3. 外部仕様

本章ではXML署名ライブラリの各機能に対し、それらが満たすべき仕様の詳細を説明する。

3.1 XAdES-T XML 署名の作成

3.1.1 XAdES-T に準拠した、タイムスタンプを含むXML 署名の作成

XML署名のフォーマットとしてはXAdES-BESを、また、それへのタイムスタンプ付加のフォーマットとして、XAdES-Tを採用する。この時、フォーマットとして次の制約を想定する。

- XAdES-BESのプロパティはSigningCertificateエレメント（下表のId 26）及びSigningTimeエレメント（25）のみを使用する。
- 署名者の公開鍵証明書のみをKeyInfoエレメント（15）にセットする。
- その公開鍵証明書の参照をSigningCertificateエレメント（26）にセットする。[5]

下記の表は、本XML署名ライブラリが扱うXAdES XML署名の構造を表す（XML署名のサンプルは『表4-1 XAdES XML署名のサンプル』を参照）。

表3-1 XAdES XML署名の構造

Id	Element	Attribute	説明
1	ds:Signature		XML署名。
1.1		Id	Signatureエレメント(1)のId。
2	ds:SignedInfo		XML署名の署名値(15)の計算対象。
3	ds:CanonicalizationMethod		署名計算時の、SignedInfoエレメント(2)の正規化アルゴリズムを指定する。
3.1		Algorithm	正規化アルゴリズムのURI。
4	ds:SignatureMethod		署名計算時の、署名アルゴリズムを指定する。
4.1		Algorithm	署名アルゴリズムのURI。
5	ds:Reference		署名対象のユーザドキュメントへの参照を定義する。
5.1		URI	署名対象のユーザドキュメントを格納するObjectエレメントのId(16.1)を参照する。
6	ds:Transforms		Referenceエレメント(5)のハッシュ計算に使用する、正規化アルゴリズムを指定する。
7	ds:Transform		
7.1		Algorithm	正規化アルゴリズムのURI。
8	ds:DigestMethod		Referenceエレメント(5)のハッシュ計算に使用する、ハッシュアルゴリズムを指定する。
8.1		Algorithm	ハッシュアルゴリズムのURI。
9	ds:DigestValue		Referenceエレメント(5)のハッシュ値。
10	ds:Reference		署名対象のXAdESプロパティエレメント(SignedProperties, 23)への参照を定義する。
10.1		Type	"http://uri.etsi.org/01903/V1.3.1#SignedProperties"(固定値)
10.2		URI	署名対象のXAdESプロパティエレメントのId(23.1)を参照する。
11	ds:Transforms		Referenceエレメント(10)のハッシュ計算に使用する、正規化アルゴリズムを指定する。
12	ds:Transform		
12.1		Algorithm	正規化アルゴリズムのURI。

[5] 『XAdES長期署名プロファイル』p.15

13	ds:DigestMethod		Referenceエレメント(10)のハッシュ計算に使用する、ハッシュアルゴリズムを指定する。
13.1		Algorithm	ハッシュアルゴリズムのURI。
14	ds:DigestValue		Referenceエレメント(10)のハッシュ値。
15	ds:KeyInfo		署名鍵情報。
16	ds:X509Data		署名者の公開鍵証明書。
17	ds:X509Certificate		署名者の公開鍵証明書。
18	ds:SignatureValue		XML署名の署名値。
19	ds:Object		署名対象のユーザドキュメントを格納する。
19.1		Id	Objectエレメント(19)のId。
20	UserDocument		署名対象のユーザドキュメント。
21	ds:Object		
22	XAdES:QualifyingProperties		XAdESプロパティ。
22.1		Target	プロパティが対象とするSignatureエレメント(1)のIdを参照する。
23	XAdES:SignedProperties		署名対象のXAdESプロパティ。
23.1		Id	SignedPropertiesエレメント(23)のId。
24	XAdES:SignedSignatureProperties		
25	XAdES:SigningTime		署名した日時。
26	XAdES:SigningCertificate		署名者の公開鍵証明書の参照情報。
27	XAdES:Cert		
28	XAdES:CertDigest		公開鍵証明書のハッシュ情報。
29	ds:DigestMethod		公開鍵証明書のハッシュ計算に使用する、ハッシュアルゴリズムを指定する。
29.;1		Algorithm	ハッシュアルゴリズムのURI。
30	ds:DigestValue		公開鍵証明書のハッシュ値。
31	XAdES:IssuerSerial		公開鍵証明書への参照情報。
32	ds:X509SerialNumber		公開鍵証明書のシリアル番号。
33	ds:X509IssuerName		公開鍵証明書のDN。
34	XAdES:UnsignedProperties		非署名対象のXAdESプロパティ。
35	XAdES:UnsignedSignatureProperties		
36	XAdES:SignatureTimeStamp		署名値(15)へのタイムスタンプ。
37	XAdES:EncapsulatedTimeStamp		タイムスタンプトークン。
37.1		Encoding	タイムスタンプトークン（36)のエンコード方式を指定する（DER）。 "http://uri.etsi.org/01903/v1.2.2#DER"（固定値）

3.1.2 Enveloping型のXML署名への対応

XML Signature仕様にはXML署名の構造としてEnveloping型、Enveloping型、Detached型の3種類が定義されているが、今回の開発では、Enveloping型のXML署名のみを対象とする（『図2-1 XAdES-T XML署名の構成』を参照）。

3.1.3 入れ子構造の多重署名の作成

複数署名の作成に対応する。ここでいう複数署名とはXAdES使用に定義されたCounterSignatureエレメントを使用した直列署名ではなく、Enveloping型のXML署名で対象となるユーザドキュメントを複数回、入れ子構造で署名したものである（下図参照）。

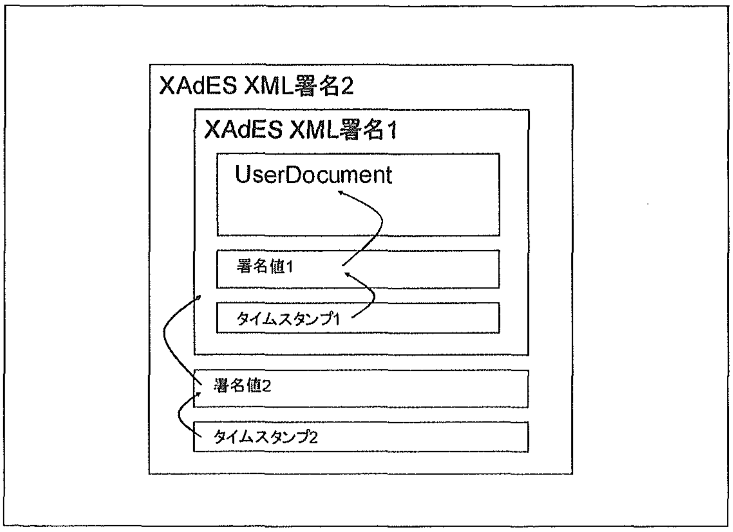


図 3-1 XAdES XML 署名と複数署名

3.1.4 RSA 及びDSA のサポート

電子署名アルゴリズムとして RSA 及び DSA をサポートする。（この機能は、XML 署名ライブラリ全体でサポートされる。）

3.1.5 SHA-1 及びSHA-256（またはそれ以上）のサポート

ハッシュアルゴリズムとして SHA-1、SHA-256、SHA-384、及び SHA-512 をサポートする。（この機能は、XML 署名ライブラリ全体でサポートされる。）

3.2 XAdES-T XML 署名の検証

3.2.1 XAdES-T 仕様に準拠した、タイムスタンプを含む XML 署名の検証

XAdES-T XML 署名を、以下の手順で検証する。[6]

3.2.1.1 署名フォーマットを検証する

XML スキーマを使用して XML 署名（Signature エレメント）が正しく記述されている事を確認する（ただし、署名対象のユーザドキュメントのフォーマットの検証は行わない）。

加えて次の通りに、XAdES-T 仕様に準拠して正しくプロパティやユーザドキュメントがセットされている事を確認する。[7]

3.2.1.1.1 SigningCertificate エレメント（26）が存在する事を確認する

3.2.1.1.2 SignedProperties エレメント（23）への参照を定義する Reference エレメント（10）の Type 属性（10.2）に "http://uri.etsi.org/01903#SignedProperties" がセットされている事を確認する。

3.2.1.2 署名者の公開鍵証明書の取得する

X509Certificate エレメント（17）から署名者の公開鍵証明書を取得する。

3.2.1.3 公開鍵証明書を検証する

『3.3 X.509 証明書の検証』を参照。

3.2.1.4 XML Signature の仕様に従い、XML 署名を公開鍵証明書で検証する[8]

この処理は XML Signature の仕様にしたがって行われる。そのため、外部の XML Security ライブラリに処理を任せることとし、ここでは処理の詳細は記述しない。

3.2.1.5 SigningTime エレメント（25）に記述された時間が、検証時より前である事を確認する

3.2.1.6 SignatureTimeStamp エレメント（36）からタイムスタンプトークンを取得する

3.2.1.7 タイムスタンプトークンを検証する

『3.4 タイムスタンプトークンの検証』を参照。

3.2.1.8 SignatureValue エレメント（18）から、署名値を取得する

3.2.1.9 タイムスタンプトークンに含まれるハッシュ値と、署名値のハッシュ値とが一致する事を確認する

この時、指定されたアルゴリズムで SignatureValue エレメントの正規化及びハッシュ地の取得を行う。

3.2.2 多重署名された XML 署名の検証

XML 署名の検証において、署名対象のユーザドキュメントが入れ子の XML 書名（Signature エレメント）であった場合、全ての署名及びタイムスタンプを検証するま

[6] 『ETSI TS 101 903 V\1.3.2』 p.89 G.2

[7] 『ETSI TS 101 903 V\1.3.2』 p.89 G.2.2.1

[8] 『XML-Signature Core Syntax and Processing』

で再帰的に検証を行う。

3.3 X.509 証明書の検証

3.3.1 HTTP プロトコルによるリポジトリからの公開鍵証明書及び CRL の取得

HTTP プロトコルによる外部サーバのリソースへのアクセスをサポートする。

3.3.2 階層型の証明書チェーン構築及び検証

次の手順で公開鍵証明書の証明書チェーンの構築及び検証を行う。

3.3.2.1 証明書チェーンの構築

信頼点の CA 証明書までの認証パスを構築する（Microsoft の認証パス構築アルゴリズムを参照[9]）。

3.3.2.1.1 公開鍵証明書の AKI（Authority Kei Identifier）プロパティの値をキーに信頼点キーストアを検索する

3.3.2.1.2 3.3.2.1.1で証明書が見つからない場合は Issuer を中間 CA とみなし、AIA（Authority Information Access）プロパティを参照して親証明書を取得する

AIA が指定する URL へのアクセスには HTTP プロトコルのみをサポートする。

またこの時、AIA の URI にマップ設定がある場合は、その設定に従いローカルファイルを参照する。

3.3.2.1.3 信頼点の証明書を取得し、証明書チェーンを構築するまで、3.3.2.1.1～3.3.2.1.3を繰り返す

3.3.2.2 CRL・ARL の取得

3.3.2.2.1 証明書チェーン上の全ての証明書の CRLDistributionPoints 属性から、CRL・ARL を取得する

CRLDistributionPoints が指定する URL へのアクセスには HTTP プロトコルのみをサポートする。

またこの時、CRLDistributionPoints の URI にマップ設定がある場合は、その設定に従いローカルファイルを参照する

3.3.2.3 証明書チェーンの検証

3.3.2.3.1 パス上の全ての証明書が正しい事を確認する（証明書を発行した CA の公開鍵で検証する）

3.3.2.3.2 パス上の全ての証明書について有効期限を過ぎていない事を確認する

3.3.2.3.3 パス上の全ての証明書について CRL・ARL を検査し、証明書が失効していないことを確認する

3.3.2.3.4 RFC 3280 に従いパス上の全ての証明書について証明書拡張のポリシの一致や制約条件を満たしていることを確認する[10]

この処理は標準的な X.509 証明書の検証アルゴリズムにしたがって行われる。そのため、外部の Security ライブラリに処理を任せることとし、ここでは処理の詳細は記述しない。

[9] http://www.microsoft.com/japan/technet/prodtechnol/windows2000serv/deploy/confeat/cryptpki.mspx

[10] RFC 3280 Chapter6

3.3.3 拡張領域を含むX.509証明書プロパティの取得

『保健医療福祉分野PKI認証局 証明書ポリシ』に定義された、（hcRoleといった）拡張領域を含むX.509証明書のプロパティへのアクセスをサポートする。

サポートするプロパティについては同仕様書の『表7.1.1 証明書のプロファイル（基本領域）』及び『表7.1.2 証明書のプロファイル（拡張領域Extensions）』を参照。

3.4 タイムスタンプトークンの検証

3.4.1 TSP over HTTP プロトコルによる TSA サーバからのタイムスタンプトークン取得

HTTP プロトコルを利用した TSA サーバとの TSP Request 及び TSP Response セッションをサポートする。

3.4.2 タイムスタンプトークンの検証

次の手順でタイムスタンプトークンの検証を行う。

3.4.2.1 タイムスタンプトークンのフォーマットの検証

タイムスタンプトークンを取り出し、そのフォーマットが正しい事を確認する。

3.4.2.2 タイムスタンプトークンから署名者（TSA）の証明書を取り出す。

3.4.2.3 タイムスタンプトークンの有効性検証

TSA の公開鍵を使用して、タイムスタンプトークンの署名値を検証する。

3.4.2.4 TSA 証明書の検証

『3.3 X.509 証明書の検証』を参照。ただし、TSA 証明書の非検証オプションがある場合はこのステップを省略する。

3.4.3 タイムスタンプトークンのプロパティの取得

タイムスタンプトークンのプロパティへのアクセスをサポートする。サポートするプロパティについては下記の表を参照（作成中）。

4. XML 署名のサンプル

本章では XML 署名ライブラリが作成・検証の対象とするサンプル XML を示す。

4.1 XML 署名のサンプル

次の『表 4-1 XAdES XML 署名のサンプル』は、XAdES 準拠の XML 署名のサンプルである。左列の番号は、前節の『表 3-1 XAdES XML 署名の構造』を参照。

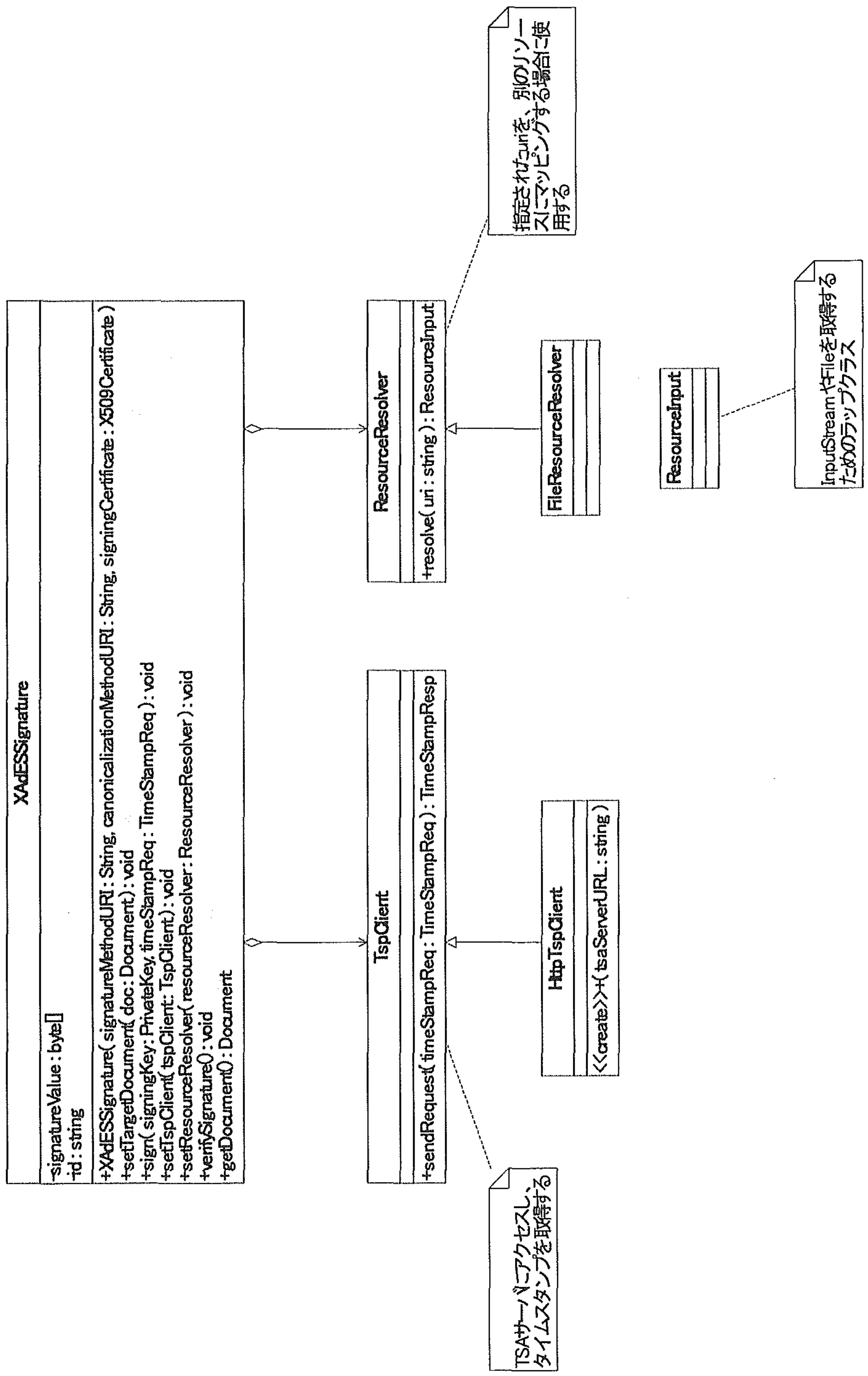
表 4-1 XAdES XML 署名のサンプル

```
     <?xml version="1.0" encoding="UTF-8"?>
1    <ds:Signature xmlns:XAdES="http://uri.etsi.org/01903/v1.3.2#"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       Id="Id_Signature">
2      <ds:SignedInfo>
3        <ds:CanonicalizationMethod
           Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
4        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
5        <ds:Reference URI="#Id_Object">
6          <ds:Transforms>
7            <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
           </ds:Transforms>
8          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
9          <ds:DigestValue>8IEa14qyWsA/tS4FW5T1zvINX7Y=</ds:DigestValue>
         </ds:Reference>
10       <ds:Reference Type="http://uri.etsi.org/01903#SignedProperties" URI="#Id_SignedProperties">
11         <ds:Transforms>
12           <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
           </ds:Transforms>
13         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
14         <ds:DigestValue>pJcypcjrxfuxJ5//+m/xDxtyFJc=</ds:DigestValue>
         </ds:Reference>
       </ds:SignedInfo>
15     <ds:SignatureValue>T/wAXnzjC+aqAT4IcTPcS60+wFtsU0wpZvYe5HhR5uJSRQc61XzumQ==</ds:SignatureValue>
16     <ds:KeyInfo>
17       <ds:X509Data>
18         <ds:X509Certificate>P/IpDHIgpjGCejHeJauae53U/y1j3WecI//WtJ2Nx3BXd5NRHw4Odg==</ds:X509Certificate>
         </ds:X509Data>
       </ds:KeyInfo>
19     <ds:Object Id="Id_Object">
20       <UserDocument />
       </Object>
21     <ds:Object>
22       <XAdES:QualifyingProperties Target="#Id_Signature">
23         <XAdES:SignedProperties Id="Id_SignedProperties">
24           <XAdES:SignedSignatureProperties>
25             <XAdES:SigningTime>2006-04-01T12:00:00.00000+09:00</XAdES:SigningTime>
26             <XAdES:SigningCertificate>
27               <XAdES:Cert>
28                 <XAdES:CertDigest>
29                   <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
30                   <ds:DigestValue>f+pDsT3LzyKV9Sg6rdK5bBrQIbo=</ds:DigestValue>
```

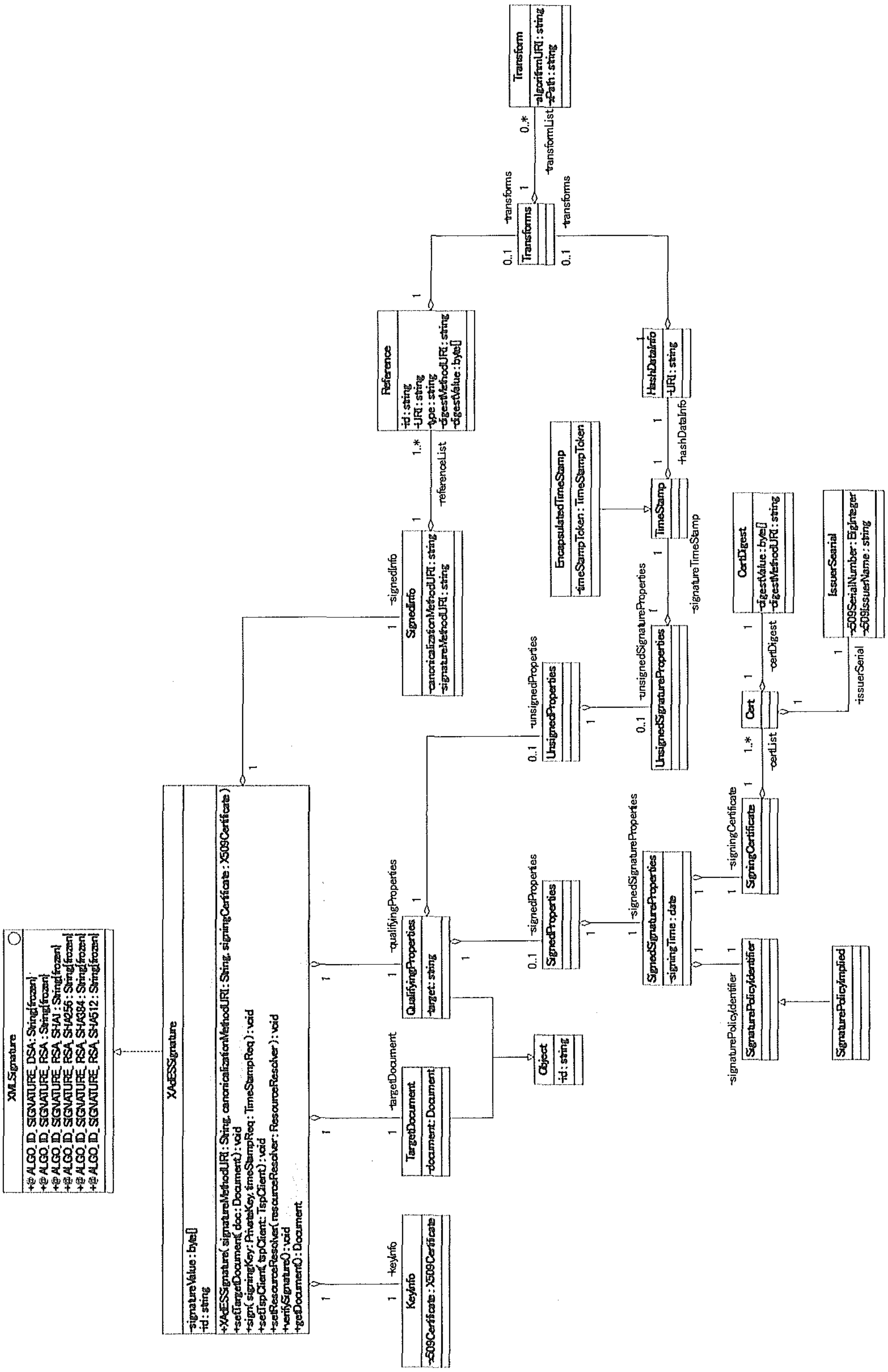
```
                  </XAdES:CertDigest>
31                <XAdES:IssuerSerial>
32                  <ds:X509IssuerName>CN=Human Welfare, Inc.</ds:X509IssuerName>
33                  <ds:X509SerialNumber>1906523</ds:X509SerialNumber>
                  </XAdES:IssuerSerial>
                </XAdES:Cert>
              </XAdES:SigningCertificate>
            </XAdES:SignedSignatureProperties>
          </XAdES:SignedProperties>
34        <XAdES:UnsignedProperties>
35          <XAdES:UnsignedSignatureProperties>
36            <XAdES:SignatureTimeStamp>
37              <XAdES:EncapsulatedTimeStamp Encoding="http://uri.etsi.org/01903/v1.2.2#DER">
                  T/wAXnzjC+aqAT4lcTPcS60+wFtsU0wpZvYe5HhR5uJSRQc61XzumQ==
                </XAdES:EncapsulatedTimeStamp>
              </XAdES:SignatureTimeStamp>
            </XAdES:UnsignedSignatureProperties>
          </XAdES:UnsignedProperties>
        </XAdES:QualifyingProperties>
      </ds:Object>
    </ds:Signature>
```

5. クラス図

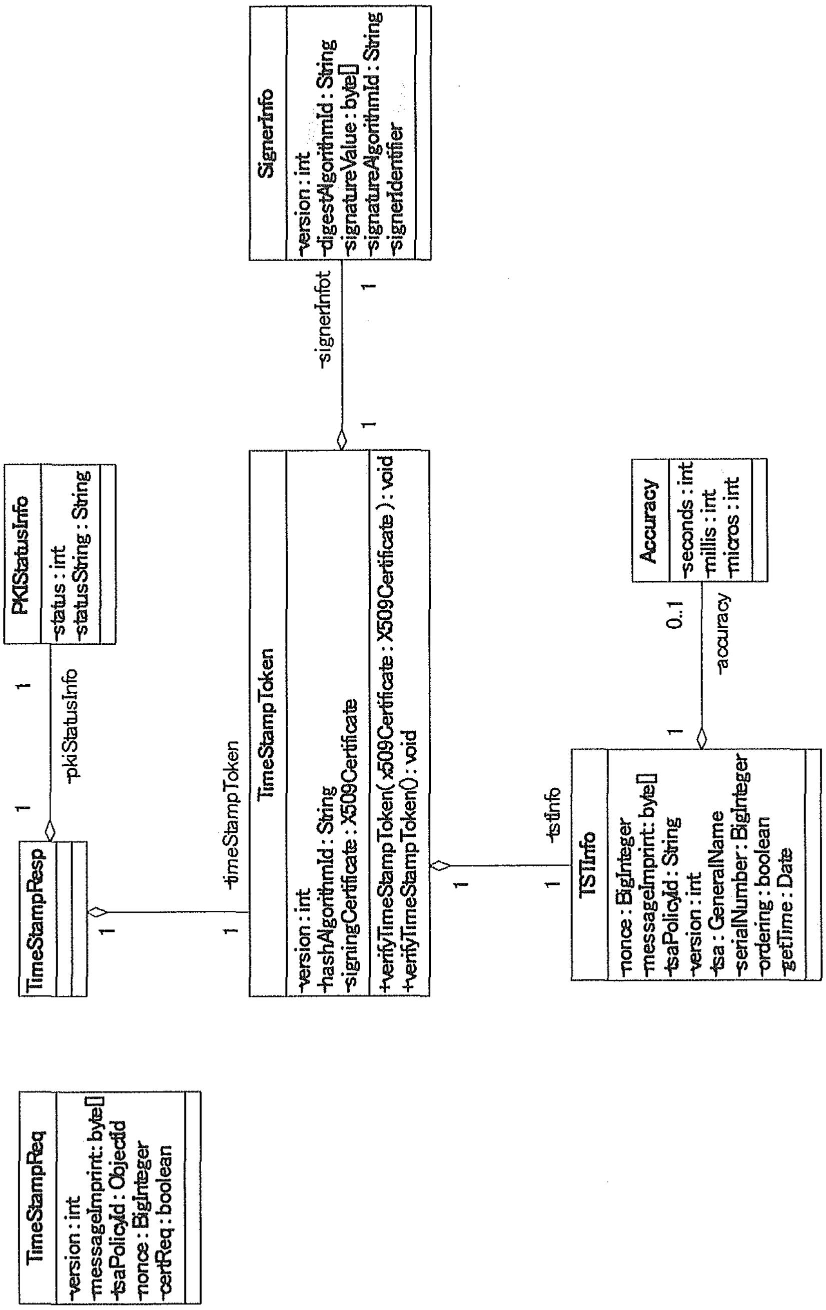
5.1 コントローラ


XAdESSignature
-signatureValue : byte[]
-id : string
+XAdESSignature(signatureMethodURI : String, canonicalizationMethodURI : String, signingCertificate : X509Certificate)
+setTargetDocument(doc : Document) : void
+sign(signingKey : PrivateKey, timeStampReq : TimeStampReq) : void
+setTspClient(tspClient : TspClient) : void
+setResourceResolver(resourceResolver : ResourceResolver) : void
+verifySignature() : void
+getDocument() : Document
ResourceResolver
+resolve(uri : string) : ResourceInput
指定されたuriを、別のリソースにマッピングする場合に使用する
FileResourceResolver
ResourceInput
InputStreamやFileを取得するためのラップクラス
TspClient
+sendRequest(timeStampReq : TimeStampReq) : TimeStampResp
HttpTspClient
<<create>>+(tsaServerURL : string)
TSAサーバにアクセスし、タイムスタンプを取得する

5.2 XAdESSignature（データモデル）


XMLSignature
+ALGO_ID_SIGNATURE_DSA : String{frozen}
+ALGO_ID_SIGNATURE_RSA : String{frozen}
+ALGO_ID_SIGNATURE_RSA_SHA1 : String{frozen}
+ALGO_ID_SIGNATURE_RSA_SHA256 : String{frozen}
+ALGO_ID_SIGNATURE_RSA_SHA384 : String{frozen}
+ALGO_ID_SIGNATURE_RSA_SHA512 : String{frozen}
XAdESSignature
-signatureValue : byte[]
-id : string
+XAdESSignature(signatureMethodURI : String, canonicalizationMethodURI : String, signingCertificate : X509Certificate)
+setTargetDocument(doc : Document) : void
+sign(signingKey : PrivateKey, timeStampReq : TimeStampReq) : void
+setTspClient(tspClient : TspClient) : void
+setResourceResolver(resourceResolver : ResourceResolver) : void
+verifySignature() : void
+getDocument() : Document
-signedInfo
SignedInfo
-canonicalizationMethodURI : string
-signatureMethodURI : string
-referenceList
1..*
Reference
-id : string
-URI : string
-type : string
-digestMethodURI : string
-digestValue : byte[]
-transforms
0..1
Transforms
-transformList
0..*
Transform
-algorithmURI : string
-xPath : string
-transforms
0..1
HashDataInfo
-URI : string
-hashDataInfo
TimeStamp
EncapsulatedTimeStamp
-timeStampToken : TimeStampToken
-signatureTimeStamp
-unsignedSignatureProperties
0..1
UnsignedSignatureProperties
-unsignedProperties
0..1
UnsignedProperties
-qualifyingProperties
QualifyingProperties
-target : string
-signedProperties
0..1
SignedProperties
-signedSignatureProperties
SignedSignatureProperties
-signingTime : date
-signingCertificate
SigningCertificate
-certList
1..*
Cert
-certDigest
CertDigest
-digestValue : byte[]
-digestMethodURI : string
-issuerSerial
IssuerSerial
-x509SerialNumber : BigInteger
-x509IssuerName : string
-signaturePolicyIdentifier
SignaturePolicyIdentifier
SignaturePolicyImplied
-targetDocument
TargetDocument
-document : Document
Object
-id : string
-keyInfo
KeyInfo
-x509Certificate : X509Certificate

5.3 TimeStamp（データモデル）



6. シーケンス図

Manufacturer Disclosure Statement for Medical Device Security – MDS^2

Device Category †	Manufacturer †	Document ID	Document Release Date
Device Model	Software Revision	Software Release Date	

Manufacturer or Representative Contact Information:	Name	Title	Department
	Company Name	Telephone #	e-mail

MANAGEMENT OF ELECTRONIC PROTECTED HEALTH INFORMATION (ePHI) *As defined by HIPAA Security Rule, 45 CFR Part 164)* Yes No N/A Note #

1. Can this device transmit or maintain *electronic Protected Health Information* (ePHI)? † .... _____ _____
2. Types of ePHI data elements that can be maintained by the device:
   a. Demographic (e.g., name, address, location, unique identification number)? .... _____ _____
   b. Medical record (e.g., medical record #, account #, test or treatment date, device identification number)? .... _____ _____
   c. Diagnostic/therapeutic (e.g., photo/radiograph, test results, or physiologic data with identifying characteristics)?. _____ _____
   d. Open, unstructured text entered by device user/operator? .... _____ _____
3. Maintaining ePHI: *Can the device*
   a. Maintain ePHI temporarily in volatile memory (i.e., until cleared on by power-off or reset)?............ _____ _____
   b. Store ePHI persistently on local media?............ _____ _____
   c. Import/export ePHI with other systems?............ _____ _____
4. Mechanisms used for the transmitting, importing/exporting of ePHI: *Can the device*
   a. Display ePHI (e.g., video display)? .... _____ _____
   b. Generate hardcopy reports or images containing ePHI? .... _____ _____
   c. Retrieve ePHI from or record ePHI to removable media (e.g., disk, DVD, CD-ROM, tape, CF/SD card, memory stick)?. _____ _____
   d. Transmit/receive or import/export ePHI via dedicated cable connection (e.g., IEEE 1073, serial port, USB, FireWire*)*? ... _____ _____
   e. Transmit/receive ePHI via a network connection (e.g., LAN, WAN, VPN, intranet, Internet)?............ _____ _____
   f. Transmit/receive ePHI via an integrated wireless connection (e.g., WiFi, Bluetooth, infrared)?† .... _____ _____
   g. Other ____ ?............ _____ _____

ADMINISTRATIVE SAFEGUARDS Yes No N/A Note #

5. Does manufacturer offer operator and technical support training or documentation on device security features?............ _____ _____
6. What underlying operating system(s) (including version number) are used by the device? ____________________ _____

PHYSICAL SAFEGUARDS Yes No N/A Note #

7. Are all device components maintaining ePHI (other than removable media) physically secure (i.e., cannot remove without tools)? _____ _____
8. Does the device have an integral data backup capability (i.e., backup onto removable media such as tape, disk)? .... _____ _____
9. Can the device boot from uncontrolled or removable media (i.e., a source other than an internal drive or memory component)? _____ _____

TECHNICAL SAFEGUARDS Yes No N/A Note #

10. Can software or hardware not authorized by the device manufacturer be installed on the device?............ _____ _____
11. Can the device be serviced remotely (i.e., maintenance activities performed by service person via network or remote connection)?. _____ _____
    a. Can the device restrict remote access to specific devices or network locations (e.g., specific IP addresses)? .... _____ _____
    b. Can the device log provide an audit trail of remote-service activity? .... _____ _____
    c. Can security patches or other software be installed remotely?............ _____ _____
12. Level of owner/operator service access to device operating system: *Can the device owner/operator*
    a. Apply device manufacturer-validated security patches?............ _____ _____
    b. Install or update antivirus software?............ _____ _____
    c. Update virus definitions on manufacturer-installed antivirus software?............ _____ _____
    d. Obtain administrative privileges (e.g., access operating system or application via local root or admin account)? .. _____ _____
13. Does the device support user/operator specific ID *and* password? .... _____ _____
14. Are access sessions terminated after a predetermined length of inactivity (e.g., auto logoff)? .... _____ _____
15. Events recorded in device audit log (e.g., user, date/time, action taken): *Can the audit log record*
    a. Login and logout by users/operators? .... _____ _____
    b. Viewing of ePHI? .... _____ _____
    c. Creation, modification or deletion of ePHI? .... _____ _____
    d. Import/export or transmittal/receipt of ePHI? .... _____ _____
16. Does the device incorporate an emergency access ("break-glass") feature that logs each instance of use? .... _____ _____
17. Can the device maintain ePHI (e.g., by internal battery) during power service interruptions? .... _____ _____
18. Controls when exchanging ePHI with other devices:
    a. Transmitted only via a physically secure connection (e.g., dedicated cable)? .... _____ _____
    b. Encrypted prior to transmission via a network or removable media? .... _____ _____
    c. Restricted to a fixed list of network addresses (i.e., host-based access control list)? .... _____ _____
19. Does the device ensure the integrity of the ePHI data with implicit or explicit error detection/correction technology? .... _____ _____

† Recommend use of ECRI's Universal Medical Device Nomenclature System (UMDNS).

Adapted from *Information Security for Biomedical Technology: A HIPAA Compliance Guide*, ACCE/ECRI, 2004.
ACCE – the American College of Clinical Engineering; ECRI – formerly the Emergency Care Research Institute.

Manufacturer Disclosure Statement for Medical Device Security – MDS^2

RECOMMENDED SECURITY PRACTICES

EXPLANATORY NOTES *(from questions 1 – 19):*
IMPORTANT: Refer to Instructions for the Manufacturers Disclosure Statement for Medical Device Security for the proper interpretation of information provided in this form.

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.

Adapted from *Information Security for Biomedical Technology: A HIPAA Compliance Guide*, ACCE/ECRI, 2004.
ACCE – the American College of Clinical Engineering; ECRI – formerly the Emergency Care Research Institute.

Instructions for the Manufacturer Disclosure Statement for Medical Device Security – MDS^2 Version 1.0

Introduction

In light of increased focus on medical device security and the upcoming April 21, 2005 deadline for compliance with the HIPAA Security Rule, the HIMSS Medical Device Security Workgroup has created a standard Manufacturer Disclosure Statement for Medical Device Security (MDS^2). The intent of the MDS^2 is to supply healthcare providers with important information that can assist them in assessing the vulnerability and risks associated with electronic Protected Health Information (ePHI)[1] transmitted or maintained by medical devices. Because security risk assessment is a broad, organization-wide effort, this document focuses on only those elements of the risk assessment process associated with medical devices and systems that maintain or transmit ePHI. A standardized form allows manufacturers to quickly respond to a potentially large volume of requests from providers for information regarding the security-related features of the medical devices they manufacture. The standardized form also facilitates the providers' review of the large volume of security-related information supplied by the manufacturers. This form was adapted from portions of the *ACCE/ECRI Biomedical Equipment Survey Form,* a key tool found in *Information Security for Biomedical Technology: A HIPAA Compliance Guide* (ACCE/ECRI, 2004). HIMSS recommends that the information in the MDS^2 be used to help complete the ACCE/ECRI form and associated processes as part of each organization's HIPAA Security compliance efforts.

The manufacturer-completed MDS^2 should:

(1) Be useful to healthcare provider organizations worldwide.
While the form does supply information important to providers who must comply with the HIPAA Security Rule, the information presented is intended to be useful for *any* healthcare provider who aspires to have an effective information security and risk management program. Outside the US, providers would therefore find the MDS^2 an effective tool in addressing such regional regulations as EC 95/46, HPB 517, and PIPEDA.[2]

(2) Include device-specific information addressing the technical security-related attributes of the individual device model.

This completed MDS^2 form provides a simple, flexible way of collecting the technical, device-specific elements of the total information needed by provider organizations (device users/operators) in preparing for their first round of medical device risk assessments. Providers around the world should find a completed MDS^2 form useful in controlling information security (i.e., confidentiality, integrity, and availability) risks. Note, however, that the MDS^2 is not intended and should not be used as a basis for medical device procurement. Writing procurement specifications requires a deeper and more extensive knowledge of security and the provider's mission.

Using the information provided by the manufacturer in the MDS^2 combined with information collected about the care delivery environment (e.g., through tools like ACCE / ECRI's guide for *Information Security for Biomedical Technology*), the provider's multidisciplinary risk assessment team can review assembled information and make informed decisions on implementing a local security management plan.

The Role of Healthcare Providers and Medical Device Manufacturers in the Security Management Process

Responsibility for effective security management must ultimately lie with the provider organization. Generally the device manufacturers can assist providers in their security management programs by offering information associated with

- the *type* of data maintained / transmitted by the manufacturer's device or system
- *how* data is maintained / transmitted by the manufacturer's device or system
- any *security–related features* incorporated in the manufacturer's device or system

[1] As defined by HIPAA Security Rule, 45 CFR Part 164.

[2] EC 95/46 is the European Parliament and Council's Directive 95/46/EC on the *Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data*,
HPB 517 is the Japanese *Electronic Storage of Clinical Records* law ; and
PIPEDA is the Canadian *Personal Information Protection and Electronic Documents Act.*

In order to effectively manage medical information security and comply with relevant regulations, healthcare providers must employ administrative, physical and technical safeguards, most of which (other than some technical safeguards) must be adopted and employed on-site extrinsic to the actual device. Other than some general recommendations with regard to medical devices:

- there are few ADMINISTRATIVE safeguards manufacturers can address beyond providing assistance in security training
- there are few PHYSICAL safeguards manufacturers can address beyond incorporating physical security features (e.g., component lock & key, theft/intrusion alarms) in their devices

The greatest impact manufacturers can have on medical device security is to incorporate TECHNICAL Safeguards (i.e., security features) in their devices to facilitate the providers efforts in maintaining an effective security program and meeting any relevant regulations. The medical device manufacturing industry is increasingly aware of the importance of having effective security features in their devices and systems. Manufacturers are generally including such features in the production of new devices and systems based provider needs and requirements.

Instructions for Obtaining and Using the MDS^2

Information provided on the MDS^2 is intended to assist professionals knowledgeable in security and risk assessment processes in their management of medical device security issues. The information on the MDS^2 is not intended and may be inappropriate for any other purpose.

Completed MDS^2 forms for many devices and systems may be available directly from the device manufacturer. Check the manufacturer's web site first for relevant forms and, when not available there, contact a manufacturer's representative to request a MDS^2 for the appropriate device(s)/system(s). If a manufacturer does not have a completed MDS^2 for the appropriate device(s)/system(s), enter the device category, manufacturer and model information in the appropriate boxes on the top of a blank form[3] and submit the form(s) and these instructions to the manufacturer's compliance office for their completion.

Note that HIMSS suggests that a standard naming convention be used for the device category terms and manufacturer names listed on the form. This assists providers in matching information from the form to their equipment inventories. ECRI's Universal Medical Device Nomenclature System (UMDNS) is the most widely used. Adopted by thousands of healthcare providers worldwide, UMDNS has been adopted by the National Library of Medicine into its Universal Medical Language System, and has been recommended by the Institute of Medicine for inclusion in the US Department of Health and Human Services (HHS) National Committee on Vital and Health Statistics (NCVHS) core terminology group. For more information about UMDNS contact ECRI at www.ecri.org.

Side 1 of the MDS^2 contains descriptive information on the *type* of data maintained/transmitted by device, *how* the data is maintained/transmitted, and any *security–related features* incorporated in the device. Side 2 contains manufacturer-optional recommended security practices and space for numbered explanatory notes that may expand on answers to questions 1 through 19. Manufacturers may elect to attach supplementary material if additional space for recommended practices or explanatory notes is necessary.

Question-specific notes.

1. *Maintaining* ePHI includes storage on an internal disk, removable media, short-term computer memory, etc. *Transmitting* ePHI includes *receiving/sending* external to the device via network, telephone, direct connect cable, removable media, etc.

[3] Additional form copies and instructions may be downloaded from http://www.himss.org/asp/medicalDeviceSecurity.asp

2. The four sections of this question relate directly to the 18 data elements referred to in HIPAA, any of which, if present, render the entire electronically transmitted/maintained data set as ePHI. The 18 data elements included in the Rule are:

    - *Name*
    - *Geographic data (e.g., address)*
    - *Dates (e.g., date of birth, admission, discharge, death, treatment)*
    - *Telephone No.*
    - *Fax No.*
    - *E-mail address*
    - *Social Security No.*
    - *Medical record No.*
    - *Health plan beneficiary No.*
    - *Account No.*
    - *Certificate/license numbers*
    - *Vehicle identifiers*
    - *Device identifiers*
    - *Universal Resource Locators (URLs)*
    - *IP address numbers*
    - *Biometric identifiers*
    - *Full face (or comparable) photographic images*
    - *Any unique identifying number, characteristic, or code*

    The *open, unstructured text* question (2d) is intended to indicate an additional element where a provider might put further identifying information.

3. Here the manufacturer provides more detail on how ePHI is maintained. Note that a fully networked device is likely to have all three items with a 'Yes' response.
    - Persistently on local media refers to media created or directly attached to the device under consideration (e.g., MR Scanner, 3-D workstation), not a remote archive.
    - Import/export ePHI refers to data that is sourced or destined to remote storage devices (e.g., an MR scanner that relies on an image server in a PACS system).

4. *Import and export* refer to movement of information via published open protocols to devices outside of the medical device under consideration (e.g., medical information bus (IEEE 1073), serial port, and published protocol that allows general access to ePHI). *Dedicated cable* here refers to communication via a point to point cable to a device or system outside of the device under consideration.

5. This question includes either explicit security training or explicit sections of administrator or user manuals that detail the device security features and their use.

6. This question identifies the underlying 3rd party system software platform (operating system) name or indicates if there is no 3rd party platform (i.e., proprietary system created for this manufacturer alone).

7. Refers to the typical installation of the manufacturer's device.

8. Refers to an integrated feature that supports information backup onto removable media (e.g., optical disk, magnetic disk, tape).

9. Identifies whether it is possible to start the device with software from any source other than the manufacturer's normal startup device (e.g., an integral hard disk or ROM).

10. Does the device allow, through root access, administrative privilege, or other non-intrusive method, a local user and/or IT staff to install software not provided and not explicitly authorized by the manufacturer (e.g., email client, office applications, virus scanner, browsers, games)?

11. Remote service refers to device maintenance activities performed by a service person via network or other remote connection.

12. Level of owner/operator access to device operating system. Here the manufacturer details what is technically possible if the device owner (generally the healthcare provider) has the technical ability to install security controls on the medical device under consideration. A MANUFACTURER ANSWERING 'YES' TO ANY OF THESE QUESTIONS DOES NOT MEAN THAT THE MANUFACTURER AUTHORIZES THE OWNER TO PERFORM THESE FUNCTIONS. THE OWNER ASSUMES ALL RESPONSIBILITY FOR UNAUTHORIZED INSTALLATION or REPAIRS. UNAUTHORIZED INSTALLATION or REPAIRS MAY VOID APPLICABLE WARRANTIES AND SERVICE AGREEMENTS. Authorization to perform these security-related services or changes to a medical device should be obtained in writing from the device manufacturer. Unauthorized changes to a medical device may remove it from government regulatory controls (e.g., FDA) and render the device an experimental medical device.

13. Self-explanatory.

14. Self-explanatory.

15. Clarifications:
    - Controlled viewing refers to operations that have to do with the display, printing, or other use of ePHI (e.g., image display, record print-out).
    - Creation, modification, or deletion would mean that all these events are tracked in the log file.
    - Export or transmittal refers to the movement of ePHI outside of the device under consideration.
16. *Emergency access* features allow operators emergency access to the device in cases where the normal authentication cannot be successfully completed or is not working properly.
17. Self-explanatory.
18. Clarifications:
    - *Physically secure connection* is a cabling system that is not accessible to the general public. (i.e., it is in a physically controlled space such as examining rooms, communication closets, or building plenum).
    - *Fixed list* is an explicit mechanism that limits the connections and nature of connections on a per-device basis.
19. *Ensure integrity* refers to methods that can detect and/or correct differences between the source makeup of an ePHI message and the ePHI message received by an external device. Is such a method available for use as part of the device under consideration (either in transmission or receipt of ePHI)?

Disclaimer

This document is intended to assist healthcare providers in meeting their regulatory obligations regarding medical device security. It is the obligation of the users of this document (e.g., the healthcare provider) to employ all necessary and appropriate safeguards to meet their regulatory and organizational requirements. HIMSS does not assume any responsibility for the application or the content of this form.