



## **Política de Sellado de Tiempo (Timestamping, TSA)**

	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>		Edición	Página
			1	2

<b>Título del documento:</b>	Política de sellado de tiempo (Timestamping, TSA)
<b>Nombre del fichero:</b>	ACEDICOM - Política Servicio Timestamping.odt
<b>Versión:</b>	V1.1
<b>Estado:</b>	VIGENTE
<b>Fecha:</b>	27/01/11
<b>Autor:</b>	JOSE VILATA

Historial de cambios			
Versión	Fecha	Descripción de la acción	Páginas
V1.0	25/08/08	Documento inicial	
V1.1	27/01/11	Se añaden ejemplos en JAVA de conexión al servicio	Apartado 7

## Tabla de Contenidos

1 INTRODUCCIÓN.....	4
2 REFERENCIAS.....	5
3 DEFINICIONES Y ABREVIATURAS.....	6
3.1. DEFINICIONES .....	6
3.2. ABREVIATURAS .....	6
4 CONCEPTOS GENERALES.....	7
4.1 SERVICIO DE SELLADO DE TIEMPO.....	7
4.2. AUTORIDAD DE SELLADO DE TIEMPO (TSA) .....	8
4.3. SUBSCRIPTORES .....	8
5. POLÍTICA DE SELLADO DE TIEMPO .....	9
5.1. VISTA GENERAL .....	9
5.2. IDENTIFICACIÓN DE LA POLÍTICA DE SELLADO DE TIEMPO .....	9
5.3. APLICACIÓN DEL SELLADO DE TIEMPO .....	10
6 OBLIGACIONES Y RESPONSABILIDADES.....	11
6.1 OBLIGACIONES DE LA TSA.....	11
6.1.1 Obligaciones.....	11
6.1.2 Responsabilidad financiera.....	11
6.1.3 Exoneración de responsabilidad.....	12
6.1.4 Cese de la actividad de la TSA.....	12
6.2 CLIENTE.....	12
6.3 TERCERO QUE CONFÍA EN LOS SELLOS DE TIEMPO.....	13
7 REQUERIMIENTOS OPERACIONALES.....	14
7.1 OBTENCIÓN DEL TIEMPO FIABLE.....	14
7.2 CERTIFICADO DE TSA.....	14
7.2.1. Generación de claves de la TSA .....	14
7.2.2. Protección de la clave privada de la TSA .....	15
7.2.3 Publicación del certificado de TSA.....	15
7.2.3 Cambio de certificado de TSA.....	15
7.3 IMPLEMENTACIÓN DE LA SOLICITUD Y RESPUESTA DE SELLOS DE TIEMPO.....	15
7.3.1. Protocolo <i>Timestamp</i> vía <i>HTTP</i> .....	15
7.4 FORMATO DE LOS MENSAJES.....	16
7.4.1 <i>Timestamp Request</i> .....	16
7.4.2 <i>Timestamp Response</i> .....	17
7.4.3 <i>Validate Request</i> .....	19
7.4.4 <i>Validate Reply</i> .....	19
7.5 EJEMPLO DE CONEXIÓN EN JAVA.....	20

	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>	
	Edición 1	Página 4

## 1 Introducción

---

EDICOM, como Prestador de Servicios de Certificación que emite certificados reconocidos según la Ley 59/2003 de 19 de diciembre de firma electrónica, también ofrece servicios de Sellado de Tiempo.

Esta Política establece las reglas generales empleadas por la Autoridad de Sellado de Tiempo de la Autoridad de Certificación de EDICOM (en adelante ACEDICOM), para la emisión de tokens que contienen sellos de tiempo firmados. Se establecen en este documento los participantes de estos procesos, especificando sus responsabilidades, derechos y ámbito de aplicación.

La Ley 59/2003 de Firma Electrónica no recoge ni regula la emisión de sellos de tiempos. Sin embargo, es intención de EDICOM dotar a los sellos de tiempo emitidos la condición de “Sellos de Tiempo reconocidos” equivalente a la condición de “Firmas electrónicas reconocidas”, en la medida que esto sea posible y comprometiéndose a cumplir con la legislación aplicable en cada caso.

La presente política es conforme a la norma del ETSI TS 102 023 v1.2.1 “Policy requirements for time-stamping authorities” y a su especificación equivalente RFC-3268 “Requirements for time-stamping authorities”.

Esta Política asume cierto grado de conocimiento por parte del lector de conceptos relacionados con las infraestructuras de clave pública y los sellos de tiempo. Si este no fuera el caso, se recomienda al lector que se informe sobre los temas anteriores antes de continuar con la lectura del presente documento.

El presente documento puede ser usado por las partes confiantes y los subscriptores de los servicios proporcionados por la ACEDICOM como base para garantizar la confianza de los servicios que se describen en este documento.

Esta política esta basada en criptografía de clave pública, fuentes de tiempo fiables y certificados X.509 v3 y está subordinada al cumplimiento de las Condiciones Generales expuestas en la **Declaración de Prácticas de Certificación (CPS)** de la ACEDICOM.

	POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)		
		Edición 1	Página 5

## 2 Referencias

---

Los documentos que se citan a continuación se mencionan a lo largo del texto:

- [1] Declaración de Prácticas de Certificación de la ACEDICOM (CSP)
- [2] ETSI TS 102 023 “Policy Requirements for time-stamping authorities”
- [3] RFC-3161 “Internet X.509 Public Key Infrastructure – Time Stamp Protocol (TSP)”
- [4] ETSI TS 101 861 “Time Stamping Profile”
- [5] ETSI SR 002 176 “Algorithms and Parameters for Secure Electronic Signatures”

	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>	
	Edición 1	Página 6

## 3 Definiciones y abreviaturas

---

### 3.1. Definiciones

Para los propósitos del presente documento, se aplican los siguientes términos y definiciones:

- **Autoridad de Sellado de Tiempo:** Sistema de emisión y gestión de sellos de tiempo seguros
- **Subscriber:** Persona o entidad que solicita los servicios proporcionados por la Autoridad de Sellado de Tiempo.
- **Token de sello de tiempo:** Dispositivo de datos empleado en un proceso de creación de firma electrónica, que une la representación de un dato a un tiempo concreto, estableciendo así evidencia de que el dato existía antes de ese tiempo.
- **Usuario:** Destinatario de un Token de sello de tiempo y que confía en el mismo.
- **Declaración de Prácticas de sellado de tiempo:** Declaración de las Prácticas que una Autoridad de sellado de tiempo emplea en la emisión. En el caso de la ACEDICOM, todos los puntos que debe tratar esta declaración se encuentra integrada con los documentos operacionales, de procedimiento y técnicos que engloban toda la plataforma de la misma.

### 3.2. Abreviaturas

TSA: Autoridad de Sellado de Tiempo  
 TSS: Servicio de sellado de tiempo  
 TSQ: Solicitud de sello de tiempo  
 ACEDICOM: Autoridad de Certificación de EDICOM  
 TST: Token de sello de tiempo  
 IETF: Internet Engineering Task Force  
 CEN: Comité Europeo de Normalización  
 CWA: Cen Workshop Agreement  
 RFC: Request for comment  
 UTC: Universal Time Coordinated  
 CRL: Certificate Revocation List  
 FIPS: Federal Information Processing Standards  
 HSM: Hardware Security Module  
 GPS: Global Positioning System

## 4 CONCEPTOS GENERALES

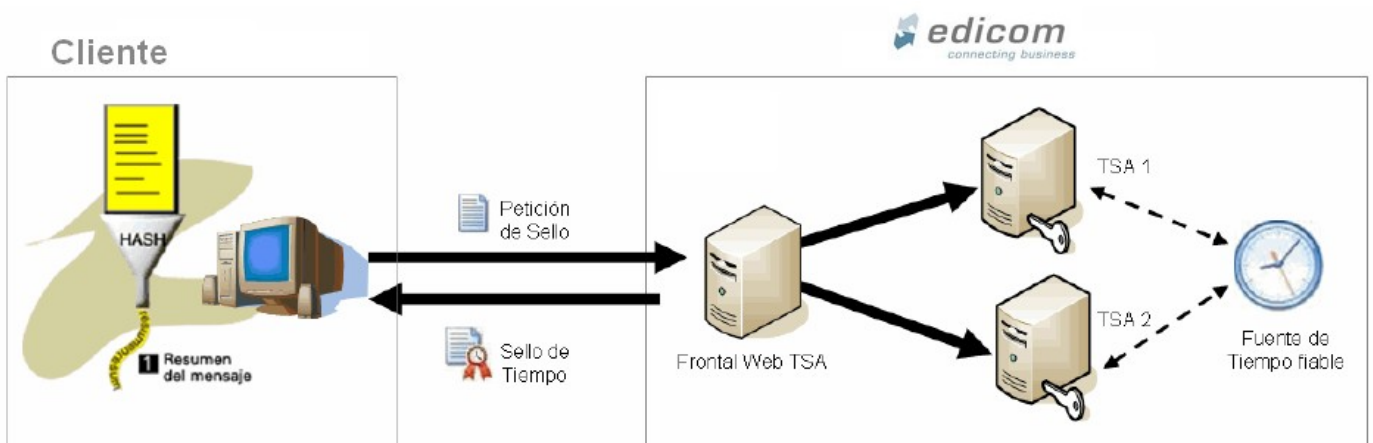
### 4.1 servicio de sellado de tiempo

El sellado de tiempo (Timestamping) es un mecanismo on-line que permite demostrar que una serie de datos han existido y no han sido alterados desde un instante específico en el tiempo.

La implementación de la política de sellado de tiempo se debe cumplir con el protocolo definido en la norma **RFC 3161 "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)"**.

Los pasos para generar un sello de tiempo son los siguientes:

- El cliente calcula el hash del documento a sellar
- El cliente envía una solicitud de sello de tiempo a una URL determinada de ACEDICOM siguiendo el protocolo RFC 3161, incluyendo el hash del documento a sellar
- ACEDICOM recibe la petición, revisa si la petición está completa y correcta y realiza un control de acceso en función del usuario y password del cliente.
- Si el resultado es correcto, la TSA firma la petición generando un Sello de Tiempo (incluyendo el hash del documento, la fecha y hora obtenida de una fuente fiable y la firma electrónica de la TSA).
- El sello de tiempo se envía de vuelta al Cliente
- El Cliente debe validar la firma del sello y custodiarlo debidamente
- La TSA de la ACEDICOM también mantendrá un registro de los sellos emitidos para su futura verificación si así se ha contratado con el cliente



	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>	
	Edición 1	Página 8

## 4.2. Autoridad de Sellado de Tiempo (TSA)

La autoridad en la que confían los usuarios de los servicios de sellado de tiempo (suscriptores y partes confiantes) para la emisión de los sellos de tiempo. La TSA tiene responsabilidad global en la provisión del servicio de sellado de tiempo que se identifica en la cláusula 4.1.

## 4.3. Suscriptores

Los suscriptores de este servicio son los usuarios del sistema con los que se haya suscrito el correspondiente convenio de prestación de servicios de sellado de tiempo electrónico.

Los clientes envían peticiones de sellado y reciben sellos de tiempo siguiendo el protocolo RFC3161 Time Stamp Protocol (TSP).

Los clientes deben adaptar sus sistemas para poder realizar peticiones de sellado de tiempo. Existen librerías públicas que implantan el protocolo TSP en diversos lenguajes de programación:

- **BouncyCastle** (<http://www.bouncycastle.org>): Conjunto de librerías criptográficas que implementan el protocolo TSP en los lenguajes Java y C#
- **OpenTSA** (<http://www.opentsa.org>): Es una ampliación de la librería criptográfica OpenSSL que implementa el protocolo TSP en lenguaje C.
- **Digistamp** (<http://digistamp.com/toolkitDoc/MSToolKit.htm>): Toolkit basado en la librería criptográfica CryptoAPI de Microsoft que implementa el protocolo TSP en Visual Basic
- **IAIK**: Incluye librerías criptográficas en Java que implementan el protocolo TSP. Estas librerías son gratuitas únicamente para propósitos no comerciales
- **Adobe Reader**: La aplicación Adobe Reader 8 permite validar sellos de tiempo incluidos en documentos PDF.

## 5. Política de Sellado de Tiempo

### 5.1. Vista general

La presente política establece el conjunto de reglas utilizadas durante la emisión y el control de los tokens de sello de tiempo (TST), y regulan además el nivel de seguridad para la TSA.

Los tokens de sellado de tiempo son emitidos con una desviación máxima de 500ms.

El perfil del certificado de la TSA, utilizado en la firma de los TST, se ajusta a lo especificado por el IETF en RFC-3161. En la siguiente tabla se detallan los campos básicos de este perfil:


Campo	Contenido	I	C	T
<b>Versión</b>	v3	S		F
<b>Serial Number</b>	7F6A783DE9C70E0E	S		F
<b>Signature Algorithm</b>	SHA1withRSAEncryption	S		F
<b>Issuer Distinguished Name</b>	CN=ACEDICOM Servidores,OU=PKI,O=EDICOM,C=ES	S		F
<b>Validez</b>	2 años	S		F
<b>Subject Public Key Info</b>	Tipo de clave: RSA Longitud de la clave: 2048 bits	S		F
<b>Subject</b>		S		D
CommonName (CN)	ACEDICOM TSA	S		D
Organization Unit (OU)	PKI	S		D
Organization (O)	EDICOM	S		D
Country (C)	ES	S		D
<b>SubjectKeyIdentifier</b>	E3:CB:B2:FF:06:9F:3C:D6:37:A7:AD:6F:8B:93:61:43:B8:5A:79:81	S		D
<b>AuthorityKeyIdentifier</b>	Keyid:AD:C1:E8:40:30:96:01:16:52:02:41:38:3D:B6:51:F3:9E:82:46:4B	S		F
<b>BasicConstraints</b>		S	X	F
CA	Falso	S	X	F
pathLength	No aplicable (0)			F
<b>KeyUsage</b>	digitalSignature	S	X	F
<b>ExtendedKeyUsage</b>	TimeStamping	S	X	F

La TSA que proporciona sus servicios bajo la estructura de la ACEDICOM, emite los sellos de tiempo acorde a la recomendación ETSI TS 101 861. Cada sello de tiempo incluye el identificador de la política, descrito en el capítulo 5.2 "Identificación de la política de sellado de tiempo", de la presente política.

El servicio de Sellado de Tiempo es accesible vía http en la dirección **[tsa.acedicom.edicomgroup.com](http://tsa.acedicom.edicomgroup.com)** por el puerto 9026 y 9027 (https). La URL a definir en el cliente **<http://tsa.acedicom.edicomgroup.com:9026>** o **<https://tsa.acedicom.edicomgroup.com:9027>**.

### 5.2. Identificación de la política de sellado de tiempo

La información de la política, que controla la emisión y el control de los tokens de sellado de tiempo, esta definida en la siguiente Tabla :

	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>		
		Edición 1	Página 10

<b>Nombre del documento</b>	Política de sellado de tiempo de la ACEDICOM
<b>Versión del documento</b>	V1.1
<b>Estado del documento</b>	Vigente
<b>OID (Object Identifier)</b>	1.3.6.1.4.1.30051.2.1.3.1
<b>Fecha de emisión</b>	27 de enero de 2011
<b>Fecha de expiración</b>	No aplicable.
<b>CPS relacionada</b>	Declaración de Prácticas de Certificación (CPS) de la ACEDICOM. Versión 1.5 OID: 1.3.6.1.4.1.30051.2.1.1.1 Disponible en : <a href="http://acedicom.edicomgroup.com">http://acedicom.edicomgroup.com</a>
<b>Localización</b>	<a href="http://acedicom.edicomgroup.com">http://acedicom.edicomgroup.com</a>

El identificador de la política de la Autoridad de Sellado de Tiempo de la ACEDICOM esta incluido en cada sello de tiempo. También aparece en el documento de Declaración de Términos y Condiciones de Uso de la Autoridad de Sellado de Tiempo.

### 5.3. Aplicación del sellado de tiempo

Los sellos de tiempo emitidos por la Autoridad de Sellado de Tiempo de la ACEDICOM pueden emplearse para garantizar las transacciones y el no repudio en procesos en los cuales intervenga cualquier organismo o entidad con los que se haya formalizado un convenio de certificación.

	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>	
	Edición 1	Página 11

## **6 OBLIGACIONES Y RESPONSABILIDADES**

### **6.1 Obligaciones de la TSA**

#### **6.1.1 Obligaciones**

ACEDICOM, actuando como Autoridad de Sellado de Tiempo (TSA) se obliga a:

- Respetar lo dispuesto en esta Política de Sellado de Tiempo.
- Proteger sus claves privadas de forma segura.
- Emitir sellos de tiempo conforme a esta Política y a los estándares de aplicación.
- Garantizar que la hora y fecha incluidas en los sellos se mantienen dentro de los márgenes precisión establecida en el contrato entre el cliente y ACEDICOM, que en ningún caso pueden ser superiores a lo establecido en el apartado 5.1.
- Emitir sellos de tiempo según la información enviada por el cliente y libres de errores de entrada de datos.
- Emitir sellos de tiempos cuyo contenido mínimo sea el definido por la normativa vigente, cuando sea aplicable.
- Publicar esta Política de Sellado de Tiempo
- Informar sobre las modificaciones de la Política de Sellado de Tiempo a clientes y terceros que confían en los sellos de tiempo.
- Establecer los mecanismos de generación y custodia de la información relevante en las actividades descritas, protegiéndolas ante pérdida, destrucción o falsificación.
- Custodiar los sellos de tiempo emitidos para los clientes que contraten el servicio de custodia
- ACEDICOM, en su actividad de prestación de servicios de certificación, responderá por el incumplimiento de lo establecido en esta Política de Sellado de Tiempo y, allí donde sea aplicable, por lo que dispone la Ley 59/2003, de 19 de diciembre, de firma electrónica o su normativa de desarrollo.

Sin perjuicio de lo anterior ACEDICOM no garantizará los algoritmos y estándares criptográficos utilizados ni responderá de los daños causados por ataques externos a los mismos, siempre que hubiere aplicado la diligencia debida según el estado de la técnica en cada momento, y hubiere actuado conforme a lo dispuesto en las presentes Políticas de TSA y en la legislación vigente, donde sea aplicable.

#### **6.1.2 Responsabilidad financiera**

No aplicable por no tratarse de un servicio de emisión de certificados reconocidos según lo estipulado en la Ley de 59/2003 de Firma electrónica. La TSA no se hace responsable en caso de pérdidas por transacciones

	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>	
	Edición 1	Página 12

### 6.1.3 Exoneración de responsabilidad

ACEDICOM no será responsable en ningún caso cuando se encuentre ante cualquiera de estas circunstancias:

- Estado de Guerra, desastres naturales, funcionamiento defectuoso de los servicios eléctricos, las redes telemáticas y/o telefónicas o de los equipos informáticos utilizados por el Cliente o por los Terceros, o cualquier otro caso de fuerza mayor.
- Por el uso indebido o fraudulento de los sellos de tiempo.
- Por el uso indebido de la información contenida en el Certificado o en la CRL.
- Por el contenido de los mensajes o documentos sellados.
- En relación a acciones u omisiones del Cliente
- Falta de veracidad de la información suministrada para emitir el sello
- Negligencia en la conservación de sus datos de acceso al servicio de sellado de tiempo, en el aseguramiento de su confidencialidad y en la protección de todo acceso o revelación.
- Extralimitación en el uso del sello de tiempo, según lo dispuesto en la normativa vigente y en la presente Política de TSA
- En relación a acciones u omisiones del Usuario, tercero que confía en el certificado:
- Falta de comprobación de la suspensión o pérdida de vigencia del certificado electrónico de la TSA publicada en el servicio de consulta sobre la vigencia de los certificados o falta de verificación de la firma electrónica.

### 6.1.4 Cese de la actividad de la TSA

Antes del cese de su actividad la TSA realizará las siguientes actuaciones:

- Informará a todos los suscriptores, usuarios o entidades con los cuales tenga acuerdos u otro tipo de relación del cese con la anticipación mínima de 2 meses, o el periodo que establezca la legislación vigente.
- Revocará toda autorización a entidades subcontratadas para actuar en nombre de la TSA en el procedimiento de emisión de sellos de tiempo.
- Informará a la administración competente, con la antelación indicada, el cese de su actividad y el destino que se vaya a dar a los sellos de tiempo emitidos hasta la fecha, especificando, en su caso, si se va a transferir la gestión y a quien.

## 6.2 CLIENTE

El Cliente estará obligado a cumplir con lo dispuesto por la normativa y además a:

- Respetar lo dispuesto en los documentos contractuales firmados con la TSA.
- Verificar la corrección de la firma digital del sello de tiempo y la validez del certificado de la TSA en el momento de firmarlo.

 <b>edicom</b> <i>connecting business</i>	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>		
		Edición 1	Página 13

### **6.3 TERCERO QUE CONFÍA EN LOS SELLOS DE TIEMPO**

Será obligación de los Usuarios cumplir con lo dispuesto por la normativa vigente y además:

- Verificar la corrección de la firma del sello de tiempo y la validez del certificado de la TSA en el momento de firmarlo.

	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>	
	Edición 1	Página 14

## 7 REQUERIMIENTOS OPERACIONALES

---

### 7.1 OBTENCIÓN DEL TIEMPO FIABLE

Las fuentes de tiempos utilizadas, basadas en el protocolo NTP (Network Time Protocol) se auto calibran por distintos caminos, haciendo que la exactitud no disminuya por debajo de los requerimientos especificados (capítulo 5.1 del presente documento), utilizando como referencia la del Real Instituto y Observatorio de la Armada en San Fernando (ROA) y la sincronización GPS vía satélite. Se disponen de distintos caminos de sincronización de forma que la manipulación de los sistemas no afecta a la exactitud del sello de tiempo.

La Sección de Hora del ROA tiene como misión principal el mantenimiento de la unidad básica de Tiempo, declarado a efectos legales como Patrón Nacional de dicha unidad, así como el mantenimiento y difusión oficial de la escala "Tiempo Universal Coordinado" (UTC(ROA)), considerada a todos los efectos como la base de la hora legal en todo el territorio nacional (R. D. 23 octubre 1992, núm. 1308/1992).

A nivel interno la ACEDICOM dispone de mecanismos de seguridad que evitan la manipulación física de sus sistemas (información adicional en la CPS, apartado 5.1 "Controles de Seguridad Física").

La ACEDICOM incorpora mecanismos que detectan diferencias entre el tiempo suministrado y el que se incluye en los sellos de tiempo. El cálculo del tiempo se realiza de acuerdo al protocolo.

### 7.2 CERTIFICADO DE TSA

#### 7.2.1. Generación de claves de la TSA

Las claves de la TSA se generan en módulo de seguridad hardware (en adelante HSM), que cumple con el estándar NIST FIPS 140-2 nivel 3, por personal autorizado de la ACEDICOM. La descripción de los roles y controles del personal puede encontrarse en la CPS, en el apartados 5.2 "Controles Procedimentales" [1].

El entorno de generación de las claves cumple los requisitos normativos impuestos por la ACEDICOM, de acuerdo con la CPS y cumplen con los requerimientos descritos en ISO 15408 (Information technology. Security techniques. Evaluation criteria for IT security).

El algoritmo y tamaño de claves se describen en el capítulo 5.1 "Vista general" de esta política, cumpliendo lo referenciado por ETSI SR 002 176.

	<b>POLITICA DE SELLADO DE TIEMPO (TIMESTAMPING)</b>	
	Edición 1	Página 15

## 7.2.2. Protección de la clave privada de la TSA

Los niveles de seguridad del HSM donde se almacena la clave se describen en el capítulo 7.2.1 “Generación de la clave de la TSA” de esta política.

Esta clave se encuentra bajo control multipersonal. Se encuentra dividida en varios fragmentos y es necesario un mínimo de dos de estos fragmentos para recomponer la clave.

Las copias de Backup de la clave privada se almacenan cifradas en archivos seguros ignífugos.

## 7.2.3 Publicación del certificado de TSA

El certificado de la TSA, que incluye su clave publica, se distribuye utilizando los mecanismos facilitados por la ACEDICOM principalmente a través del sitio Web <http://acedicom.edicomgroup.com>.

## 7.2.3 Cambio de certificado de TSA

El certificado de la TSA puede ser cambiado en cualquier momento por otro certificado de TSA igualmente válido.

Este cambio no se comunicará a los usuarios del servicio, los cuales deberían confiar en todos los sellos emitidos por ACEDICOM y firmados con certificados válidos de TSA dentro de la jerarquía de certificación.

## 7.3 IMPLEMENTACIÓN DE LA SOLICITUD Y RESPUESTA DE SELLOS DE TIEMPO

Las solicitudes y respuestas de sellos se se adherirán a la sintaxis de la especificación “**RFC3161 Time Stamp Protocol (TSP)**” descrito en el Apartado 3.4. “Time-Stamp Protocol via http” de la especificación, con las restricciones de la norma ETSI TS 101 861.

La URL del servicio de Sellado de Tiempo de ACEDICOM es la especificada en el apartado 5.1 de este documento

### 7.3.1. Protocolo Timestamp vía HTTP

#### Formato de la petición

Content-Type: application/timestamp-query

<<the ASN.1 DER-encoded Time-Stamp Request message>>

#### Formato de la respuesta

Content-Type: application/timestamp-reply

<<the ASN.1 DER-encoded Time-Stamp Response message>>

## 7.4 FORMATO DE LOS MENSAJES

Existen dos tipos de mensajes: los que el cliente envía a la TSA, y los que envía la TSA al cliente. Todos los mensajes están representados en notación ASN.1

### 7.4.1 Timestamp Request

Este mensaje lo utiliza la entidad que quiere un sello de tiempo (solicitante) para acceder al servicio que ofrece una TSA. Tiene el siguiente formato:

```

TimeStampReq ::= SEQUENCE {
    version                INTEGER { v1(1) },
    messageImprint        MessageImprint,
    reqPolicy              TSAPolicyId                OPTIONAL,
    nonce                  INTEGER                    OPTIONAL,
    certReq                BOOLEAN                    DEFAULT FALSE,
    extensions              [0] IMPLICIT Extensions OPTIONAL
}

```

Campo	Descripción
<b>version</b>	Versión de la petición TimeStamp (v1)
<b>messageImprint</b>	OID del algoritmo hash y el valor del hash de los datos
<b>reqPolicy</b>	OID de la política de la TSA  Indica a la TSA la política bajo la cuál quiere que se proporcione el sello
<b>Nonce</b>	Si se incluye el <b>nonce</b> permite al cliente comprobar el retardo en la respuesta cuando no se dispone de reloj local. La respuesta debe contener este mismo número o se rechazará. El <b>nonce</b> es un número aleatorio con una elevada probabilidad de que el cliente lo genere una única vez (entero de 64 bits).
<b>CertReq</b>	Si el campo <b>certReq</b> está presente y con valor true, la clave publica de la TSA debe estar referenciada por el identificador ESSCertID dentro de un atributo SigningCertificate de la estructura SignedData en la respuesta. Ese campo además puede contener otros certificados.  Si falta el campo <b>certReq</b> o tiene valor false entonces, el campo SigningCertificate de la estructura SignedData no debe aparecer en la respuesta.
<b>extensions</b>	Es una forma de permitir añadir nuevos campos en el futuro. Si se incluye algún campo de extensión que la TSA no reconozca, ésta devolverá un mensaje de error de extensión no aceptada (unacceptedExtension).  Más información en: RFC 2459

```

MessageImprint ::= SEQUENCE {
    hashAlgorithm          AlgorithmIdentifier,
    hashedMessage          OCTET STRING
}

```

Campo	Descripción
<b>hashAlgorithm</b>	<p>OID del algoritmo hash:</p> <p>El algoritmo de hash indicado en <i>hashAlgorithm</i> debería ser uno conocido por la TSA. También comprobará que sea suficientemente fuerte. Si la TSA no reconoce el algoritmo usado o piensa que es débil, entonces la TSA denegará el servicio al cliente devolviendo un <b>pkiStatusInfo</b> de 'bad_alg'.</p>
<b>hashedMessage</b>	Este campo contiene el hash de los datos que se quiere sellar. La longitud del hash tiene que coincidir con la longitud de hash del algoritmo utilizado

El mensaje **Timestamp Request** no identifica al cliente, y esta información no es validada por la TSA. En el caso en que la TSA requiera su identidad deberá utilizar un mecanismo alternativo de identificación o autenticación

### 7.4.2 Timestamp Response

Es la respuesta que la TSA da a una mensaje *time stamp request*. Tiene la siguiente representación:

```

TimeStampResp ::= SEQUENCE {
    status                PKIStatusInfo,
    timeStampToken        TimeStampToken OPTIONAL
}

```

Campo	Descripción
<b>Status</b>	Estado de la respuesta. Ver sección 3.2.3 del RFC 2510
<b>timeStampToken</b>	Este campo que contiene la marca de tiempo generado. Es una estructura <b>ContentInfo</b> que encapsula información firmada en una estructura <b>TSTInfo</b> . Está definida en la RFC 2630.

```

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

```

Campo	Descripción
<b>Status</b>	<p>Estado de la respuesta:</p> <ul style="list-style-type: none"> <li>• <b>granted(0)</b>: Marca de tiempo presente.</li> <li>• <b>grantedWithMods(1)</b>: Marca de tiempo presente con modificaciones.</li> <li>• <b>rejection(2)</b>: Petición rechazada</li> <li>• <b>waiting(3)</b>: Esperando</li> <li>• <b>revocationWarning(4)</b> : Advertencia de revocación inminente</li> <li>• <b>revocationNotification (5)</b>: Notificación de revocación</li> </ul>
<b>statusString</b>	Puede usarse para indicar eventos de error
<b>failInfo</b>	<p>Causas del fallo:</p> <ul style="list-style-type: none"> <li>• <b>badAlg(0)</b>: Identificador de algoritmo no soportado</li> <li>• <b>badRequest(2)</b>: Transacción no permitida o soportada</li> <li>• <b>badDataFormat(5)</b>: Datos enviados con formato incorrecto</li> </ul>

- **timeNotAvailable(14):** Origen de tiempo no disponible
- **unacceptedPolicy(15):** Política solicitada no soportada
- **unacceptedExtension(16):** Extensión no soportada
- **addInfoNotAvailable(17):** Información adicional no disponible
- **systemFailure(25):** Error del sistema

```

TSTInfo ::= SEQUENCE {
    version                INTEGER { v1(1) },
    policy                 TSAPolicyId,
    messageImprint        MessageImprint,
    serialNumber          INTEGER,
    genTime               GeneralizedTime,
    accuracy              Accuracy OPTIONAL,
    ordering              BOOLEAN DEFAULT FALSE,
    nonce                INTEGER OPTIONAL,
    tsa                  [0] GeneralName OPTIONAL,
    extensions            [1] IMPLICIT Extensions OPTIONAL
}

```

Campo	Descripción
<b>Version</b>	Versión de la respuesta TimeStamp (v1)
<b>ReqPolicy</b>	OID de la política de la TSA Indica la política de la TSA bajo la cual se proporciona el sello, si se ha generado el sello, será igual al del mensaje de petición
<b>messageImprint</b>	OID del algoritmo hash y el valor del hash de los datos. Debe tener el mismo valor que el campo correspondiente de la petición.
<b>SerialNumber</b>	Es un entero asignado por la TSA y debe ser único para cada sello que genere. Por tanto, un sello será identificado por el nombre de la TSA que lo generó y el número de serie asignado. Permite hasta 160 bits
<b>genTime</b>	Es el instante de tiempo en el que se creó el sello. Tanto ISO como el IETF expresan el instante de tiempo referido a la escala <i>UTC</i> , para evitar confusiones con las horas locales. El formato debe ser el siguiente: <b>CC YY MM DD hh mm ss Z</b> <ul style="list-style-type: none"> <li>• CC representa el siglo (19-99)</li> <li>• YY representa el año (00-99)</li> <li>• MM representa el mes (01-12)</li> <li>• DD representa el día (01-31)</li> <li>• hh representa la hora (00-23)</li> <li>• mm representa los minutos (00-59)</li> <li>• ss representa los segundos (00-59)</li> <li>• Z viene de <i>zulu</i>, que es como se conoce a la escala <i>UTC</i></li> </ul>
<b>accuracy</b>	Representa la desviación del tiempo UTC contenido en genTime, en los casos que sea necesario, proporciona una precisión incluso de microsegundos:  <pre> Accuracy ::= SEQUENCE {     seconds [1] Integer OPTIONAL,     millis [2] Integer (1..999) OPTIONAL,     micros [3] Integer (1..999) OPTIONAL, } </pre> <p>Cuando este campo no está presente la precisión puede obtener a través de otros métodos, por ejemplo <b>TSAPolicyId</b>.</p>
<b>ordering</b>	Si falta el campo ordering o está presente y tiene valor false, entonces el

	<p>campo <b>genTime</b> solo indica el momento en el que la marca de tiempo ha sido creada por la TSA.</p> <p>En este caso, el orden de la marcas de tiempo emitidas por una misma TSA o distintas TSAs solo es posible cuando la diferencia entre el genTime de la primera marca de tiempo es mayor que la suma de las precisiones del genTime de cada marca de tiempo.</p>
<b>nonce</b>	El <b>nonce</b> es un número aleatorio con una elevada probabilidad de que el cliente lo genere una única vez (entero de 64 bits). Debe tener el mismo valor que el campo correspondiente de la petición.
<b>Tsa</b>	Identificador de la TSA
<b>extensions</b>	Es una forma de permitir añadir nuevos campos en el futuro. Más información en: RFC 2459

### 7.4.3 Validate Request

Es el mensaje que una entidad envía a una TSA cuando quiere comprobar la validez y la autenticidad de un sello. En la RFC 3161 del IETF no se contempla el proceso de verificación del sello de tiempo:

```
ValidateRequest ::= SEQUENCE {
    version      INTEGER { v1(0) },
    tst          TimeStampToken,
    requestID    [0] OCTET STRING OPTIONAL
}
```

- *tst*: contiene el sello que se quiere verificar
- *requestID*: identificador que se utiliza para vincular una petición con su respuesta

### 7.4.4 Validate Reply

La TSA envía este mensaje como respuesta a una petición de verificación:

```
ValidateReply ::= SEQUENCE {
    version      INTEGER { v1(0) },
    status       PKIStatusInfo,
    tst          TimeStampToken,
    requestID    [0] OCTET STRING OPTIONAL
}
```

## 7.5 EJEMPLO DE CONEXIÓN EN JAVA

A continuación se describe un ejemplo de conexión a la TSA de EDICOM mediante código JAVA y utilizando las librerías *bouncycastle* ([http://www.bouncycastle.org/latest\\_releases.html](http://www.bouncycastle.org/latest_releases.html)): *bcprov-jdkxx-145.jar* y *bctsp-jdkxx-145.jar*

```
package com.edicom.map.net;

import java.io.*;
import java.math.*;
import java.net.*;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import org.bouncycastle.asn1.cmp.*;
import org.bouncycastle.asn1.x509.*;
import org.bouncycastle.tsp.*;
import org.bouncycastle.util.encoders.Base64;

public class TimestampClient {
    /** URL of the Time Stamp Authority */
    protected String tsaURL;
    /** TSA Username */
    protected String tsaUsername;
    /** TSA password */
    protected String tsaPassword;
    /** Estimate of the received time stamp token */
    protected int tokSzEstimate;

    /**
     * Creates an instance of a TSAClient that will use BouncyCastle.
     * @param url String - Time Stamp Authority URL (i.e.
     * "http://tsatest1.digistamp.com/TSA")
     */
    public TimestampClient(String url) {
        this(url, null, null, 4096);
    }

    /**
     * Creates an instance of a TSAClient that will use BouncyCastle.
     * @param url String - Time Stamp Authority URL (i.e.
     * "http://tsatest1.digistamp.com/TSA")
     * @param username String - user(account) name
     * @param password String - password
     */
    public TimestampClient(String url, String username, String password) {
        this(url, username, password, 4096);
    }

    /**
     * Constructor.
     * Note the token size estimate is updated by each call, as the token
     * size is not likely to change (as long as we call the same TSA using
     * the same imprint length).
     * @param url String - Time Stamp Authority URL (i.e.
     * "http://tsatest1.digistamp.com/TSA")
     * @param username String - user(account) name
     * @param password String - password
     * @param tokSzEstimate int - estimated size of received time stamp token (DER
     encoded)
     */
    public TimestampClient(String url, String username, String password, int
    tokSzEstimate) {
        this.tsaURL = url;
        this.tsaUsername = username;
        this.tsaPassword = password;
        this.tokSzEstimate = tokSzEstimate;
    }
}
```

```
/**
 * Get timestamp token - Bouncy Castle request encoding / decoding layer
 */
protected byte[] getTimeStampToken(byte[] imprint) throws Exception {
    byte[] respBytes = null;
    try {
        // Setup the time stamp request
        TimeStampRequestGenerator tsqGenerator = new
TimeStampRequestGenerator();
        tsqGenerator.setCertReq(true);
        // tsqGenerator.setReqPolicy("1.3.6.1.4.1.601.10.3.1");
        BigInteger nonce = BigInteger.valueOf(System.currentTimeMillis());
        TimeStampRequest request =
tsqGenerator.generate(X509ObjectIdentifiers.id_SHA1.getId() , imprint, nonce);
        byte[] requestBytes = request.getEncoded();

        // Call the communications layer
        respBytes = getTSAResponse(requestBytes);

        // Handle the TSA response
        TimeStampResponse response = new TimeStampResponse(respBytes);

        // validate communication level attributes (RFC 3161 PKIStatus)
        response.validate(request);
        PKIFailureInfo failure = response.getFailInfo();
        int value = (failure == null) ? 0 : failure.intValue();
        if (value != 0) {
            // @todo: Translate value of 15 error codes defined by
PKIFailureInfo to string
            throw new Exception("[invalid.tsa.1.response.code] url: " + tsaURL
+ ". status: " + String.valueOf(value));
        }
        // @todo: validate the time stamp certificate chain (if we want
// assure we do not sign using an invalid timestamp).

        // extract just the time stamp token (removes communication status
info)
        TimeStampToken tsToken = response.getTimeStampToken();
        if (tsToken == null) {
            throw new Exception("[tsa.1.failed.to.return.time.stamp.token] url:
" + tsaURL + ". status: " + response.getStatusString());
        }
        TimeStampTokenInfo info = tsToken.getTimeStampInfo(); // to view
details
        byte[] encoded = tsToken.getEncoded();
        long stop = System.currentTimeMillis();

        // Update our token size estimate for the next call (padded to be safe)
        this.tokSzEstimate = encoded.length + 32;
        return encoded;
    } catch (Exception e) {
        throw e;
    } catch (Throwable t) {
        throw new Exception("[failed.to.get.tsa.response.from.1] url: " +
tsaURL + ". status: " + t);
    }
}
```

```
/**
 * Get timestamp token - communications layer
 * @return - byte[] - TSA response, raw bytes (RFC 3161 encoded)
 */
protected byte[] getTSAResponse(byte[] requestBytes) throws Exception {
    // Setup the TSA connection
    URL url = new URL(tsaURL);
    URLConnection tsaConnection;
    tsaConnection = (URLConnection) url.openConnection();

    tsaConnection.setDoInput(true);
    tsaConnection.setDoOutput(true);
    tsaConnection.setUseCaches(false);
    tsaConnection.setRequestProperty("Content-Type", "application/timestamp-
query");
    //tsaConnection.setRequestProperty("Content-Transfer-Encoding", "base64");
    tsaConnection.setRequestProperty("Content-Transfer-Encoding", "binary");

    if ((tsaUsername != null) && !tsaUsername.equals("")) {
        String userPassword = tsaUsername + ":" + tsaPassword;
        tsaConnection.setRequestProperty("Authorization", "Basic " +
            new String(Base64.encode(userPassword.getBytes())));
    }
    OutputStream out = tsaConnection.getOutputStream();
    out.write(requestBytes);
    out.close();

    // Get TSA response as a byte array
    InputStream inp = tsaConnection.getInputStream();
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    byte[] buffer = new byte[1024];
    int bytesRead = 0;
    while ((bytesRead = inp.read(buffer, 0, buffer.length)) >= 0) {
        baos.write(buffer, 0, bytesRead);
    }
    byte[] respBytes = baos.toByteArray();

    String encoding = tsaConnection.getContentEncoding();
    if (encoding != null && encoding.equalsIgnoreCase("base64")) {
        respBytes = Base64.decode(new String(respBytes));
    }
    return respBytes;
}

public byte[] generateImprint(byte[] data) throws NoSuchAlgorithmException {
    MessageDigest md = MessageDigest.getInstance("SHA-1");
    return md.digest(data);
}

public byte[] generateImprintFromFile(String filename) throws Exception {
    byte[] data = loadBytesFromFile(filename);
    return generateImprint(data);
}

public byte[] loadBytesFromFile(String name) {
    FileInputStream in = null;

    try {
        in = new FileInputStream(name);
        ByteArrayOutputStream buffer = new ByteArrayOutputStream();
        int ch;
        while ((ch = in.read()) != -1)
            buffer.write(ch);
        return buffer.toByteArray();
    } catch (IOException e) {
        if (in != null) {
            try {
                in.close();
            } catch (IOException e2) {
            }
        }
        return null;
    }
}
```

```
public void writeFile(byte[] data, String filename) throws IOException{
    OutputStream out = new FileOutputStream(filename);
    try {
        out.write(data);
    } finally {
        out.close();
    }
}

public static void main(String[] args) {
    String url = "http://tsa.acedicom.edicomgroup.com:9026/";
    if (args.length > 0) {
        try {
            String user = "usuario";
            String password = "password";
            TimestampClient tsc = new TimestampClient(url, user, password);
            byte[] imprint = tsc.generateImprintFromFile(args[0]);
            byte[] response = tsc.getTimeStampToken(imprint);
            tsc.writeFile(response, args[0] + ".tsr");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```