
—	RelativeDistinguishedName	
(organizationalUnitName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.11
value	DirectoryString (UTF8String)	Ministry of Justice
—	RelativeDistinguishedName	
(commonName)	AttributeTypeAndValue	

type	OBJECT IDENTIFIER	2.5.4.3
value	DirectoryString (UTF8String)	Registrar of Tokyo Legal Affairs Bureau
validity	Validity	
notBefore	Time(UTCTime)	(注3)
notAfter	Time(UTCTime)	(注4)
subject	Name(RDNSequence)	
—	RelativeDistinguishedName	
(countryName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.6
value	PrintableString	JP
—	RelativeDistinguishedName	
(organizationName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.10
value	DirectoryString (UTF8String)	Japanese Government
—	RelativeDistinguishedName	
(organizationalUnitName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.11

value	DirectoryString (UTF8String)	Ministry of Justice
—	RelativeDistinguishedName	
(commonName)	AttributeTypeAndValue	
type	OBJECT IDENTIFIER	2.5.4.3
value	DirectoryString (UTF8String)	Registrar of Tokyo Legal Affairs Bureau
subjectPublicKeyInfo	SubjectPublicKeyInfo	
algorithm	AlgorithmIdentifier	
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1
parameters	NULL	(注2)
subjectPublicKey	BIT STRING	(注5)
extensions	[3]	
—	Extensions	
(subjectKeyIdentifier)	Extension	
extnId	OBJECT IDENTIFIER	2.5.29.14
extnValue	OCTET STRING	
—	SubjectKeyIdentifier	(注6)
(keyUsage)	Extension	
extnId	OBJECT IDENTIFIER	2.5.29.15

extnValue	OCTET STRING	
—	KeyUsage	1 0 1 1 0 1 1 0 0
(privateKeyUsagePeriod)	Extension	
extnId	OBJECT IDENTIFIER	2.5.29.16
extnValue	OCTET STRING	
—	PrivateKeyUsagePeriod	
notBefore	[0] GeneralizedTime	(注7)
notAfter	[1] GeneralizedTime	(注8)
(basicConstraints)	Extension	
extnId	OBJECT IDENTIFIER	2.5.29.19
extnValue	OCTET STRING	
—	BasicConstraintsSyntax	
cA	BOOLEAN DEFAULT FALSE	TRUE
(registrar)	Extension	
extnId	OBJECT IDENTIFIER	1.2.392.100300.1.1.2
extnValue	OCTET STRING	
—	RegistrarSyntax(UTF8String)	東京法務局登記官
algorithm	AlgorithmIdentifier	

algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11
parameters	NULL	(注2)
signature	BIT STRING	(注9)

注1　登記官の電子証明書の番号を記録する。

注2　長さオクテットに「0」を記録する。

注3　登記官が電子証明書の使用を開始する日の日本時間0時0分0秒をグリニッジ標準時により記録する。

注4　登記官が電子証明書の使用を開始する日から起算して、72月を経過した日の日本時間23時59分59秒をグリニッジ標準時により記録する。

注5　登記官の公開かぎを記録する。

注6　電子証明書の「subjectPublicKey」の値(識別子オクテット、長さオクテット及び未使用ビットを除く。)をSHA-1により変換した値を記録する。

注7　登記官が電子証明書の使用を開始する日の日本時間0時0分0秒をグリニッジ標準時により記録する。

注8　登記官が電子証明書の使用を開始する日から起算して、36月を経過した日の日本時間の23時59分59秒をグリニッジ標準時により記録する。

注9　「tbsCertificate」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

第3　電子証明書の送信の方式

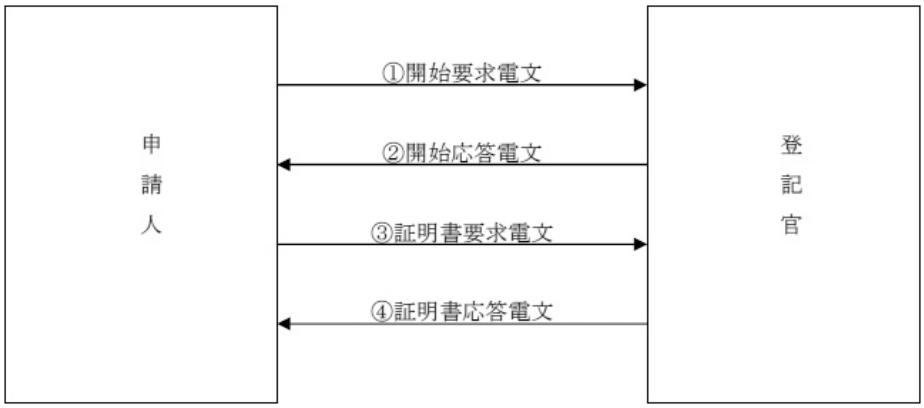
1　通信プロトコル

商業登記規則第33条の8第1項の方式による送信に用いる通信プロトコルは、インターネットエンジニアリングタスクフォースがRequest for Comments:2616において定めたHypertext Transfer Protocol--HTTP/1.1(以下「HTTP」という。)とする。

2　電文の送受信の基本形式

登記官と申請人との間の電文の送受信は、以下の手順による。

なお、①から④までの各電文の構成は、3以下で定める。



① 申請人は、登記官に「開始要求電文」を送信する。

② 登記官は、申請人にsha-256WithRSAによる電子署名を講じた「開始応答電文」を送信する。

③ 申請人は、登記官にsha-256WithRSAによる電子署名を講じた「証明書要求電文」を送信する。

④ 登記官は、申請人の電子証明書を共通かぎ暗号方式(共通かぎによる対称アルゴリズムに基づく暗号方式をいう。)により暗号化したもの及び暗号化に用いた共通かぎを申請人の公開かぎを用いて暗号化したものに、sha-256WithRSAによる電子署名を講じた「証明書応答電文」を送信する。

3 各電文の構成

前記2の各電文の構成は、次の(1)及び(2)に定めるところにより、各項目に該当する設定値を記録し、その次に区切り欄に掲げる制御記号(「CR」は「復帰」を、「LF」は「改行」を示す。以下同じ。)を記録する。

(1) 「開始要求電文」及び「証明書要求電文」の構成

項番	項目	設定値	区切り
1	Request-Line	POST△/bin/dcwcgi/DC_HUSR/cert/cert △HTTP/1.1(注1)	CR＋LF
2	request-header	Host:crca.moj.go.jp	CR＋LF
3	entity-header	Content-Type:application/pkixcmp	CR＋LF
		Content-Length:N(注2)	CR＋LF
4	general-header	Connection:close	CR＋LF CR＋LF
5	entity-body	(注3)	

注1　「△」は、スペース(間隔)を表す。以下同じ。

注2　「N」は、項番5「entity-body」のバイト長を記録する。

注3　項番5「entity-body」の設定値は、後記4に定めるところによる。

(2)　「開始応答電文」及び「証明書応答電文」の構成

項番	項目	設定値	区切り
1	Status-Line	HTTP/1.1△200△OK	CR＋LF
2	entity-header	Content-Type:application/pkixcmp	CR＋LF
3		Content-Length:N(注1)	CR＋LF
4	general-header	Date:(注2)	CR＋LF
5		Connection:close	CR＋LF
6	response-header	Server:(注3)	CR＋LF
7	Set-Cookie	Set-Cookie: (注4)	CR＋LF CR＋LF
8	entity-body	(注5)	

注1　「N」は、項番8「entity-body」のバイト長を記録する。

注2　送信の日時をグリニッジ標準時により記録する。

注3　「Server:」の次に、登記官が適宜の事項を記録することができる。

注4　「Set-Cookie:」の次に、登記官が適宜の事項を記録することができる。

注5　項番8「entity-body」の設定値は、後記4に定めるところによる。

4　各電文中の「entity-body」の内容

(1)　前記3の各電文中の「entity-body」には、次の(2)から(5)までに定めるところにより、「データ型」欄に掲げる形式を用いて、付録3の様式に従って、「フィールド」欄に掲げる事項を記録する。この場合において、「データ型」欄に掲げる形式は、ASN.1及び付録3に定めるところによる。データの符号化は、DERによる。

(2)　「開始要求電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須 (注1)
—	PKIMessage		
header	PKIHeader		

pvno	INTEGER	1	◎
sender	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
transactionID	[4]		
—	OCTET STRING	1バイト以上32バイト以下の乱数を記録する。	◎
senderNonce	[5]		
—	OCTET STRING	1バイト以上32バイト以下の乱数を記録する。	◎
body	PKIBody([21])		
—	GenMsgContent		
—	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.21	◎
infoValue	GenmInfoReqContent		
—	NegotiationKey		
symmAlg	AlgorithmIdentifier		

algorithm	OBJECT IDENTIFIER	1.2.840.113549.3.7	◎
parameters	NULL		△
pubAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1	◎
parameters	NULL		△
hashAlg	AlgorithmIdentifier		
Algorithm	OBJECT IDENTIFIER	2.16.840.1.101.3.4.2.1	◎
Parameters	NULL		△

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。

「必須」欄中に○印のあるフィールドは、必ず設けなければならない。「必須」欄中に△印のあるフィールドは、記録する内容によって任意に設けることができる。

注2　長さオクテットに「0」を記録する。

(3)　「開始応答電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須(注1)
—	PKIMessage		
Header	PKIHeader		
Pvno	INTEGER	1	◎
sender	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		

—	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注2)	○
senderKID	[2]		
—	KeyIdentifier	(注3)	◎
transactionID	[4]		
—	OCTET STRING	(注4)	◎
senderNonce	[5]		
—	OCTET STRING	(注5)	◎
recipNonce	[6]		
—	OCTET STRING	(注6)	◎
body	PKIBody([22])		
—	GenRepContent		
—	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.22	◎
infoValue	GenpInfoResContent		
status	PKIStatusInfo		
status	PKIStatus	0(注7)	◎
negotiationKeys	SEQUENCE OF		

	NegotiationKey		
—	NegotiationKey		
symmAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.3.7	◎
parameters	NULL	(注2)	○
pubAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1	◎
parameters	NULL	(注2)	○
hashAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	2.16.840.1.101.3.4.2.1	◎
parameters	NULL	(注2)	○
protection	[0]		
—	PKIProtection	(注8)	◎
extraCerts	[1]		
—	SEQUENCE OF Certificate	(注9)	◎

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。
「必須」欄中に○印のあるフィールドは、必ず設けなければならない。

注2　長さオクテットに「0」を記録する。

注3　注9に記録された登記官の電子証明書の「subjectPublicKey」の値(識別子オクテット、長さオクテット及び未使用ビットを除く。)をSHA-1により変換した値を記録する。

注4　「開始要求電文」中の「transactionID」に記録された値を記録する。

注5　1バイト以上32バイト以下の乱数を記録する。

注6　「開始要求電文」中の「senderNonce」に記録された値を記録する。

注7　「開始要求電文」を正常に受信したことを示すため、「0」を記録する。「開始要求電文」に異常があった場合には、後記5による。

注8　付録3に示す「ProtectedPart」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

注9　登記官の最新の電子証明書を記録する。

(4)　「証明書要求電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須(注1)
―	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
transactionID	[4]		
―	OCTET STRING	「開始応答電文」中の「transactionID」を記録する。	◎
senderNonce	[5]		
―	OCTET STRING	「開始応答電文」中の「recipNonce」を記録する。	◎

recipNonce	[6]		
—	OCTET STRING	「開始応答電文」中の「senderNonce」を記録する。	◎
body	PKIBody([0])		
—	CertReqMessages		
—	CertReqMsg		
certReq	CertRequest		
certReqId	INTEGER	0	◎
certTemplate	CertTemplate		
serialNumber	[1] INTEGER	商業登記規則第33条の8第2項第2号の電子証明書の番号を記録する。	◎
publicKey	[6] SubjectPublicKeyInfo		
algorithm	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1	◎
parameters	NULL	(注2)	○
subjectPublicKey	BIT STRING	請求に係る公開かぎを記録する。	◎
pop	ProofOfPossession ([1] POPOSigningKey)		

algorithmIdentifier	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL		△
signature	BIT STRING	(注3)	◎

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。「必須」欄中に○印のあるフィールドは、必ず設けなければならない。「必須」欄中に△印のあるフィールドは、記録する内容によって任意に設けることができる。

注2　長さオクテットに「0」を記録する。

注3　「certReq」に属する部分をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。ただし、「subjectPublicKey」に記録した公開かぎにより当該電子署名を講じた措置が検証できるものでなければならない。

(5)　「証明書応答電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須 (注1)
―	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
―	AlgorithmIdentifier		

algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注2)	○
senderKID	[2]		
—	KeyIdentifier	(注3)	◎
transactionID	[4]		
—	OCTET STRING	(注4)	◎
senderNonce	[5]		
—	OCTET STRING	(注5)	◎
recipNonce	[6]		
—	OCTET STRING	(注6)	◎
body	PKIBody([1])		
—	CertRepMessage		
response	SEQUENCE OF CertResponse		
—	CertResponse		
certReqId	INTEGER	0	◎
status	PKIStatusInfo		
status	PKIStatus	0(注7)	◎
certifiedKeyPair	CertifiedKeyPair		

certOrEncCert	CertOrEncCert([1])		
—	EncryptedValue		
symmAlg	[1]AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.3.7	◎
parameters	CBCParameter	IV	◎
encSymmKey	[2] BIT STRING	(注8)	◎
keyAlg	[3]AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1	◎
parameters	NULL	(注2)	○
encValue	BIT STRING	(注9)	◎
protection	[0]		
—	PKIProtection	(注10)	◎
extraCerts	[1]		
—	SEQUENCE OF Certificate	(注11)	◎

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。

「必須」欄中に○印のあるフィールドは、必ず設けなければならない。

注2　長さオクテットに「0」を記録する。

注3　登記官の最新の電子証明書の「subjectPublicKey」の値(識別子オクテット、長さオクテット及び未使用ビットを除く。)を、SHA-1により変換した値を記録する。

注4　「証明書要求電文」中の「transactionID」に記録された値を記録する。

注5　「証明書要求電文」中の「recipNonce」に記録された値を記録する。

注6　「証明書要求電文」中の「senderNonce」に記録された値を記録する。

注7　「証明書要求電文」を正常に受信したことを示すため、「0」を記録する。「証明書要求電文」に異常があった場合には、後記5による。

注8　電子証明書の暗号化(オブジェクト識別子を「1.2.840.113549.3.7」とする共通かぎ暗号方式によるものに限る。)に用いる共通かぎの値を「証明要求電文」の「subjectPublicKey」に記録された公開かぎを用いて、オブジェクト識別子を「1.2.840.113549.1.1.1」とする暗号アルゴリズムにより暗号化した値を記録する。

注9　電子証明書を注8の共通かぎを用いて暗号化した値を記録する。

注10　付録3に示す「ProtectedPart」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

注11　電子証明書に署名をした登記官の電子証明書を記録する。この場合において、その登記官の電子証明書が最新のものでないときは、最新の電子証明書も併せて記録する。

5　異常時の処理

(1)　「開始要求電文」中の「entity-body」の内容が前記4の(2)に定める形式に適合しない場合(後記(2)の場合を除く。)には、登記官は、「開始応答電文」の項番2については、「Content-Type:application/pkixcmp」に代えて「Content-Type:text/html」と記録し、「entity-body」については、前記4の(3)の内容に代えて、次の内容を記録したものを送信する。この場合には、申請人は、「証明書要求電文」を送信することができない。

＜!DOCTYPE HTML PUBLIC”-//W3C//DTD HTML 4.0//EN”＞ ＜HTML lang=”ja”＞ ＜META HTTP-EQUIV=”Content-Type” CONTENT=”text/html;charset=SHIFT_JIS”＞ ＜TITLE＞メッセージ異常＜/TITLE＞
＜BODY＞ メッセージ内容に問題があるため、処理できませんでした。 ＜/BODY＞ ＜/HTML＞

(2)　「開始要求電文」中の「entity-body」に記録された「symmAlg」、「pubAlg」又は「hashAlg」が、前記4の(2)に定める形式に適合しない場合には、登記官は、「開始応答電文」の「entity-body」に、前記4の(3)の内容に代えて、前記4の(1)に定める方式により、次の内容を記録したものを送信する。この場合には、申請人は、「証明書要求電文」を送信することができない。

フィールド	データ型	設定値	必須

			(注1)
—	PKIMessage		
Header	PKIHeader		
Pvno	INTEGER	1	◎
Sender	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
Recipient	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
—	AlgorithmIdentifier		
Algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
Parameters	NULL	(注2)	○
senderKID	[2]		
—	KeyIdentifier	(注3)	◎
transactionID	[4]		
—	OCTET STRING	(注4)	◎
senderNonce	[5]		
—	OCTET STRING	(注5)	◎
recipNonce	[6]		

―	OCTET STRING	(注6)	◎
Body	PKIBody([22])		
―	GenRepContent		
―	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.22	◎
infoValue	GenpInfoResContent		
Status	PKIStatusInfo		
Status	PKIStatus	2(注7)	◎
Protection	[0]		
―	PKIProtection	(注8)	◎
extraCerts	[1]		
―	SEQUENCE OF Certificate	(注9)	◎

注1から注6まで、注8及び注9については、前記4の(3)の注1から注6まで、注8及び注9に同じ。

注7　「開始要求電文」に異常があったことを示すため、「2」を記録する。

(3)　「証明書要求電文」中の「entity-body」が、前記4の(4)に定める形式に適合しない場合(後記5の(4)の場合を除く。)には、登記官は、「証明書応答電文」の「entity-body」に、前記4の(5)の内容に代えて、前記4の(1)に定める方式により、次の内容を記録したものを送信する。この場合には、申請人は、「開始要求電文」を再送信しなければならない。

フィールド	データ型	設定値	必須(注1)
―	PKIMessage		

Header	PKIHeader		
Pvno	INTEGER	1	◎
Sender	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
Recipient	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
—	AlgorithmIdentifier		
Algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
Parameters	NULL	(注2)	○
senderKID	[2]		
—	KeyIdentifier	(注3)	◎
transactionID	[4]		
—	OCTET STRING	(注4)	◎
senderNonce	[5]		
—	OCTET STRING	(注5)	◎
recipNonce	[6]		
—	OCTET STRING	(注6)	◎
Body	PKIBody([23])		

―	ErrorMsgContent		
pKIStatusInfo	PKIStatusInfo		
Status	PKIStatus	2(注7)	◎
Protection	[0]		
―	PKIProtection	(注8)	◎
extraCerts	[1]		
―	SEQUENCE OF Certificate	(注9)	◎

注1から注6までについては、前記4の(5)の注1から注6までに同じ。

注7　「証明書要求電文」に異常があったことを示すため、「2」を記録する。

注8　付録3に示す「ProtectedPart」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

注9　登記官の最新の電子証明書を記録する。

(4)　「証明書要求電文」中の「entity-body」に記録された「transactionID」、「senderNonce」、「recipNonce」、「serialNumber」又は「subjectPublicKey」が、前記4の(4)に定める形式に適合しないときは、登記官は、「証明書応答電文」の「entity-body」に、前記4の(5)の内容に代えて、前記4の(1)に定める方式により、次の内容を記録したものを送信する。この場合には、申請人は、「開始要求電文」を再送信しなければならない。

フィールド	データ型	設定値	必須 (注1)
―	PKIMessage		
Header	PKIHeader		
Pvno	INTEGER	1	◎

Sender	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
—	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注2)	○
senderKID	[2]		
—	KeyIdentifier	(注3)	◎
transactionID	[4]		
—	OCTET STRING	(注4)	◎
senderNonce	[5]		
—	OCTET STRING	(注5)	◎
recipNonce	[6]		
—	OCTET STRING	(注6)	◎
Body	PKIBody([1])		
—	CertRepMessage		
response	SEQUENCE OF CertResponse		

—	CertResponse		
certReqId	INTEGER	0	◎
status	PKIStatusInfo		
status	PKIStatus	2(注7)	◎
protection	[0]		
—	PKIProtection	(注8)	◎
extraCerts	[1]		
—	SEQUENCE OF Certificate	(注9)	◎

注1から注9までについては、前記(3)の注1から注9までと同じ。

第4　識別符号の変更の届出に使用する電磁的記録への記録の方式

1　使用する電磁的記録媒体及びファイル名

使用する電磁的記録媒体、ファイル名及びファイルへの記録方式については、前記第1の1から3までに定めるところによる。

2　識別符号の変更届出ファイル

フィールド	データ型	設定値	必須(注1)
—	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
—	Name(RDNSequence)		○
recipient	GeneralName([4])		

—	Name(RDNSequence)		○
Body	PKIBody([0])		
—	CertReqMessages		
—	CertReqMsg		
certReq	CertRequest		○
regInfo	SEQUENCE OF AttributeTypeAndValue		
(suspensionSecretCode)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	1.2.392.100300.1.2.105	◎
value	SuspensionSecretCode		
hashAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	2.16.840.1.101.3.4.2.1	◎
parameters	NULL		△
hashedSecretCode	OCTET STRING	(注2)	◎

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。「必須」欄中に○印のあるフィールドは、必ず設けなければならない。「必須」欄中に△印のあるフィールドは、記録する内容によって任意に設けることができる。

注2　変更後の識別符号をSHA-256により変換した値を記録する。識別符号の長さは、8バイト以上64バイト以下とする。使用する文字等の範囲は、日本産業規格X0201で規定されたラテン文字用図形文字集合とする。文字等の符号化表現は、日本産業規格X0208-1997附属書1に規定する方式による。

第5　休止届の送信の方式

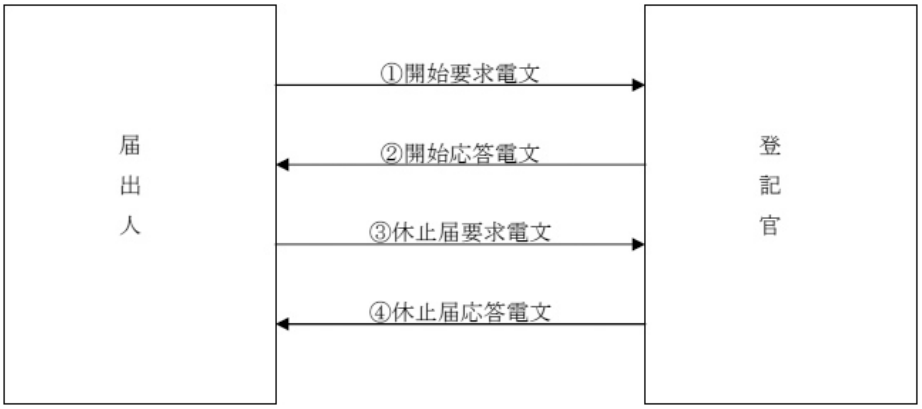
1　通信プロトコル

商業登記規則第33条の13第2項の規定による送信については、登記官が使用する電子計算機と届出人が使用する電子計算機とを接続する電気通信回線を通じて行うものとし、その通信プロトコルは、HTTPとする。

2　電文の送受信の基本形式

登記官と届出人との間の電文の送受信は、以下の手順による。

なお、①から④までの各電文の構成については、後記3以下で定める。



①　届出人は、登記官に「開始要求電文」を送信する。

②　登記官は、届出人にsha-256WithRSAによる電子署名を講じた「開始応答電文」を送信する。

③　届出人は、登記官に識別符号を含むデータ等を登記官の公開かぎを用いて暗号化した「休止届要求電文」を送信する。

④　登記官は、届出人にsha-256WithRSAによる電子署名を講じた「休止届応答電文」を送信する。

3　各電文の構成

前記2の各電文の構成は、次の(1)及び(2)に定めるところにより、各項目に該当する設定値を記録し、その次に区切り欄に掲げる制御記号を記録する。

(1)　「開始要求電文」及び「休止届要求電文」の構成

項番	項目	設定値	区切り
1	Request-Line	POST△/bin/dcwcgi/DC_HUSR/cert/cert △HTTP/1.1	CR＋LF
2	request-header	Host:crca.moj.go.jp	CR＋LF
3	entity-header	Content-Type:application/pkixcmp	CR＋LF

		Content-Length:N(注1)	CR＋LF
4	general-header	Connection:close	CR＋LF CR＋LF
5	entity-body	(注2)	

注1　「N」は、項番5「entity-body」のバイト長を記録する。

注2　項番5「entity-body」の設定値は、後記4に定めるところによる。

(2)　「開始応答電文」及び「休止届応答電文」の構成

項番	名前	設定値	区切り
1	Status-Line	HTTP/1.1△200△OK	CR＋LF
2	entity-header	Content-Type:application/pkixcmp	CR＋LF
3		Content-Length:N(注1)	CR＋LF
4	general-header	Date:(注2)	CR＋LF
5		Connection:close	CR＋LF
6	response-header	Server:(注3)	CR＋LF
7	Set-Cookie	Set-Cookie: (注4)	CR＋LF CR＋LF
8	entity-body	(注5)	

注1　「N」は、項番8「entity-body」のバイト長を記録する。

注2　送信の日時をグリニッジ標準時により記録する。

注3　「Server:」の次に、登記官が適宜の事項を記録することができる。

注4　「Set-Cookie:」の次に、登記官が適宜の事項を記録することができる。

注5　項番8「entity-body」の設定値は、後記4に定めるところによる。

4　各電文中の「entity-body」の内容

(1)　前記3の各電文中の「entity-body」には、次の(2)から(5)までに定めるところにより、「データ型」欄に掲げる形式を用いて、付録4の様式に従って、「フィールド」欄に掲げる事項を記録する。この場合において、「データ型」欄の形式は、ASN.1及び付録4に定めるところによる。データの符号化は、DERによる。

(2)　「開始要求電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須 (注1)

―	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
transactionID	[4]		
―	OCTET STRING	1バイト以上32バイト以下の乱数を記録する。	◎
senderNonce	[5]		
―	OCTET STRING	1バイト以上32バイト以下の乱数を記録する。	◎
Body	PKIBody([21])		
―	GenMsgContent		
―	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.21	◎
infoValue	GenmInfoReqContent		
―	NegotiationKey		

symmAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.3.7	◎
parameters	NULL		△
pubAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1	◎
parameters	NULL		△
hashAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	2.16.840.1.101.3.4.2.1	◎
parameters	NULL		△

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。「必須」欄中に○印のあるフィールドは、必ず設けなければならない。「必須」欄中に△印のあるフィールドは、記録する内容によって任意に設けることができる。

注2　長さオクテットに「0」を記録する。

(3)　「開始応答電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須(注1)
―	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		

—	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
—	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注2)	○
senderKID	[2]		
—	KeyIdentifier	(注3)	◎
transactionID	[4]		
—	OCTET STRING	(注4)	◎
senderNonce	[5]		
—	OCTET STRING	(注5)	◎
recipNonce	[6]		
—	OCTET STRING	(注6)	◎
Body	PKIBody([22])		
—	GenRepContent		
—	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.22	◎
infoValue	GenpInfoResContent		
status	PKIStatusInfo		

status	PKIStatus	0(注7)	◎
negotiationKeys	SEQUENCE OF NegotiationKey		
—	NegotiationKey		
symmAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.3.7	◎
parameters	NULL	(注2)	○
pubAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1	◎
parameters	NULL	(注2)	○
hashAlg	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	2.16.840.1.101.3.4.2.1	◎
parameters	NULL	(注2)	○
protection	[0]		
—	PKIProtection	(注8)	◎
extraCerts	[1]		
—	SEQUENCE OF Certificate	(注9)	◎

注1から注6までについては、前記第3の4の(3)の注1から注6までに同じ。

注7　「開始要求電文」を正常に受信したことを示すため、「0」を記録する。「開始要求電文」に異常があった場合には、後記5による。

注8　付録4に示す「ProtectedPart」をDERにより符号化したsha-256WithRSAによる電子署名を講じた値を記録する。

注9　登記官の最新の電子証明書を記録する。

(4)　「休止届要求電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須(注1)
―	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
transactionID	[4]		
―	OCTET STRING	「開始応答電文」中の「transactionID」を記録する。	◎
senderNonce	[5]		
―	OCTET STRING	「開始応答電文」中の「recipNonce」を記録する。	◎
recipNonce	[6]		
―	OCTET STRING	「開始応答電文」中の「senderNonce」を記録する。	◎

Body	PKIBody([21])		
—	GenMsgContent		
—	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.1	◎
infoValue	GenmSuspReqContent		
certDetails	CertTemplate		
serialNumber	[1] INTEGER	電子証明書の番号を記録する。	◎
issuer	[3]		
—	Name(RDNSequence)		
—	RelativeDistinguishedName		
(countryName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.6	◎
value	PrintableString	JP	◎
—	RelativeDistinguishedName		
(organizationName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.10	◎

value	DirectoryString (UTF8String)	Japanese Government	◎
—	RelativeDistinguishedName		
(organizationalUnitName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.11	◎
value	DirectoryString (UTF8String)	Ministry of Justice	◎
—	RelativeDistinguishedName		
(commonName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.3	◎
value	DirectoryString (UTF8String)	Registrar of Tokyo Legal Affairs Bureau	◎
revocationReason	ReasonFlags	6	◎
suspensionReasonCode	INTEGER	1	◎
suspensionDetail	EncryptedValue		
keyAlg	[3]AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.1	◎
parameters	NULL		△
encValue	BIT STRING	(注3)	◎

注1 「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。

「必須」欄中に○印のあるフィールドは、必ず設けなければならない。「必須」欄

中に△印のあるフィールドは、記録する内容によって任意に設けることができる。

注2　長さオクテットに「0」を記録する。

注3　商業登記規則第33条の6第5項第4号の識別符号の次に、DERにより符号化した「header」部をSHA-1により変換した値を付加したものを、登記官の公開かぎを用いて、オブジェクト識別子を「1.2.840.113549.1.1.1」とする暗号アルゴリズムにより暗号化した値を記録する。登記官の公開かぎは、「開始応答電文」の注9に記録されたものを使用しなければならない。

(5)　「休止届応答電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須(注1)
―	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
―	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注2)	○
senderKID	[2]		
―	KeyIdentifier	(注3)	◎

transactionID	[4]		
—	OCTET STRING	(注4)	◎
senderNonce	[5]		
—	OCTET STRING	(注5)	◎
recipNonce	[6]		
—	OCTET STRING	(注6)	◎
body	PKIBody([22])		
—	GenRepContent		
—	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.2	◎
infoValue	GenpSuspResContent		
status	PKIStatusInfo		
status	PKIStatus	0(注7)	◎
revCert	CertId		
issuer	GeneralName([4])		
—	Name(RDNSequence)		
—	RelativeDistinguishedName		

(countryName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.6	◎
value	PrintableString	JP	◎
—	RelativeDistinguishedName		
(organizationName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.10	◎
value	DirectoryString (UTF8String)	Japanese Government	◎
—	RelativeDistinguishedName		
(organizationalUnitName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.11	◎
value	DirectoryString (UTF8String)	Ministry of Justice	◎
—	RelativeDistinguishedName		
(commonName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.3	◎
value	DirectoryString (UTF8String)	Registrar of Tokyo Legal Affairs Bureau	◎

serialNumber	INTEGER	(注8)	◎
protection	[0]		
―	PKIProtection	(注9)	◎
extraCerts	[1]		
―	SEQUENCE OF Certificate	(注10)	◎

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。「必須」欄中に○印のあるフィールドは、必ず設けなければならない。

注2　長さオクテットに「0」を記録する。

注3　注10に記録された登記官の電子証明書の「subjectPublicKey」の値(識別子オクテット、長さオクテット及び未使用ビットを除く。)を、SHA-1により変換した値を記録する。

注4　「休止届要求電文」中の「transactionID」に記録された値を記録する。

注5　「休止届要求電文」中の「recipNonce」に記録された値を記録する。

注6　「休止届要求電文」中の「senderNonce」に記録された値を記録する。

注7　「休止届要求電文」を正常に受信したことを示すため、「0」を記録する。「休止届要求電文」に異常があった場合には、後記5による。

注8　電子証明書の番号を記録する。

注9　付録4に示す「ProtectedPart」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

注10　登記官の最新の電子証明書を記録する。

5　異常時の処理

(1)　「開始要求電文」中の「entity-body」が前記4の(2)に定める形式に適合しない場合(後記5の(2)に定める場合を除く。)における登記官の措置については、前記第3の5の(1)による。この場合には、届出人は、「休止届要求電文」を送信することができない。

(2)　「開始要求電文」中の「entity-body」に記録された「symmAlg」、「pubAlg」又は「hashAlg」が前記4の(2)に定める形式に適合しない場合は、登記官は、「開始応答電文」の「entity-body」に、前記4の(3)の内容に代えて、前記4の(1)の定める方

式により、次の内容を記録したものを送信する。この場合には、届出人は、「休止届要求電文」を送信することができない。

フィールド	データ型	設定値	必須 (注1)
―	PKIMessage		
Header	PKIHeader		
Pvno	INTEGER	1	◎
Sender	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
―	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注2)	○
senderKID	[2]		
―	KeyIdentifier	(注3)	◎
transactionID	[4]		
―	OCTET STRING	(注4)	◎
senderNonce	[5]		

―	OCTET STRING	(注5)	◎
recipNonce	[6]		
―	OCTET STRING	(注6)	◎
Body	PKIBody([22])		
―	GenRepContent		
―	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.22	◎
infoValue	GenpInfoResContent		
status	PKIStatusInfo		
status	PKIStatus	2(注7)	◎
protection	[0]		
―	PKIProtection	(注8)	◎
extraCerts	[1]		
―	SEQUENCE OF Certificate	(注9)	◎

注1から注6までについては、前記第3の4の(3)の注1から注6までに同じ。

注7　「開始要求電文」に異常があったことを示すため、「2」を記録する。

注8　付録4に示す「ProtectedPart」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

注9　登記官の最新の電子証明書を記録する。

(3)　「休止届要求電文」中の「entity-body」が、前記4の(4)に定める形式に適合しない場合(後記(4)の場合を除く。)には、登記官は、「休止届応答電文」の「entity-body」に、前記4の(5)の内容に代えて、前記4の(1)に定める方式により、次の内容を記録し

たものを送信する。この場合には、届出人は、「開始要求電文」を再送信しなければならない。

フィールド	データ型	設定値	必須 (注1)
―	PKIMessage		
header	PKIHeader		
pvno	INTEGER	1	◎
sender	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
―	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
―	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注2)	○
senderKID	[2]		
―	KeyIdentifier	(注3)	◎
transactionID	[4]		
―	OCTET STRING	(注4)	◎
senderNonce	[5]		

―	OCTET STRING	(注5)	◎
recipNonce	[6]		
―	OCTET STRING	(注6)	◎
Body	PKIBody([23])		
―	ErrorMsgContent		
pKIStatusInfo	PKIStatusInfo		
status	PKIStatus	2 (注7)	◎
protection	[0]		
―	PKIProtection	(注8)	◎
extraCerts	[1]		
―	SEQUENCE OF Certificate	(注9)	◎

注1から注6までについては、前記4の(5)の注1から注6までに同じ。

注7　「休止届要求電文」に異常があったことを示すため、「2」を記録する。

注8　付録4に示す「ProtectedPart」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

注9　登記官の最新の電子証明書を記録する。

(4)　「休止届要求電文」中の「entity-body」に記録された「transactionID」、「senderNonce」、「recipNonce」、「serialNumber」、「issuer」、「suspensionReasonCode」又は「encValue」が、前記4の(4)に定める形式に適合しないときは、登記官は、「休止届応答電文」の「entity-body」に、前記4の(5)の内容に代えて、前記4の(1)に定める方式により、次の内容を記録したものを送信する。この場合には、届出人は、「開始要求電文」を再送信しなければならない。

フィールド	データ型	設定値	必須 (注1)
―	PKIMessage		

header	PKIHeader		
Pvno	INTEGER	1	◎
sender	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
recipient	GeneralName([4])		
—	Name(RDNSequence)	(注2)	○
protectionAlg	[1]		
—	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注2)	○
senderKID	[2]		
—	KeyIdentifier	(注3)	◎
transactionID	[4]		
—	OCTET STRING	(注4)	◎
senderNonce	[5]		
—	OCTET STRING	(注5)	◎
recipNonce	[6]		
—	OCTET STRING	(注6)	◎
Body	PKIBody([22])		

—	GenRepContent		
—	InfoTypeAndValue		
infoType	OBJECT IDENTIFIER	1.2.392.100300.1.2.2	◎
infoValue	GenpSuspResContent		
status	PKIStatusInfo		
status	PKIStatus	2(注7)	◎
revCert	CertId		
issuer	GeneralName([4])		
—	Name(RDNSequence)		
—	RelativeDistinguishedName		
(countryName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.6	◎
value	PrintableString	JP	◎
—	RelativeDistinguishedName		
(organizationName)	AttributeTypeAndValue		

type	OBJECT IDENTIFIER	2.5.4.10	◎
value	DirectoryString (UTF8String)	Japanese Government	◎
—	RelativeDistinguishedName		
(organizationalUnitName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.11	◎
value	DirectoryString (UTF8String)	Ministry of Justice	◎
—	RelativeDistinguishedName		
(commonName)	AttributeTypeAndValue		
type	OBJECT IDENTIFIER	2.5.4.3	◎
value	DirectoryString (UTF8String)	Registrar of Tokyo Legal Affairs Bureau	◎
serialNumber	INTEGER	(注8)	◎
protection	[0]		
—	PKIProtection	(注9)	◎
extraCerts	[1]		
—	SEQUENCE OF Certificate	(注10)	◎

注1から注6までについて、前記4の(5)の注1から注6までに同じ。

注7　「休止届要求電文」に異常があったことを示すため、「2」を記録する。

注8　電子証明書の番号を記録する。

注9　付録4に示す「ProtectedPart」をDERにより符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

注10　登記官の最新の電子証明書を記録する。

第6　電子証明書に係る証明及びその請求の方式

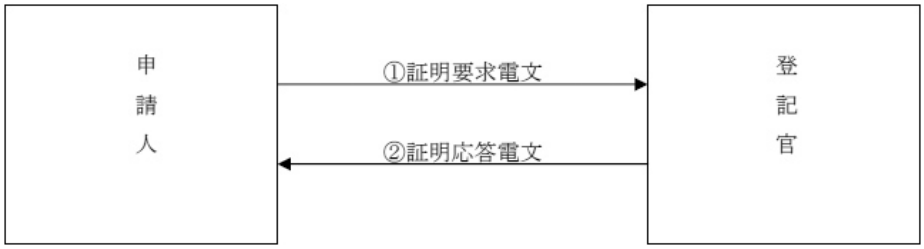
1　通信プロトコル

商業登記規則第33条の15第2項の規定及び同条第3項において準用する第33条の8第1項の規定による送信に用いる通信プロトコルは、HTTPとする。

2　電文の送受信の基本形式

登記官と申請人との間の電文の送受信は、以下の手順による。

なお、①及び②の各電文の構成については、後記3以下で定める。



①　申請人が登記官に送信する「証明要求電文」は、請求に係る電子証明書に記録された「notAfter」の日時までに、電子認証登記所に到達しなければならない。また、過去の特定の日時(電子証明書の記録された「notBefore」から「notAfter」までの範囲に限る。)についての「証明要求電文」は、請求に係る電子証明書に記録された「notAfter」の日の翌日から起算して7日を超えない日までに、電子認証登記所に到達しなければならない。

②　登記官は、申請人にsha-256WithRSAによる電子署名を講じた「証明応答電文」を送信する。

3　各電文の構成

前記2の各電文の構成は、次の(1)及び(2)に定めるところにより、各項目に該当する設定値を記録し、その次に区切り欄に掲げる制御記号を記録する。

(1)　「証明要求電文」の構成

項番	項目	設定値	区切り
1	Request-Line	POST△/bin/dcwcgi/DC_HUSR/cert/cert △HTTP/1.1	CR＋LF
2	request-header	Host:crca.moj.go.jp	CR＋LF

3	entity-header	Content-Type:application/ocsp-request	CR＋LF
		Content-Length:N(注1)	CR＋LF
4	general-header	Connection:close	CR＋LF CR＋LF
5	entity-body	(注2)	

注1　「N」は、項番5の「entity-body」のバイト長を記録する。

注2　項番5の「entity-body」の設定値は、後記4に定めるところによる。

(2)　「証明応答電文」の構成

項番	名前	設定値	区切り
1	Status-Line	HTTP/1.1△200△OK	CR＋LF
2	entity-header	Content-Type:application/ocsp-response	CR＋LF
3		Content-Length:N(注1)	CR＋LF
4	general-header	Date:(注2)	CR＋LF
5		Connection:close	CR＋LF
6	response-header	Server:(注3)	CR＋LF
7	Set-Cookie	Set-Cookie:(注4)	CR＋LF CR＋LF
8	entity-body	(注5)	

注1　「N」は、項番8「entity-body」のバイト長を記録する。

注2　送信の日時をグリニッジ標準時により記録する。

注3　「Server:」の次に、登記官が適宜の事項を記録することができる。

注4　「Set-Cookie:」の次に、登記官が適宜の事項を記録することができる。

注5　項番8の「entity-body」の設定値は、後記4に定めるところによる。

4　各電文中の「entity-body」の内容

(1)　前記3の各電文中の「entity-body」には、次の(2)又は(3)に定めるところにより、「データ型」欄に掲げる形式を用いて、付録5の様式に従って、「フィールド」欄に掲げる事項を記録する。この場合において、「データ型」欄の形式は、ASN.1及び付録5に定めるところによる。データの符号化は、DERによる。

(2)　「証明要求電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須

			(注1)
―	OCSPRequest		
tbsRequest	TBSRequest		
requestList	SEQUENCE OF Request		
―	Request		
reqCert	CertID		
hashAlgorithm	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.3.14.3.2.26	◎
parameters	NULL		△
issuerNameHash	OCTET STRING	証明の対象となる電子証明書の「issuer」に属する部分を、DERにより符号化した値を、SHA-1により変換した値を記録する。	◎
issuerKeyHash	OCTET STRING	(注2)	◎
serialNumber	CertificateSerialNumber	証明の請求に係る電子証明書の番号を記録する。	◎
singleRequestExtensions	[0]		△ (注3)
―	Extensions		
(confirmationTime)	Extension		

extnId	OBJECT IDENTIFIER	1.2.392.100300.1.2.102	↑◎
extnValue	OCTET STRING		
—	ConfirmationTime	証明の対象となる過去の特定の日時をグリニッジ標準時により記録する(注4)。	↑◎
requestExtensions	[2]		
—	Extensions		
(nonce)	Extension		
extnId	OBJECT IDENTIFIER	1.3.6.1.5.5.7.48.1.2	◎
extnValue	OCTET STRING		
—	Nonce	1バイト以上32バイト以下の乱数を記録する。	◎

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。「必須」欄中に○印のあるフィールドは、必ず設けなければならない。「必須」欄中に△印のあるフィールドは、記録する内容によって任意に設けることができる。「必須」欄中に↑◎印のあるフィールドには、上欄にフィールドを設けたときは、必ず値を記録しなければならない。

注2　証明の対象となる電子証明書を作成した登記官の電子証明書の「subjectPublicKey」の値(識別子オクテット、長さオクテット及び未使用ビットを除く。)をSHA-1により変換した値を記録する。

注3　本フィールドを設定することにより、過去の特定の日時についての証明を請求することができる。

注4　「過去の特定の日時」とは、次の①又は②とする。

①　証明の対象となる電子証明書に「notAfter」として記録された日時までの間

にあっては、電子証明書に「notBefore」として記録された日時から証明を請求するまでの間の任意の日時

② 電子証明書に「notAfter」として記録された日時以降から「notAfter」として記録された日を経過してから7日を超えない日までの間にあっては、電子証明書に「notBefore」として記録された日時から「notAfter」として記録された日時までの間の任意の日時

(3) 「証明応答電文」中の「entity-body」の内容

フィールド	データ型	設定値	必須(注1)
―	OCSPResponse		
responseStatus	OCSPResponseStatus	0(注2)	◎
responseBytes	[0]		
―	ResponseBytes		
responseType	OBJECT IDENTIFIER	1.3.6.1.5.5.7.48.1.1	◎
response	OCTET STRING		
―	BasicOCSPResponse		
tbsResponseData	ResponseData		
responderID	ResponderID([2])		
―	KeyHash	(注3)	◎
producedAt	GeneralizedTime	(注4)	◎
responses	SEQUENCE OF SingleResponse		

—	SingleResponse		
certID	CertID		
hashAlgorithm	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.3.14.3.2.26	◎
parameters	NULL	(注5)	○
issuerNameHash	OCTET STRING	(注6)	◎
issuerKeyHash	OCTET STRING	(注7)	◎
serialNumber	CertificateSerialNumber	(注8)	◎
certStatus	CertStatus(CHOICE)		
good	[0] NULL	(注9)	△○ (注9)
revoked	[1] RevokedInfo	(注9)	△○ (注9)
revocationTime	GeneralizedTime	(注10)	↑◎
revocationReason	[0] CRLReason	(注11)	↑◎
unknown	[2] UnknownInfo	(注9)	△○ (注9)
thisUpdate	GeneralizedTime	(注12)	◎
singleExtensions	[1]		
—	Extensions		
(ocspStatusCode)	Extension		
extnId	OBJECT IDENTIFIER	1.2.392.100300.1.2.103	◎

extnValue	OCTET STRING		
—	OcspStatusCode	(注11)	◎
(confirmationTime)	Extension		△ (注 13)
extnId	OBJECT IDENTIFIER	1.2.392.100300.1.2.102	↑◎
extnValue	OCTET STRING		
—	ConfirmationTime	(注14)	↑◎
responseExtensions	[1]		
—	Extensions		
(nonce)	Extension		
extnId	OBJECT IDENTIFIER	1.3.6.1.5.5.7.48.1.2	◎
extnValue	OCTET STRING		
—	Nonce	(注15)	◎
signatureAlgorithm	AlgorithmIdentifier		
algorithm	OBJECT IDENTIFIER	1.2.840.113549.1.1.11	◎
parameters	NULL	(注5)	○
signature	BIT STRING	(注16)	◎
certs	[0]		
—	SEQUENCE OF Certificate	(注17)	◎

注1 「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。

「必須」欄中に○印のあるフィールドは、必ず設けなければならない。「必須」欄中に△印のあるフィールドは、記録する内容によって任意に設けることができる。「必須」欄中に↑◎印のあるフィールドには、上欄にフィールドを設けたときは、必ず値を記録しなければならない。「必須」欄中に△○印のあるフィールドには、注9に定めるところにより、いずれか1のフィールドを必ず設定しなければならない。

注2　「証明要求電文」を正常に受信したことを示すため、「0」を記録する。「証明要求電文」に異常があった場合には、後記5による。

注3　注16に定める電子署名を行った登記官の電子証明書の「subjectPublicKey」の値(識別子オクテット、長さオクテット及び未使用ビットを除く。)をSHA-1により変換した値を記録する。

注4　この電文を作成した日時をグリニッジ標準時により記録する。

注5　長さオクテットに「0」を記録する。

注6　「証明要求電文」中の「issuerNameHash」に記録された値を記録する。

注7　「証明要求電文」中の「issuerKeyHash」に記録された値を記録する。

注8　「証明要求電文」中の「serialNumber」に記録された値を記録する。

注9　次の表に掲げる事項に該当する場合に、設定内容に定めるフィールドを設定する。

項番	事項	設定内容
1	電子証明書について項番2又は3のいずれの事項にも該当しないこと	「good」
2	注11の表中、項番1から7までのいずれかの事項に該当すること	「revoked」
3	「証明要求電文」中「issuerNameHash」、「issuerKeyHash」若しくは「serialNumber」に誤りがあり、又は電子証明書に記録された「notAfter」の日時までに「証明要求電文」が電子認証登記所に到達しなかったこと	「unknown」

注10　注11の表中項番1から7までのいずれかの事項が記録される場合に、当該事項が生じた日時をグリニッジ標準時により記録する。

注11　次の表に掲げる「事項」に該当する場合に、「記録内容」に定める数字を記録

する。この場合において、項番1から4までの複数の事項に該当するときは、最初に生じた事項についてのみ記録する。項番5から7までのいずれかの事項が生じた時以後に、項番1から4までのいずれかの事項が生じたときは、項番1から4までの事項のうち、最初に生じた事項についてのみ記録する。

項番	事項	記録内容	
		CRL Reason	OcspStatus Code
1	商業登記法第12条の2第7項の届出があったとき	5	1
2	商業登記規則第33条の12第1項第2号の規定により電子証明書に記録された登記事項に変更を生ずる登記をした旨の通知があったとき	3	2
3	商業登記規則第33条の16第1項の規定により、登記所の事故により証明をするのが相当でなくなったと認めるとき	2	4
4	商業登記規則第33条の16第1項の規定により、登記所の事故以外の事由により証明をするのが相当でなくなったと認めるとき	5	2
5	商業登記規則第33条の12第1項第1号の通知があったとき(同項第3号の通知があったときを除く。)	6	2
6	商業登記規則第33条の13第1項の規定により電子証明書の使用の休止の届出があったとき(同条第5項の届出があったときを除く。)	6	1
7	項番5及び6のいずれにも該当するとき	6	3
8	注9により、「good」のフィールドを設定したとき		0
9	注9により、「unknown」のフィールドを設定したとき		0

注12　登記官が証明の対象を確認した日時をグリニッジ標準時により記録する。

注13　「証明要求電文」に「ConfirmationTime」が記録されていたときは、これらのフィールドを設ける。

注14　「証明要求電文」に「ConfirmationTime」が記録されていたときは、フィールドに値を記録する。

注15　「証明要求電文」の「Nonce」に記録された値を記録する。

注16　「ResponseData」をDERで符号化した値にsha-256WithRSAによる電子署名を講じた値を記録する。

注17　登記官の最新の電子証明書を記録する。

5　異常時の処理

「証明要求電文」中の「entity-body」の内容が前記4の(2)に定める形式に適合しない場合(前記4の(3)の注9により「unknown」のフィールドを設定する場合を除く。)は、登記官は、「証明応答電文」の「entity-body」に、前記4の(3)の内容に代えて、前記4の(1)の定める方式により、次の内容を記録したものを送信する。

フィールド	データ型	設定値	必須 (注1)
―	OCSPResponse		
responseStatus	OCSPResponseStatus	1(注2)	◎

注1　「必須」欄中に◎印のあるフィールドには、必ず値を記録しなければならない。

注2　「証明要求電文」に異常があったことを示すため、「1」を記録する。

附　則　(令和元年六月二十八日法務省告示第六十三号)

この告示は、令和元年七月一日から施行する。

附　則　(令和元年十一月二十六日法務省告示第百八十六号)

この告示は、令和元年十一月二十九日から施行する。

付録1　電磁的記録への記録方式(ASN.1構造とオブジェクト識別子)

```
1  Explicitly Tagged Module

MOJCMPRegistration { 1 2 392 100300 1 4 41 }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS

    CertReqMessages, GeneralName, registeredCorporationInfo
    FROM MOJCRMFRegistration { 1 2 392 100300 1 4 42 };

PKIMessage ::= SEQUENCE {
      header           PKIHeader,
      body             PKIBody
}

PKIHeader ::= SEQUENCE {
      pvno                INTEGER     { ietf-version2 (1) },
      sender              GeneralName,
      recipient           GeneralName
}

PKIBody ::= CHOICE {
      ir      [0]  CertReqMessages  --Initialization Request
}

AlgorithmIdentifier     ::=  SEQUENCE {
   algorithm            ALGORITHM-ID.&id({SupportedAlgorithms}),
   parameters           ALGORITHM-ID.&Type({SupportedAlgorithms}
                                        { @algorithm }) OPTIONAL }

ALGORITHM-ID ::= CLASS {
     &id    OBJECT IDENTIFIER UNIQUE,
     &Type  OPTIONAL
}
WITH SYNTAX { OID &id [PARMS &Type] }

SupportedAlgorithms ALGORITHM-ID ::= { ..., -- extensible
                                       rsaPublicKey |
                                       rsaSHA-256 |
                                       sha256Identifier }

rsaPublicKey ALGORITHM-ID ::= { OID rsaEncryption PARMS NULL }

rsaSHA-256 ALGORITHM-ID ::= { OID sha256WithRSAEncryption PARMS NULL }

sha256Identifier ALGORITHM-ID ::= { OID id-SHA256 PARMS NULL }

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

rsaEncryption OBJECT IDENTIFIER ::=  { pkcs-1 1 }
-- subjectPublicKey syntax
RSAPublicKey  ::=  SEQUENCE  {
     modulus INTEGER,        -- n
     publicExponent INTEGER -- e
}

sha256WithRSAEncryption OBJECT IDENTIFIER  ::=  { pkcs-1 11 }

id-SHA256 OBJECT IDENTIFIER ::= {
     joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101)
     csor(3) nistalgorithm(4) hashalgs(2) 1 }

SubjectPublicKeyInfo ::= SEQUENCE{
   algorithm            AlgorithmIdentifier,
   subjectPublicKey     BIT STRING }

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

Extension ::= SEQUENCE {
   extnId            EXTENSION.&id ({ExtensionSet}),
   critical          BOOLEAN DEFAULT FALSE,
   extnValue         OCTET STRING }

ExtensionSet  EXTENSION  ::= { registeredCorporationInfo }

EXTENSION ::= CLASS {
```

```
   &id          OBJECT IDENTIFIER UNIQUE,
   &ExtnType }
WITH SYNTAX  {
   SYNTAX               &ExtnType
   IDENTIFIED BY        &id }

AttributeTypeAndValue ::= SEQUENCE {
        type    ATTRIBUTE.&id ({SupportedAttributes}),
        value   ATTRIBUTE.&Type ({SupportedAttributes}{@type})}

Name ::= CHOICE {
        rdnSequence  RDNSequence
}

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET SIZE (1 .. MAX) OF AttributeTypeAndValue

ID ::= OBJECT IDENTIFIER

ATTRIBUTE ::= CLASS {
        &Type,
        &id     OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
        WITH SYNTAX &Type
        ID          &id }

SupportedAttributes ATTRIBUTE ::= {
        commonName | organizationName |   -- RelativeDistinguishedName attributes
        timeLimit | suspensionSecretCode  -- regInfo attributes
}

commonName ATTRIBUTE ::= {
        WITH SYNTAX        DirectoryString {ub-common-name}
        ID                 id-at-commonName }

organizationName ATTRIBUTE ::= {
        WITH SYNTAX        DirectoryString {ub-organization-name}
        ID                 id-at-organizationName }

DirectoryString { INTEGER:maxSize } ::= CHOICE {
        printableString         PrintableString (SIZE (1..maxSize)),
        utf8String              UTF8String (SIZE(1..maxSize))
}

timeLimit ATTRIBUTE ::= {
```

```
        WITH SYNTAX         TimeLimit
        ID                  id-registeredcert-mg-effectiveTimeLimit }

TimeLimit ::= OCTET STRING

suspensionSecretCode ATTRIBUTE ::= {
        WITH SYNTAX         SuspensionSecretCode
        ID                  id-registeredcert-mg-suspensionSecretCode }

SuspensionSecretCode ::= SEQUENCE {
      hashAlg                     AlgorithmIdentifier,
      hashedSecretCode            OCTET STRING
}

id-at OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 4}

id-at-commonName              OBJECT IDENTIFIER ::= {id-at 3}
id-at-organizationName        OBJECT IDENTIFIER ::= {id-at 10}

id-registeredcert OBJECT IDENTIFIER ::= { 1 2 392 100300 1 }

id-registeredcert-mg OBJECT IDENTIFIER ::= { id-registeredcert 2 }

id-registeredcert-mg-effectiveTimeLimit OBJECT IDENTIFIER ::= { id-registeredcert-mg 104 }
id-registeredcert-mg-suspensionSecretCode OBJECT IDENTIFIER ::= {
    id-registeredcert-mg 105 }

ub-common-name                INTEGER ::= 64
ub-organization-name          INTEGER ::= 64

END
```

2　Implicitly Tagged Module

```
MOJCRMFRegistration { 1 2 392 100300 1 4 42 }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS
        AttributeTypeAndValue, AlgorithmIdentifier, Name,
        SubjectPublicKeyInfo, Extensions, DirectoryString, EXTENSION
        FROM MOJCMPRegistration { 1 2 392 100300 1 4 41 };


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

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

CertRequest ::= SEQUENCE {
    certReqId     INTEGER,
    certTemplate  CertTemplate }

CertTemplate ::= SEQUENCE {
    subject      [5] Name                 OPTIONAL,
    publicKey    [6] SubjectPublicKeyInfo,
    extensions   [9] Extensions
}

ProofOfPossession ::= CHOICE {
    signature         [1] POPOSigningKey }

POPOSigningKey ::= SEQUENCE {
    algorithmIdentifier   AlgorithmIdentifier,
    signature             BIT STRING }

GeneralName ::= CHOICE {
        directoryName               [4] Name
}

registeredCorporationInfo EXTENSION ::= {
        SYNTAX  RegisteredCorporationInfoSyntax
        IDENTIFIED BY id-registeredcert-pe-registeredCorporationInfo }

RegisteredCorporationInfoSyntax ::= SEQUENCE {
  corporateName              [0] EXPLICIT DirectoryString{ub-corporate-name},
  corporateAddress           [2] EXPLICIT DirectoryString{ub-corporate-address},
  representativeDirectorName  [3] EXPLICIT DirectoryString
                                                {ub-representative-director-name},
  representativeDirectorTitle [4] EXPLICIT DirectoryString
                                                {ub-representative-director-title}
}

id-registeredcert OBJECT IDENTIFIER ::= { 1 2 392 100300 1 }

id-registeredcert-pe OBJECT IDENTIFIER ::=  { id-registeredcert 1 }

id-registeredcert-pe-registeredCorporationInfo OBJECT IDENTIFIER ::= {
     id-registeredcert-pe 3 }

ub-corporate-name               INTEGER ::= 128
ub-corporate-address            INTEGER ::= 128
ub-representative-director-name   INTEGER ::= 126
ub-representative-director-title  INTEGER ::= 128

END
```

付録2　電子証明書の方式(ASN.1構造とオブジェクト識別子)

1　Explicitly Tagged Module

```
MOJCorpCertExplicit { 1 2 392 100300 1 4 1 }
DEFINITIONS EXPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS
        authorityKeyIdentifier, subjectKeyIdentifier, keyUsage,
        privateKeyUsagePeriod, certificatePolicies,
        basicConstraints, authorityInfoAccess, jCertificatePolicies,
        registrar, registeredCorporationInfo
        FROM MOJCorpCertImplicit { 1 2 392 100300 1 4 2 };


Certificate  ::=  SIGNED { TBSCertificate }

TBSCertificate  ::=  SEQUENCE {
   version                 [0]   Version,
   serialNumber                  CertificateSerialNumber,
   signature                     AlgorithmIdentifier,
   issuer                        Name,
   validity                      Validity,
   subject                       Name,
   subjectPublicKeyInfo          SubjectPublicKeyInfo,
   extensions              [3]   Extensions  }

Version  ::=  INTEGER { v3(2) }

CertificateSerialNumber ::= INTEGER

Validity ::= SEQUENCE {
   notBefore            Time,
   notAfter             Time }

Time ::= CHOICE {
        utcTime         UTCTime
}

SubjectPublicKeyInfo ::= SEQUENCE{
   algorithm            AlgorithmIdentifier,
   subjectPublicKey     BIT STRING }

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

Extension ::= SEQUENCE {
   extnId            EXTENSION.&id ({ExtensionSet}),
   critical          BOOLEAN DEFAULT FALSE,
   extnValue         OCTET STRING }

ExtensionSet  EXTENSION  ::= { authorityKeyIdentifier |
                               subjectKeyIdentifier |
                               keyUsage |
                               privateKeyUsagePeriod |
                               certificatePolicies |
                               basicConstraints |
                               authorityInfoAccess |
                               jCertificatePolicies |
                               registrar |
                               registeredCorporationInfo }

EXTENSION ::= CLASS {
   &id          OBJECT IDENTIFIER UNIQUE,
   &ExtnType }
WITH SYNTAX  {
   SYNTAX               &ExtnType
   IDENTIFIED BY        &id }

SIGNED { ToBeSigned } ::= SEQUENCE {
     toBeSigned  ToBeSigned,
     algorithm   AlgorithmIdentifier,
     signature   BIT STRING
}

AlgorithmIdentifier  ::=  SEQUENCE {
   algorithm            ALGORITHM-ID.&id({SupportedAlgorithms}),
   parameters           ALGORITHM-ID.&Type({SupportedAlgorithms}
                                          { @algorithm}) OPTIONAL }

ALGORITHM-ID ::= CLASS {
```

```
    &id    OBJECT IDENTIFIER UNIQUE,
    &Type  OPTIONAL
}
WITH SYNTAX { OID &id [PARMS &Type] }

SupportedAlgorithms ALGORITHM-ID ::= { ..., -- extensible
                                       rsaPublicKey |
                                       rsaSHA-256 }

rsaPublicKey ALGORITHM-ID ::= { OID rsaEncryption PARMS NULL }

rsaSHA-256 ALGORITHM-ID ::= { OID sha256WithRSAEncryption PARMS NULL }

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

rsaEncryption OBJECT IDENTIFIER ::=  { pkcs-1 1 }

-- subjectPublicKey syntax
RSAPublicKey  ::=  SEQUENCE  {
    modulus INTEGER, -- n
    publicExponent INTEGER -- e
}

sha256WithRSAEncryption OBJECT IDENTIFIER  ::=  { pkcs-1 11 }

AttributeTypeAndValue ::= SEQUENCE {
        type    ATTRIBUTE.&id ({SupportedAttributes}),
        value   ATTRIBUTE.&Type ({SupportedAttributes}{@type})}

Name ::= CHOICE {
        rdnSequence  RDNSequence
}

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET SIZE (1 .. MAX) OF AttributeTypeAndValue

ID ::= OBJECT IDENTIFIER

ATTRIBUTE ::= CLASS {
```

```
        &Type,
        &id     OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
        WITH SYNTAX &Type
        ID          &id }

SupportedAttributes ATTRIBUTE ::= {
        commonName | countryName | organizationName | organizationalUnitName }

commonName ATTRIBUTE ::= {
        WITH SYNTAX        DirectoryString {ub-common-name}
        ID                 id-at-commonName }

countryName ATTRIBUTE ::= {
        WITH SYNTAX        PrintableString (SIZE (2))
        ID                 id-at-countryName }

organizationName ATTRIBUTE ::= {
        WITH SYNTAX        DirectoryString {ub-organization-name}
        ID                 id-at-organizationName }

organizationalUnitName ATTRIBUTE ::= {
        WITH SYNTAX        DirectoryString {ub-organizational-unit-name}
        ID                 id-at-organizationalUnitName }

id-at OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 4}

id-at-commonName              OBJECT IDENTIFIER ::= {id-at 3}
id-at-countryName             OBJECT IDENTIFIER ::= {id-at 6}
id-at-organizationName        OBJECT IDENTIFIER ::= {id-at 10}
id-at-organizationalUnitName  OBJECT IDENTIFIER ::= {id-at 11}

DirectoryString { INTEGER:maxSize } ::= CHOICE {
        printableString         PrintableString (SIZE (1..maxSize)),
        utf8String              UTF8String (SIZE(1..maxSize))
}

ub-common-name               INTEGER ::= 64
ub-organization-name         INTEGER ::= 64
ub-organizational-unit-name  INTEGER ::= 64

END
```

2　Implicitly Tagged Module

```
MOJCorpCertImplicit { 1 2 392 100300 1 4 2 }
```

```
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

--EXPORTS ALL --

IMPORTS
        Name, CertificateSerialNumber, DirectoryString, EXTENSION
        FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 };


authorityKeyIdentifier EXTENSION ::= {
        SYNTAX          AuthorityKeyIdentifier
        IDENTIFIED BY   id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier,
    authorityCertIssuer         [1] GeneralNames,
    authorityCertSerialNumber   [2] CertificateSerialNumber }

KeyIdentifier ::= OCTET STRING

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

GeneralName ::= CHOICE {
        directoryName               [4] Name,
        uniformResourceIdentifier   [6] IA5String
}

subjectKeyIdentifier EXTENSION ::= {
        SYNTAX          SubjectKeyIdentifier
        IDENTIFIED BY   id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier

keyUsage EXTENSION ::= {
        SYNTAX KeyUsage
        IDENTIFIED BY id-ce-keyUsage }

KeyUsage ::= BIT STRING {
        digitalSignature     (0),
        keyEncipherment      (2),
        dataEncipherment     (3),
        keyCertSign          (5),
        cRLSign              (6)
}
```

```
privateKeyUsagePeriod EXTENSION ::= {
        SYNTAX  PrivateKeyUsagePeriod
        IDENTIFIED BY { id-ce-privateKeyUsagePeriod } }

PrivateKeyUsagePeriod ::= SEQUENCE {
        notBefore       [0]     GeneralizedTime,
        notAfter        [1]     GeneralizedTime }

certificatePolicies EXTENSION ::= {
        SYNTAX  CertificatePoliciesSyntax
        IDENTIFIED BY id-ce-certificatePolicies }

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

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

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
        policyQualifierId       CERT-POLICY-QUALIFIER.&id
                                    ({SupportedPolicyQualifiers}),
        qualifier               CERT-POLICY-QUALIFIER.&Qualifier
                                    ({SupportedPolicyQualifiers}
                                    {@policyQualifierId}) }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { noticeToUser }

CERT-POLICY-QUALIFIER ::= CLASS {
        &id             OBJECT IDENTIFIER UNIQUE,
        &Qualifier  }
WITH SYNTAX {
        POLICY-QUALIFIER-ID     &id
        QUALIFIER-TYPE          &Qualifier }

noticeToUser CERT-POLICY-QUALIFIER ::= {
     POLICY-QUALIFIER-ID id-qt-unotice  QUALIFIER-TYPE   UserNotice }

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

NoticeReference ::= SEQUENCE {
```

```
    organization     DisplayText,
    noticeNumbers    SEQUENCE OF INTEGER }

DisplayText ::= CHOICE {
    visibleString    VisibleString  (SIZE (1..200)),
    utf8String       UTF8String     (SIZE (1..200)) }

basicConstraints EXTENSION ::= {
        SYNTAX  BasicConstraintsSyntax
        IDENTIFIED BY id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
        cA              BOOLEAN DEFAULT FALSE
}

authorityInfoAccess EXTENSION ::= {
        SYNTAX  AuthorityInfoAccessSyntax
        IDENTIFIED BY id-pe-authorityInfoAccess }

AuthorityInfoAccessSyntax  ::=
        SEQUENCE SIZE (1..MAX) OF AccessDescription

AccessDescription  ::=  SEQUENCE {
        accessMethod          OBJECT IDENTIFIER,
        accessLocation        GeneralName  }

jCertificatePolicies EXTENSION ::= {
        SYNTAX  JCertificatePoliciesSyntax
        IDENTIFIED BY id-registeredcert-pe-jCertificatePolicies }

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

registrar EXTENSION ::= {
        SYNTAX  RegistrarSyntax
        IDENTIFIED BY id-registeredcert-pe-registrar }

RegistrarSyntax ::= DirectoryString{ub-registrar}

registeredCorporationInfo EXTENSION ::= {
        SYNTAX  RegisteredCorporationInfoSyntax
        IDENTIFIED BY id-registeredcert-pe-registeredCorporationInfo }

RegisteredCorporationInfoSyntax ::= SEQUENCE {
  corporateName               [0] EXPLICIT DirectoryString{ub-corporate-name},
  registeredNumber            [1] EXPLICIT PrintableString,
  corporateAddress            [2] EXPLICIT DirectoryString{ub-corporate-address},
```

```
  representativeDirectorName   [3] EXPLICIT DirectoryString
                                    (ub-representative-director-name),
  representativeDirectorTitle  [4] EXPLICIT DirectoryString
                                    (ub-representative-director-title),
  registryOffice               [6] EXPLICIT DirectoryString(ub-registry-office)
}


id-ce   OBJECT IDENTIFIER ::= (joint-iso-ccitt(2) ds(5) 29)

id-ce-subjectKeyIdentifier         OBJECT IDENTIFIER ::= (id-ce 14)
id-ce-keyUsage                     OBJECT IDENTIFIER ::= (id-ce 15)
id-ce-privateKeyUsagePeriod        OBJECT IDENTIFIER ::= (id-ce 16)
id-ce-basicConstraints             OBJECT IDENTIFIER ::= (id-ce 19)
id-ce-certificatePolicies          OBJECT IDENTIFIER ::= (id-ce 32)
id-ce-authorityKeyIdentifier       OBJECT IDENTIFIER ::= (id-ce 35)

id-pkix OBJECT IDENTIFIER ::=
  { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

id-pe   OBJECT IDENTIFIER ::= { id-pkix 1 }

id-pe-authorityInfoAccess OBJECT IDENTIFIER ::= { id-pe 1 }

id-qt   OBJECT IDENTIFIER ::= { id-pkix 2 }

id-qt-unotice  OBJECT IDENTIFIER ::=  { id-qt 2 }

id-ad   OBJECT IDENTIFIER ::= { id-pkix 48 }

id-ad-ocsp      OBJECT IDENTIFIER ::= { id-ad 1 }

id-registeredcert OBJECT IDENTIFIER ::= { 1 2 392 100300 1 }

id-registeredcert-pe OBJECT IDENTIFIER ::=  { id-registeredcert 1 }

id-registeredcert-pe-jCertificatePolicies OBJECT IDENTIFIER ::= (id-registeredcert-pe 1 }
id-registeredcert-pe-registrar OBJECT IDENTIFIER ::= { id-registeredcert-pe 2 }
id-registeredcert-pe-registeredCorporationInfo OBJECT IDENTIFIER ::= {
     id-registeredcert-pe 3 }

ub-registrar                    INTEGER ::= 128
ub-corporate-name               INTEGER ::= 128
ub-corporate-address            INTEGER ::= 128
ub-representative-director-name INTEGER ::= 126
ub-representative-director-title INTEGER ::= 128
ub-registry-office              INTEGER ::= 128

END
```

付録3　電子証明書の送信の方式(送受信電文のASN.1構造とオブジェクト識別子)

```
1  Explicitly Tagged Module

MOJCMPCertReq { 1 2 392 100300 1 4 11 }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS

      Certificate
            FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 }

      GeneralName, CertReqMessages, EncryptedValue
            FROM MOJCRMFCertReq { 1 2 392 100300 1 4 12 };

AlgorithmIdentifier ::= SEQUENCE {
   algorithm            ALGORITHM-ID.&id ({SupportedAlgorithms}),
   parameters           ALGORITHM-ID.&Type ({SupportedAlgorithms}
                                    {@algorithm }) OPTIONAL }

ALGORITHM-ID ::= CLASS {
     &id    OBJECT IDENTIFIER UNIQUE,
     &Type  OPTIONAL
}
WITH SYNTAX { OID &id [PARMS &Type] }

SupportedAlgorithms ALGORITHM-ID ::= { ..., -- extensible
                                    rsaPublicKey |
                                    rsaSHA-256 |
                                    des-EDE3-CBC |
                                    des-EDE3-CBC-NoParms |
                                    sha256Identifier }

rsaPublicKey ALGORITHM-ID ::= { OID rsaEncryption PARMS NULL }

rsaSHA-256 ALGORITHM-ID ::= { OID sha256WithRSAEncryption PARMS NULL }

des-EDE3-CBC ALGORITHM-ID ::= { OID dES-EDE3-CBC PARMS CBCParameter }

des-EDE3-CBC-NoParms ALGORITHM-ID ::= { OID dES-EDE3-CBC PARMS NULL }

sha256Identifier ALGORITHM-ID ::= { OID id-SHA256 PARMS NULL }


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

rsaEncryption OBJECT IDENTIFIER ::=  { pkcs-1 1 }
-- Public key syntax
RSAPublicKey  ::=  SEQUENCE  {
     modulus INTEGER, -- n
     publicExponent INTEGER -- e
}

sha256WithRSAEncryption OBJECT IDENTIFIER  ::=  { pkcs-1 11 }

dES-EDE3-CBC OBJECT IDENTIFIER  ::=  {
     iso(1) member-body(2) us(840) rsadsi(113549) encryptionAlgorithm(3) 7 }
-- dES-EDE3-CBC parameters
CBCParameter ::= IV
IV ::= OCTET STRING (SIZE (8..8))

id-SHA256 OBJECT IDENTIFIER ::= {
     joint-iso-itu-t(2) country(16) us(840) organization(1)
     gov(101) csor(3) nistalgorithm(4) hashalgs(2) 1 }


InfoTypeAndValue ::= SEQUENCE {
      infoType       INFORMATION-ID.&id({InfoSet}),
      infoValue      INFORMATION-ID.&Type({InfoSet}{@infoType})
}

INFORMATION-ID ::= CLASS {
     &id    OBJECT IDENTIFIER UNIQUE,
```

```
    &Type
}
WITH SYNTAX { SYNTAX &Type IDENTIFIED BY &id }

InfoSet INFORMATION-ID ::= { genmInfoReqContent |
                             genpInfoResContent
}

genmInfoReqContent INFORMATION-ID ::= {
  SYNTAX         GenmInfoReqContent
  IDENTIFIED BY  id-registeredcert-mg-genminforeq
}

genpInfoResContent INFORMATION-ID ::= {
  SYNTAX         GenmInfoReqContent
  IDENTIFIED BY  id-registeredcert-mg-genpinfores
}

PKIMessage ::= SEQUENCE {
      header           PKIHeader,
      body             PKIBody,
      protection   [0] PKIProtection OPTIONAL,
      extraCerts   [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL
}

PKIHeader ::= SEQUENCE {
      pvno                INTEGER     { ietf-version2 (1) },
      sender              GeneralName,
      recipient           GeneralName,
      protectionAlg   [1] AlgorithmIdentifier     OPTIONAL,
      senderKID       [2] KeyIdentifier           OPTIONAL,
      transactionID   [4] OCTET STRING,
      senderNonce     [5] OCTET STRING,
      recipNonce      [6] OCTET STRING            OPTIONAL
}

PKIBody ::= CHOICE {       -- message-specific body elements
      ir       [0]  CertReqMessages,        --Initialization Request
      ip       [1]  CertRepMessage,         --Initialization Response
      genm     [21] GenMsgContent,          --General Message
      genp     [22] GenRepContent,          --General Response
      error    [23] ErrorMsgContent         --Error Message
}

PKIProtection ::= BIT STRING
```

```
ProtectedPart ::= SEQUENCE {
      header    PKIHeader,
      body      PKIBody
}

KeyIdentifier ::= OCTET STRING

PKIStatus ::= INTEGER {
      granted                 (0),
      -- you got exactly what you asked for
      rejection               (2)
      -- you don't get it, more information elsewhere in the message
}

PKIStatusInfo ::= SEQUENCE {
      status        PKIStatus
}

CertRepMessage ::= SEQUENCE {
      response         SEQUENCE OF CertResponse
}

CertResponse ::= SEQUENCE {
      certReqId           INTEGER,
      status              PKIStatusInfo,
      certifiedKeyPair    CertifiedKeyPair
}

CertifiedKeyPair ::= SEQUENCE {
      certOrEncCert       CertOrEncCert
}

CertOrEncCert ::= CHOICE {
      encryptedCert   [1] EncryptedValue
}

GenMsgContent ::= SEQUENCE OF InfoTypeAndValue

GenRepContent ::= SEQUENCE OF InfoTypeAndValue

ErrorMsgContent ::= SEQUENCE {
      pKIStatusInfo          PKIStatusInfo
}

id-registeredcert OBJECT IDENTIFIER ::= { 1 2 392 100300 1 }
```

```
id-registeredcert-mg OBJECT IDENTIFIER ::= { id-registeredcert 2 }

id-registeredcert-mg-genminforeq OBJECT IDENTIFIER ::= { id-registeredcert-mg 21 }
id-registeredcert-mg-genpinfores OBJECT IDENTIFIER ::= { id-registeredcert-mg 22 }

GenmInfoReqContent ::= SEQUENCE OF NegotiationKey

NegotiationKey ::= SEQUENCE {
     symmAlg  AlgorithmIdentifier,
     pubAlg   AlgorithmIdentifier,
     hashAlg  AlgorithmIdentifier
}

GenpInfoResContent ::= SEQUENCE {
     status           PKIStatusInfo,
     negotiationKeys  SEQUENCE OF NegotiationKey OPTIONAL
}

END
```

2　Implicitly Tagged Module

```
MOJCRMFCertReq { 1 2 392 100300 1 4 12 }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS

        AlgorithmIdentifier
        FROM MOJCMPCertReq { 1 2 392 100300 1 4 11 }

        SubjectPublicKeyInfo, Name, RDNSequence
        FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 };

CertReqMessages ::= SEQUENCE SIZE (1..MAX) OF CertReqMsg

CertReqMsg ::= SEQUENCE {
    certReq   CertRequest,
    pop       ProofOfPossession }

CertRequest ::= SEQUENCE {
    certReqId     INTEGER,
    certTemplate  CertTemplate }

CertTemplate ::= SEQUENCE {

    serialNumber  [1] INTEGER,
    publicKey     [6] SubjectPublicKeyInfo }

ProofOfPossession ::= CHOICE {
    signature             [1] POPOSigningKey }

POPOSigningKey ::= SEQUENCE {
    algorithmIdentifier   AlgorithmIdentifier,
    signature             BIT STRING }

EncryptedValue ::= SEQUENCE {
    symmAlg       [1] AlgorithmIdentifier,
    encSymmKey    [2] BIT STRING,
    keyAlg        [3] AlgorithmIdentifier,
    encValue       BIT STRING }

GeneralName ::= CHOICE {
    directoryName         [4] Name }

END
```

付録4　休止届の送信の方式(送受信電文のASN.1構造とオブジェクト識別子)

```
1  Explicitly Tagged Module

MOJCMPSuspReq { 1 2 392 100300 1 4 21 }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS

      Certificate
      FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 }

      GeneralName, EncryptedValue, CertTemplate, CertId, ReasonFlags
      FROM MOJCRMFSuspReq { 1 2 392 100300 1 4 22 };


AlgorithmIdentifier     ::=  SEQUENCE {
   algorithm            ALGORITHM-ID.&id({SupportedAlgorithms}),
   parameters           ALGORITHM-ID.&Type({SupportedAlgorithms}
                                          { @algorithm }) OPTIONAL }

ALGORITHM-ID ::= CLASS {

     &id    OBJECT IDENTIFIER UNIQUE,
     &Type  OPTIONAL
}
WITH SYNTAX { OID &id [PARMS &Type] }

SupportedAlgorithms  ALGORITHM-ID  ::=  { ..., -- extensible
                                          rsaPublicKey |
                                          rsaSHA-256 |
                                          des-EDE3-CBC |
                                          sha256Identifier }

rsaPublicKey ALGORITHM-ID ::= { OID rsaEncryption PARMS NULL }

rsaSHA-256 ALGORITHM-ID ::= { OID sha256WithRSAEncryption PARMS NULL }

des-EDE3-CBC ALGORITHM-ID ::= { OID dES-EDE3-CBC PARMS CBCParameter }

sha256Identifier ALGORITHM-ID ::= { OID id-SHA256 PARMS NULL }

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

rsaEncryption OBJECT IDENTIFIER ::=  { pkcs-1 1 }
-- subjectPublicKey syntax
RSAPublicKey  ::=  SEQUENCE  {
     modulus INTEGER, -- n
     publicExponent INTEGER -- e
}

sha256WithRSAEncryption OBJECT IDENTIFIER  ::=  { pkcs-1 11 }

dES-EDE3-CBC OBJECT IDENTIFIER  ::=  {
     iso(1) member-body(2) us(840) rsadsi(113549) encryptionAlgorithm(3) 7 }
-- dES-EDE3-CBC parameters
CBCParameter ::= IV
IV ::= OCTET STRING (SIZE (8..8))
```

```
id-SHA256 OBJECT IDENTIFIER ::= {
     joint-iso-itu-t(2) country(16) us(840) organization(1)
     gov(101) csor(3) nistalgorithm(4) hashalgs(2) 1 }

InfoTypeAndValue ::= SEQUENCE {
      infoType       INFORMATION-ID.&id({InfoSet}),
      infoValue      INFORMATION-ID.&Type({InfoSet}{@infoType})
}

INFORMATION-ID ::= CLASS {
     &id    OBJECT IDENTIFIER UNIQUE,
     &Type
}
WITH SYNTAX { SYNTAX &Type IDENTIFIED BY &id }

InfoSet INFORMATION-ID ::= { genmSuspReqContent |
                             genpSuspResContent |
                             genmInfoReqContent |
                             genpInfoResContent
}

genmSuspReqContent INFORMATION-ID ::= {
  SYNTAX        GenmSuspReqContent
  IDENTIFIED BY id-registeredcert-mg-genmsuspreq
}

genpSuspResContent INFORMATION-ID ::= {
  SYNTAX        GenpSuspResContent
  IDENTIFIED BY id-registeredcert-mg-genpsuspres
}

genmInfoReqContent INFORMATION-ID ::= {
  SYNTAX        GenmInfoReqContent
  IDENTIFIED BY id-registeredcert-mg-genminforeq
}

genpInfoResContent INFORMATION-ID ::= {
  SYNTAX        GenpInfoResContent
  IDENTIFIED BY id-registeredcert-mg-genpinfores
}

PKIMessage ::= SEQUENCE {
      header           PKIHeader,
      body             PKIBody,
```

```
      protection   [0] PKIProtection OPTIONAL,
      extraCerts   [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL
}

PKIHeader ::= SEQUENCE {
      pvno                INTEGER     { ietf-version2 (1) },
      sender              GeneralName,
      recipient           GeneralName,
      protectionAlg   [1] AlgorithmIdentifier     OPTIONAL,
      senderKID       [2] KeyIdentifier           OPTIONAL,
      transactionID   [4] OCTET STRING,
      senderNonce     [5] OCTET STRING,
      recipNonce      [6] OCTET STRING            OPTIONAL
}

PKIBody ::= CHOICE {       -- message-specific body elements
      genm    [21] GenMsgContent,          --General Message
      genp    [22] GenRepContent,          --General Response
      error   [23] ErrorMsgContent         --Error Message
}

PKIProtection ::= BIT STRING

ProtectedPart ::= SEQUENCE {
      header    PKIHeader,
      body      PKIBody
}

PKIStatus ::= INTEGER {
      granted                (0),
      -- you got exactly what you asked for
      rejection              (2)
      -- you don't get it, more information elsewhere in the message
}

KeyIdentifier ::= OCTET STRING

PKIStatusInfo ::= SEQUENCE {
      status        PKIStatus
}

GenMsgContent ::= SEQUENCE OF InfoTypeAndValue

GenRepContent ::= SEQUENCE OF InfoTypeAndValue

ErrorMsgContent ::= SEQUENCE {
```

```
      pKIStatusInfo          PKIStatusInfo
}

id-registeredcert OBJECT IDENTIFIER ::= { 1 2 392 100300 1 }

id-registeredcert-mg OBJECT IDENTIFIER ::= { id-registeredcert 2 }

id-registeredcert-mg-genmsuspreq OBJECT IDENTIFIER ::= { id-registeredcert-mg 1 }
id-registeredcert-mg-genpsuspres OBJECT IDENTIFIER ::= { id-registeredcert-mg 2 }
id-registeredcert-mg-genminforeq OBJECT IDENTIFIER ::= { id-registeredcert-mg 21 }
id-registeredcert-mg-genpinfores OBJECT IDENTIFIER ::= { id-registeredcert-mg 22 }

GenmSuspReqContent ::= SEQUENCE {
      certDetails           CertTemplate,
      revocationReason      ReasonFlags,
      suspensionReasonCode  INTEGER,
      suspensionDetail      EncryptedValue  -- encrypted SuspData
}
-- SuspData is made with conection of
-- "suspensionSecretCode" (which is password without tag and length) and
-- hashed "header PKIHeader".
-- The max size of Suspdata is 84Bytes. (Max64Bytes + 20Bytes).

GenpSuspResContent ::= SEQUENCE {
      status         PKIStatusInfo,
      -- status information of suspension results
      revCert        CertId
      -- IDs for which revocation was requested (same order as status)
}

GenmInfoReqContent ::= SEQUENCE OF NegotiationKey

NegotiationKey ::= SEQUENCE {
     symmAlg  AlgorithmIdentifier,
     pubAlg   AlgorithmIdentifier,
     hashAlg  AlgorithmIdentifier  }

GenpInfoResContent ::= SEQUENCE {
     status           PKIStatusInfo,
     negotiationKeys  SEQUENCE OF NegotiationKey OPTIONAL  }

END
```

2　Implicitly Tagged Module

```
MOJCRMFSuspReq { 1 2 392 100300 1 4 22 }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS
        AlgorithmIdentifier
        FROM MOJCMPSuspReq { 1 2 392 100300 1 4 21 }

        SubjectPublicKeyInfo, Name, RDNSequence
        FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 };


CertTemplate ::= SEQUENCE {
    serialNumber [1] INTEGER,
    issuer       [3] Name }

EncryptedValue ::= SEQUENCE {
    keyAlg        [3] AlgorithmIdentifier,
    encValue       BIT STRING }

CertId ::= SEQUENCE {
    issuer           GeneralName,
    serialNumber     INTEGER }

GeneralName ::= CHOICE {
        directoryName              [4] Name  }

ReasonFlags ::= BIT STRING {
        certificateHold         (6) }

END
```

付録5　電子証明書に係る証明及びその請求の方式(送受信電文のASN.1構造とオブジェクト識別子)

```
MOJOCSP { 1 2 392 100300 1 4 31 }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS

         Certificate
         FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 };

OCSPRequest ::=  SEQUENCE {
    tbsRequest                  TBSRequest  }

TBSRequest ::= SEQUENCE {
    version             [0] EXPLICIT Version DEFAULT v1,
    requestList             SEQUENCE OF Request,
    requestExtensions   [2] EXPLICIT Extensions  -- nonce
}

Version ::= INTEGER  { v1(0) }

Request ::= SEQUENCE {
    reqCert                     CertID,
    singleRequestExtensions     [0] EXPLICIT Extensions OPTIONAL  -- confirmationTime
}

CertID ::= SEQUENCE {
    hashAlgorithm       AlgorithmIdentifier,
    issuerNameHash      OCTET STRING, -- Hash of Issuer's DN
    issuerKeyHash       OCTET STRING, -- Hash of Issuers public key
    serialNumber        CertificateSerialNumber }

CertificateSerialNumber ::= INTEGER

OCSPResponse ::= SEQUENCE {
   responseStatus         OCSPResponseStatus,
   responseBytes          [0] EXPLICIT ResponseBytes OPTIONAL }

OCSPResponseStatus ::= ENUMERATED {
    successful            (0),      --Response has valid confirmations
    malformedRequest      (1)       --Illegal confirmation request
}

ResponseBytes ::=       SEQUENCE {
    responseType   RESPONSE.&Type({SupportedResponses}),
    response       RESPONSE.&Value({SupportedResponses}{ @responceType })
}

RESPONSE ::= CLASS {
     &Type    OBJECT IDENTIFIER UNIQUE,
     &Value
}
WITH SYNTAX { SYNTAX &Value IDENTIFIED BY &Type }

SupportedResponses RESPONSE ::= { ..., -- extensible
                                  basicOCSPResponse }
```

```
basicOCSPResponse RESPONSE ::= {
  SYNTAX          BasicOCSPResponse
  IDENTIFIED BY   id-pkix-ocsp-basic
}

BasicOCSPResponse ::= SEQUENCE {
    tbsResponseData      ResponseData,
    signatureAlgorithm   AlgorithmIdentifier,
    signature            BIT STRING,
    certs                [0] EXPLICIT SEQUENCE OF Certificate }

ResponseData ::= SEQUENCE {
    version              [0] EXPLICIT Version DEFAULT v1,
    responderID              ResponderID,
    producedAt               GeneralizedTime,
    responses                SEQUENCE OF SingleResponse,
    responseExtensions   [1] EXPLICIT Extensions   -- nonce
}

ResponderID ::= CHOICE {
    byKey    [2] KeyHash }

KeyHash ::= OCTET STRING -- SHA-1 hash of responder's public key
                         -- (excluding the tag and length fields)

SingleResponse ::= SEQUENCE {
    certID                       CertID,
    certStatus                   CertStatus,
    thisUpdate                   GeneralizedTime,
    singleExtensions     [1]     EXPLICIT Extensions
                                 -- confirmationTime and ocspStatusCode
}

CertStatus ::= CHOICE {
    good                 [0]     IMPLICIT NULL,
    revoked              [1]     IMPLICIT RevokedInfo,
    unknown              [2]     IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE {
    revocationTime               GeneralizedTime,
    revocationReason     [0]     EXPLICIT CRLReason }

UnknownInfo ::= NULL

AlgorithmIdentifier     ::=  SEQUENCE {
```

```
   algorithm            ALGORITHM-ID.&id({SupportedAlgorithms}),
   parameters           ALGORITHM-ID.&Type({SupportedAlgorithms}
                                          { @algorithm }) }

ALGORITHM-ID ::= CLASS {
     &id    OBJECT IDENTIFIER UNIQUE,
     &Type
}
WITH SYNTAX { OID &id PARMS &Type }

SupportedAlgorithms ALGORITHM-ID ::= { ..., -- extensible
                                       rsaSHA-256 }

rsaSHA-256 ALGORITHM-ID ::= { OID sha256WithRSAEncryption PARMS NULL }

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

sha256WithRSAEncryption OBJECT IDENTIFIER  ::=  { pkcs-1 11 }

CRLReason ::= ENUMERATED {
        cACompromise            (2),
        affiliationChanged      (3),
        cessationOfOperation    (5),
        certificateHold         (6)
}

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

Extension ::= SEQUENCE {
   extnId            EXTENSION.&id ({ExtensionSet}),
   critical          BOOLEAN DEFAULT FALSE,
   extnValue         OCTET STRING }

ExtensionSet  EXTENSION  ::= { ..., -- extensible
                               nonce |
                               confirmationTime |
                               ocspStatusCode
}

EXTENSION ::= CLASS {
```

```
   &id          OBJECT IDENTIFIER UNIQUE,
   &ExtnType
}
WITH SYNTAX  {
   SYNTAX        &ExtnType
   IDENTIFIED BY  &id }

nonce EXTENSION ::= {
        SYNTAX  Nonce
        IDENTIFIED BY id-pkix-ocsp-nonce }

Nonce ::= OCTET STRING

confirmationTime EXTENSION ::= {
        SYNTAX  ConfirmationTime
        IDENTIFIED BY id-registeredcert-mg-confirmationtime }

ConfirmationTime ::= GeneralizedTime

ocspStatusCode EXTENSION ::= {
        SYNTAX  OcspStatusCode
        IDENTIFIED BY id-registeredcert-mg-ocspstatuscode }

OcspStatusCode ::= INTEGER

id-pkix OBJECT IDENTIFIER ::=
  { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

id-ad   OBJECT IDENTIFIER ::= { id-pkix 48 }

id-ad-ocsp      OBJECT IDENTIFIER ::= { id-ad 1 }

id-pkix-ocsp                OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-basic          OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }
id-pkix-ocsp-nonce          OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }
-- The nonce will be identified by the object identifier id-pkix-ocsp-nonce,
-- while the extnValue is the value of the nonce.

id-registeredcert OBJECT IDENTIFIER ::= { 1 2 392 100300 1 }

id-registeredcert-mg OBJECT IDENTIFIER ::= { id-registeredcert 2 }

id-registeredcert-mg-confirmationtime OBJECT IDENTIFIER ::= { id-registeredcert-mg 102 }
id-registeredcert-mg-ocspstatuscode OBJECT IDENTIFIER ::= { id-registeredcert-mg 103 }

END
```