



**PKITS**  
**Public Key Infrastructure with Time-Stamping Authority**

**ETS PROJECT: 23.192**

**Deliverable D7b**  
**Description and Results of the EDI Documents Time-Stamping Protocol Implementations**

**Produced by: *UPC***  
**Date of issue: *18th December 1998***  
**Revision Number: 2**

## TABLE OF CONTENTS

<b>1. EXECUTIVE SUMMARY .....</b>	<b>5</b>
<b>2. REFERENCES .....</b>	<b>5</b>
<b>3. DESCRIPTION OF THE STRUCTURED DATA (EDI MESSAGES) TIME-STAMPING TESTBED: SYSTEM ARCHITECTURE .....</b>	<b>6</b>
3.1. SYSTEM ARCHITECTURE OF THE PKITS EDITIMESTAMPING SERVER.....	7
3.1.1. <i>TimeStamping service infrastructure</i> .....	7
3.1.2. <i>Hash functions, digital signatures and security certificates</i> .....	7
3.1.3. <i>Time synchronisation</i> .....	7
3.1.4. <i>TimeStamps storage</i> .....	8
3.1.5. <i>Supported platforms</i> .....	8
3.1.6. <i>Communications and service access</i> .....	8
3.2. SYSTEM ARCHITECTURE OF THE PKITS EDITIMESTAMPING CLIENT .....	9
3.2.1. <i>TimeStamping service infrastructure</i> .....	9
3.2.2. <i>Hash functions, digital signatures and security certificates</i> .....	9
3.2.3. <i>Supported platforms</i> .....	9
3.2.4. <i>Communications and service access</i> .....	10
3.3. GRAPHICAL REPRESENTATION OF THE SERVICE ARCHITECTURE.....	13
<b>4. APPENDIX A: TIME-STAMPING EDI MESSAGES EXAMPLE .....</b>	<b>14</b>
4.1. USER'S ORIGINAL EDI MESSAGE .....	14
4.2. TIMESTAMP REQUEST EDI MESSAGE.....	15
4.3. TIMESTAMP TOKEN EDI MESSAGE .....	17
<b>5. APPENDIX B: BIBLIOGRAPHY.....</b>	<b>19</b>

## HISTORY

Version	date	Author	comment
1	1998-12-18	Manel Medina, Eduard Bel Serra, Juan Carlos Cruellas, Montserrat Rubia	1 <sup>st</sup> Version
2	1998-12-23	Manel Medina, Eduard Bel Serra, Juan Carlos Cruellas, Montserrat Rubia	2 <sup>nd</sup> Version

## GLOSSARY OF TERMS

<b><u>Specifications:</u></b>	
SHALL	Essential requirement. A requirement must be fulfilled or a feature implemented wherever this term occurs. The designer is requested, however, to indicate if one or more “shall requirements” would increase the cost or time unreasonably in relation to the total cost or design cost, in which case the specification may have to be revised.
SHOULD	Important requirement. Shall be implemented without or with minimum extra cost. Valid reasons in particular circumstances may allow ignoring such requirements.
MAY	Optional requirement. From case to case, it should be decided whether implementing it or not, in any case without exceeding the budget planned for the related activity.

<b><u>Technical:</u></b>	
CA	Certification Authority
DS	Digital Signature
EDI	Electronic data Interchange
EDIFACT	Electronic data Interchange for administration, commerce and transport
NTP	Time Network Protocol
SJWG	Security Joint Working Group
TS	Time-Stamping
TSA	Time-Stamping Authority
UTC	Coordinated Universal Time
UNA	Service string advice
UNB	Interchange header
UNH	Message header segment
UNT	Message trailer segment
USA	Security algorithm segment
USH	Security header segment
USR	Security result segment
UST	Security trailer segment

## 1. EXECUTIVE SUMMARY

This document describes the activities performed in order to validate the time-stamping protocols and supporting algorithms developed within the PKITS project.

This document details the implementation of the time-stamping protocols for structured data prototype. The term “structured data” refers to those messages complying with the UN/EDIFACT syntax rules (Electronic Data Interchange for Administration, Commerce and Transport) in a batch environment processing defined by ISO. An UN/EDIFACT message has an internal structure known by the TSS context, so the TSA embed a trusted time reference into some specific data fields from the original EDI document (EDI message) the requester submitted. Finally, the whole EDI document (EDI message) is returned to the requester in a secure manner.

As specified on previous deliverable [PKITS D5], this time-stamping service prototype follows the guidelines developed by "Security Joint Working Group" UN/EDIFACT (SJWG).

This document is organised as follows: Section 3 covers the system architecture and performance results. Section 4 is an extract of the client software log file. It details the EDI messages interchanged by the client and the server applications.

## 2. REFERENCES

- [EDIINT-1] EDIINT: "Propuesta de implantación de servicios de seguridad a mensajes EDIFACT". March 98
- [MIL85] Mills, D.L. "Network Time Protocol (NTP)". DARPA Network Working Group Report RFC-958, M/A-COM Linkabit, September 1985.
- [MIL89] Mills, D.L. (09/89), "Network Time Protocol (version 2) - specification and implementation". DARPA Network Working Group Report RFC-1119, University of Delaware.
- [MIL92] Mills, D.L. (03/92), "Network Time Protocol (version 3) - Specification, Implementation and Analysis". Network Working Group Report RFC-1119, University of Delaware.
- [PKITS D3] Architecture of Time-Stamping Service and Scenarios of Use: Service and Features, Deliverable D3 of ETS Project 23.192 Public Key Infrastructure with Time-Stamping Authority, May, 1998.
- [PKITS D5] Time-Stamping Service Functional Specification and Protocols for Structured Data: EDI documents, Deliverable D5 of ETS Project 23.192 Public Key Infrastructure with Time-Stamping Authority, August, 1998.
- [RFC 2313] RSA Laboratories, "PKCS #1 - RSA Encryption Standard", version 1.5, Nov. 1993.  
<http://www.rsa.com/rsalabs/pubs/PKCS>

### **3. DESCRIPTION OF THE STRUCTURED DATA (EDI MESSAGES) TIME-STAMPING TESTBED: SYSTEM ARCHITECTURE**

The time-stamping protocols for structured data prototype deals with messages complying with the UN/EDIFACT syntax rules (Electronic Data Interchange for Administration, Commerce and Transport) in a batch environment processing defined by ISO. As defined in previous deliverable [PKITS D3] and [PKITS D5], time-stamping protocols for structured data covers the provision of a time-stamping service when the time-stamping authority has knowledge of the internal structure of data.

Time-stamping protocols for structured data were specified on [PKITS D5] following standard EDIFACT version 4 syntax and we propose to use digital signatures in the interchanged messages through the syntax rules extensions for the EDI batch environment OSI 9735-5 proposed by SJWG.

We defined time-stamping protocols for requesting EDI timestamps, EDI timestamps renewal protocols and EDI timestamps verification protocols. We also distinguished three time-stamping protocol schemes on [PKITS D3] and [PKITS D5]: simple protocol, linking protocol and distributed protocol.

The implementation prototype described in this document supports time-stamping protocols for requesting EDI timestamps over EDI messages in simple or linking protocol schemes, and they have been closely implemented as defined in previous deliverables.

The time-stamping service prototype for structured data is composed by two software packages. We have developed a server application that implements a Time Stamping Authority (PKITS EDITimeStamping Server) and a client application that adds time-stamping functionality to end-users (PKITS EDITimeStamping Client). Both software packages were developed with Visual C++ 5.0 and EDIINT toolkit v.1.0

On the following chapters we describe the main features of the two packages

### **3.1. SYSTEM ARCHITECTURE OF THE PKITS EDITIMESTAMPING SERVER**

The resulting prototype features the following requirements considered during the specification of the PKITS EDITimeStamping Server application:

#### **3.1.1. TimeStamping service infrastructure**

We initially considered implement a standalone TSA that supports time-stamping protocols for requesting timestamps for batch environment EDI messages in simple or linking protocol schemes. Renewal and verification protocols were not implemented

#### **3.1.2. Hash functions, digital signatures and security certificates**

As defined on these service protocols, a TSA shall use certificates to digitally sign the responses sent to the requesters. Both server and client software packages uses EDIFACT certificates generated by the EDIINT toolkit.

The PKITS EDITimeStamping Server uses RSA algorithm and 1024 bits key length to digitally sign all the EDI timestamps sent to the requesters.

The PKITS EDITimeStamping Server supports SHA-1 hash algorithm in order to store a message digest of every EDI timestamp issued, and also to compute the RSA digital signature.

J++ASN Programming Language and Toolkit v.1.6085 and EDIINT toolkit are the software packages used to apply cryptographic operations (digitally signatures, hash computations) and to manage EDI time-stamping protocols.

#### **3.1.3. Time synchronisation**

The PKITS EDITimeStamping Server periodically synchronises its clock with a time server through NTP (Network Time Protocol). In the prototype scheme, we use the time server of the Universitat Politècnica de Catalunya SIRIUS.AC.UPC.ES.

The Intellisoft TimeSync software is used to synchronise the PKITS EDITimeStamping Server clock.

The PKITS EDITimeStamping Server stamps UTC (Coordinated Universal Time) time values on the timestamps with a precision of a second.

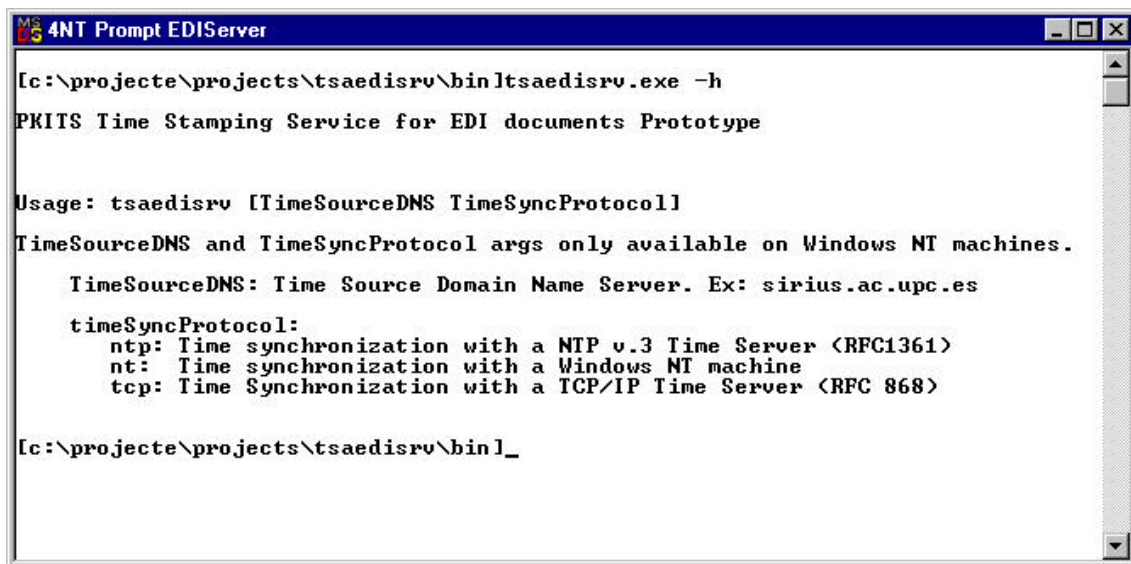
### 3.1.4. TimeStamps storage

The PKITS EDITimeStamping Server stores a SHA-1 message digest of every timestamp it issues and also logs all its activity in order to keep evidence of timestamps generation

### 3.1.5. Supported platforms

PKITS EDITimeStamping Server was developed on Windows NT 4 Server-Intel Pentium-II platform.

It runs on Windows NT and Windows 95/98 platforms as a console application. The following illustration shows the program arguments:



```
MS-DOS 4NT Prompt EDIServer

[c:\projecte\projects\tsaedisrv\bin]tsaedisrv.exe -h
PKITS Time Stamping Service for EDI documents Prototype

Usage: tsaedisrv [TimeSourceDNS TimeSyncProtocol]
TimeSourceDNS and TimeSyncProtocol args only available on Windows NT machines.
    TimeSourceDNS: Time Source Domain Name Server. Ex: sirius.ac.upc.es
    timeSyncProtocol:
        ntp: Time synchronization with a NTP v.3 Time Server <RFC1361>
        nt:  Time synchronization with a Windows NT machine
        tcp: Time Synchronization with a TCP/IP Time Server <RFC 868>

[c:\projecte\projects\tsaedisrv\bin]_
```

### 3.1.6. Communications and service access

PKITS EDITimeStamping Server uses TCP/IP protocols to interchange EDI time-stamping protocols. It listens user requests on port 310.

In order to access this EDI time-stamping service the user shall use the PKITS EDITimeStamping Client software packages that will be detailed on the following chapter.



### **3.2. SYSTEM ARCHITECTURE OF THE PKITS EDITIMESTAMPING CLIENT**

The resulting prototype features the following requirements considered during the specification of the PKITS EDITimeStamping Client application:

#### **3.2.1. TimeStamping service infrastructure**

The PKITS EDITimeStamping Client supports time-stamping protocols for requesting timestamps over EDI messages in simple or linking protocol schemes. It requires the filename of the EDI message to be timestamped, the target filename where the resulting timestamped EDI message will be stored and the PKITS EDITimeStamping Server to work with.

#### **3.2.2. Hash functions, digital signatures and security certificates**

The PKITS EDITimeStamping client software package also uses EDIFACT certificates generated by the EDIINT toolkit.

As the PKITS EDITimeStamping Server, the PKITS EDITimeStamping Client uses RSA algorithm and 1024 bits key length to digitally sign all the EDI timestamp requests submitted to the time-stamping server

The PKITS EDITimeStamping Server supports SHA-1 as the message-dependent hash function to compute the EDI time-stamping request RSA digital signature.

EDIINT toolkit is also the software used to apply cryptographic operations (digitally signatures, hash computations) and to manage EDI time-stamping protocols.

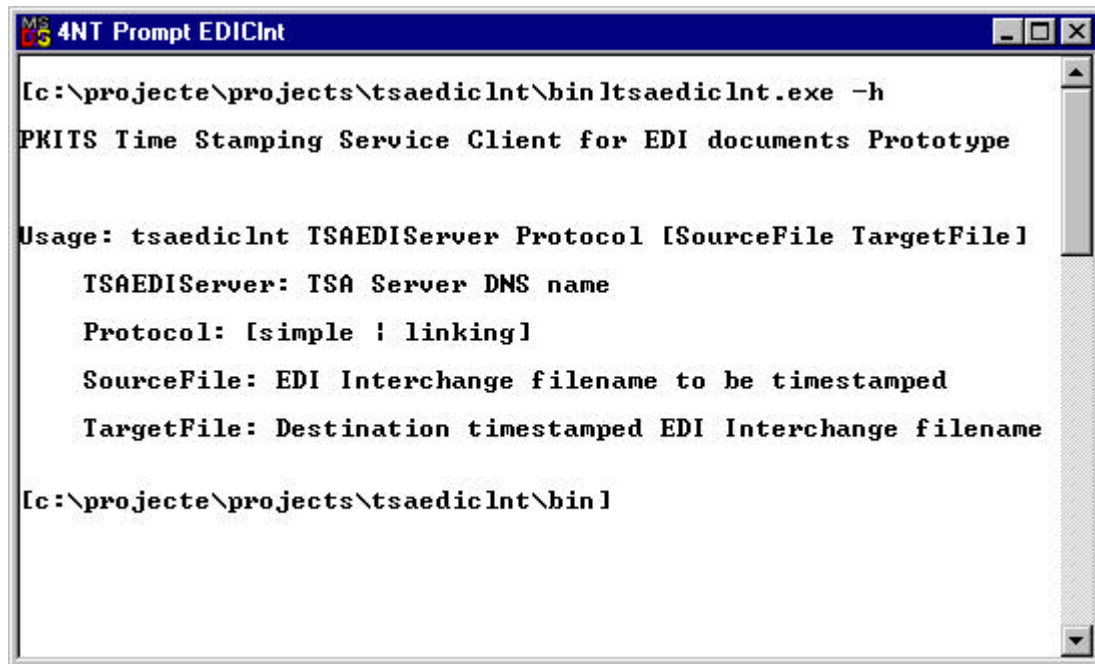
#### **3.2.3. Supported platforms**

PKITS EDITimeStamping Client runs on Windows 95/98 and Windows NT 4 platforms as a console application. This software can be easily called from other applications in order to extend their EDI capabilities with time-stamping client functionality.

### 3.2.4. Communications and service access

PKITS EDITimeStamping Client uses TCP/IP protocols to interchange time-stamping protocols with the PKITS EDITimeStamping Server through port 310.

The following illustration displays the client program arguments:



```
MS 4NT Prompt EDICInt
[c:\projecte\projects\tsaedicInt\bin\tsaedicInt.exe -h
PKITS Time Stamping Service Client for EDI documents Prototype

Usage: tsaedicInt TSAEDIServer Protocol [SourceFile TargetFile]
      TSAEDIServer: TSA Server DNS name
      Protocol: [simple | linking]
      SourceFile: EDI Interchange filename to be timestamped
      TargetFile: Destination timestamped EDI Interchange filename

[c:\projecte\projects\tsaedicInt\bin]
```

When executing the PKITS EDITimeStamping Client software, the performed operations are:

1. The source EDI message is encapsulated in an EDI message with the related header and trailer segments. The following table shows the segment groups that define the security header and trailer in syntax rules version 4:

S: Status of the service string character (M: Mandatory)

R: Maximum number of occurrences of the stand-alone data element or composite-data element in the segment

TAG	Name	S	R	
UNH	Message Header	M	1	
-----	Segment Group 1 -----	C	99	-----+
USH	Security Header (specifies a time-stamping request in simple or linking protocol scheme)	M	1	I
USA	Security Algorithm (SHA-1 as the hash algorithm used to compute RSA signature)	C	3	I
-----	Segment Group 2 -----	C	2	-----+ I
USC	Certificate (EDIFACT client certificate)	M	1	I I
USA	Security Algorithm	C	3	I I
USR	Security Result	C	1	-----+
EDI source message stored in SourceFile argument				
-----	Segment Group n -----	C	99	-----+
UST	Security Trailer	M	1	I
USR	Security Result (PKITS EDITimeStamping Client digital signature of the EDI time-stamping request message)	C	1	-----+
UNT	Message Trailer	M	1	

This encapsulation process, the segment groups and the data elements involved were detailed on [PKITS D5]

2. The resulting message is submitted to the PKITS EDITimeStamping Server and the application waits for a response. The PKITS EDITimeStamping Server will extract the EDI message encapsulated in the request. Then, it will generate the response by encapsulating the received message in a new EDI message with the related new header and trailer segments that will contain the stamped time. The following table exemplifies the time-stamping response sent to the requester by the server if simple protocol scheme is requested

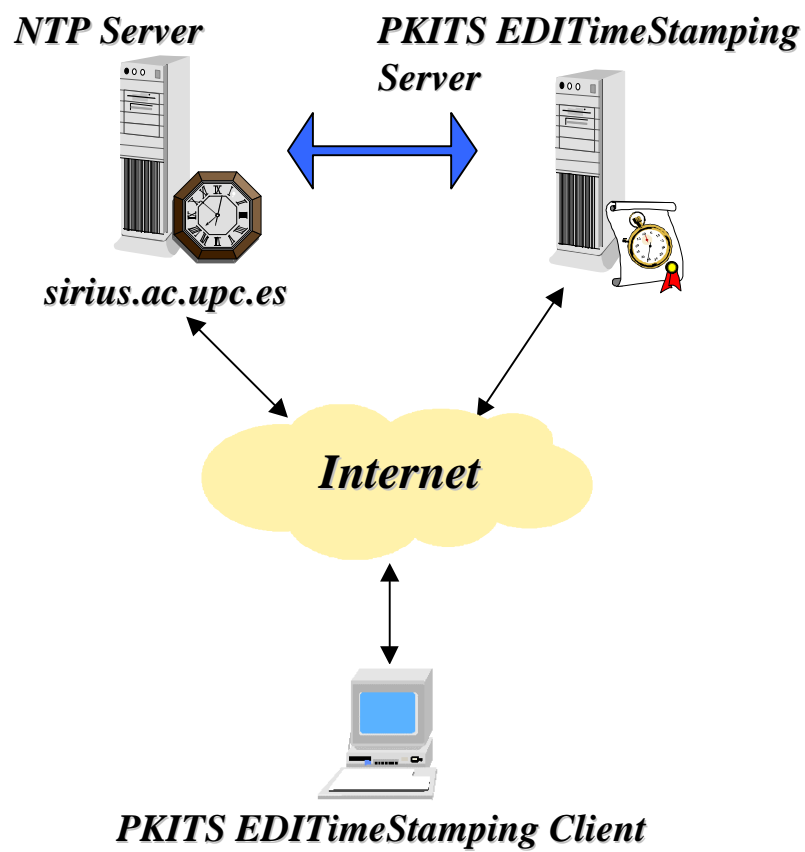
TAG	Name	S	R	
UNH	Message Header	M	1	
-----	Segment Group 1 -----	C	99	-----+
USH	Security Header (Specifies a time-stamping token in simple protocol scheme and encapsulates the stamped time in its S501 composite data element)	M	1	I
USA	Security Algorithm (SHA-1 as the hash algorithm used to compute RSA signature)	C	3	I
-----	Segment Group 2 -----	C	2	-----+ I
USC	Certificate (EDIFACT server certificate)	M	1	I I
USA	Security Algorithm	C	3	I I
USR	Security Result	C	1	-----+
EDI source message received in the time-stamping request				
-----	Segment Group n -----	C	99	-----+
UST	Security Trailer	M	1	I
USR	Security Result (PKITS EDITimeStamping Server digital signature of the EDI timestamped message)	C	1	-----+
UNT	Message Trailer	M	1	

If linking protocol scheme were requested, there would appear an additional segment group 1, segment group 2 and segment group 3 with the related linking information.

More detailed explanation can be found on [PKITS D5] about this process and linking protocol scheme

3. When receiving the response, the client software checks the digital signature from the PKITS EDITimeStamping Server and if no errors are encountered the timestamp received is displayed in a pretty-print manner and it is stored in the file specified as the TargetFile argument

### 3.3. GRAPHICAL REPRESENTATION OF THE SERVICE ARCHITECTURE



## 4. APPENDIX A: TIME-STAMPING EDI MESSAGES EXAMPLE

This appendix is a commented extract of the log file generated by the PKITS EDITimeStamping Client in a timestamp request transaction with Linking protocol scheme. A user requested a new timestamp over an EDI message and the server responded with a new one in Linking protocol scheme.

There are three data structures: The first one is the original EDI message the user wants to timestamp. The second one is the timestamp request EDI message, this is, the previous EDI message encapsulated in a security header and trailer pair with the related time-stamping service information. The last one is the timestamp returned by the PKITS EDITimeStamping Server

The more important data segments are indicated in bold-italic text style

### 4.1. USER'S ORIGINAL EDI MESSAGE

Lets we suppose that this is the message the user wants to timestamp. This message does not have any security header and trailer pair.

```
UNH+1+INVOIC:D:93A:UN:EAN007'BGM+380+INV1'DTM+137:19960702:102'  
RFF+DQ:A87564'RFF+ON:P7564'NAD+SU+8456789000007::9++AECOC::reg.  
merc. 37/82:pag. 6782 Barcelona+MALLORCA 288  
ENTLO.+BARCELONA++08037'RFF+VA:G08557985'NAD+BY+845678990000  
0::9++SERVIDOR DE MENSAJES+--  
++00000'RFF+VA:B23456765'PAT+21'DTM+13:19961015:102'MOA+23:98600'  
PAT+21'DTM+13:19961115:102'MOA+23:98600'ALC+A++++TD'MOA+8:20000  
'LIN+1++8456789900000:EN'IMD+F+M+:::AECOQUITOS 100  
GR.'QTY+47:2000'MOA+66:95000'PRI+AAB:50'TAX+7+VAT+++:::16'MOA+12  
4:15200'ALC+A+++1'PCD+1:5'MOA+204:5000'LIN+2++8456789567890:EN'IM  
D+F+M+:::AECOQUITOS 200  
GR.'QTY+47:1000'MOA+66:95000'PRI+AAB:100'TAX+7+VAT+++:::16'MOA+1  
24:15200'ALC+A+++1'PCD+1:5'MOA+204:5000'UNS+S'CNT+2:2'MOA+125:17  
0000'MOA+176:27200'MOA+79:190000'MOA+139:197200'MOA+131:-  
20000'TAX+7+VAT+++:::16'MOA+176:27200'UNT+47+20'
```

## 4.2. TIMESTAMP REQUEST EDI MESSAGE

In order to request a timestamp to the PKITS EDITimeStamping Server, the client software packages encapsulates the original EDI message into a security header and trailer pair as defined on previous [PKITS D5]. The resulting EDI secure message timestamping request is the following:

```
UNH+1+INVOIC:D:93A:UN:EAN007'USH+10+1+1+4+2+2++1::EDITSA:::TS  
AEDI2'USA+1:16::6'USC+81+3:KEYUSER1TS:222222222222::USER1TS+  
4:CAKEY1:1234567891234::CANAME1+3+2+2+1+++++2:20041217:181101  
+3:20041217:181101+4:20051217:181101'USA+3:0:1:10:1'USA+4:0:1:16:1'U  
SA+6:0:1:10:1+12:E177E00EADFB037E580D543ACAF9D004FE2A8BBA78  
D3F0FCA8AE62998F08E6DCB0A83248BC003817C791A9C1DEF4BDC281D  
489A991F1C0271E5CE9E4BD8FAC1F1BADAF47859B591407266613925B5  
68922B8E1E80B309E44646D723053E9C140429E1D427D73F960454956594  
0D01391263769E0AAC93724C4AB099FA84F39E1+13:03+14:1024'USR+1:8  
B8D38F85A11C3286099B773D493190EA398ED7635C3A07A2D7E906B0886  
F19731FDB62126BE292492CCC5A3D0D32327465F9F880689252A0530AC8  
B87831D5DC37B7C2569E1F98834C6977FB414B3F0CE0FF67EDC0D96968  
73CA22E97C926A5B597FC628F8D897A022259156CD5E08BC4D2250A6BE  
C2089E2E0C8EEDFD6CD1F'BGM+380+INV1'DTM+137:19960702:102'RFF+  
DQ:A87564'RFF+ON:P7564'NAD+SU+8456789000007::9++AECOC::reg.  
merc. 37/82:pag. 6782 Barcelona+MALLORCA 288  
ENTLO.+BARCELONA++08037'RFF+VA:G08557985'NAD+BY+845678990000  
0::9++SERVIDOR DE MENSAJES+--+  
++00000'RFF+VA:B23456765'PAT+21'DTM+13:19961015:102'MOA+23:98600'  
PAT+21'DTM+13:19961115:102'MOA+23:98600'ALC+A++++TD'MOA+8:20000  
'LIN+1++8456789900000:EN'IMD+F+M+:::AECOQUITOS 100  
GR.'QTY+47:2000'MOA+66:95000'PRI+AAB:50'TAX+7+VAT+++:::16'MOA+12  
4:15200'ALC+A+++1'PCD+1:5'MOA+204:5000'LIN+2++8456789567890:EN'IM  
D+F+M+:::AECOQUITOS 200  
GR.'QTY+47:1000'MOA+66:95000'PRI+AAB:100'TAX+7+VAT+++:::16'MOA+1  
24:15200'ALC+A+++1'PCD+1:5'MOA+204:5000'UNS+S'CNT+2:2'MOA+125:17  
0000'MOA+176:27200'MOA+79:190000'MOA+139:197200'MOA+131:-  
20000'TAX+7+VAT+++:::16'MOA+176:27200'UST+1+9'USR+1:A4A9F4F77FC  
CA35BA95FB67A1D32CCA857CED38F585B5CA3AE2E18109F22EC1B25D  
489ECB83C94D8BB5A1F9E789F7738FE304CFF5ACF194D48785FB405ED4  
108C89976AF488C42A0869B434140B705A0757B84766DA4FEAADD9E1A1  
25EBB925541D269C2C04A6C90A4EC9D91E0ABDDB19E52B19B2564F8D4  
DCDC8C3C7F531CDC'UNT+56+20'
```

The first bold-italic text paragraph is the segment group 1 and the segment group 2 of the secure EDI message. The time-stamping service related data fields are the following:

***USH+10+1+1+4+2+2++1::EDITSA::::TSAEDI2:*** this is the security header of the segment group 1. The time-stamping service related values are:

- 10      It indicates a time-stamping service request
- 4        The user request a new timestampin in linking protocol scheme
- EDITSA and TSAEDI2: Identifies PKITS EDITimeStamping Server

***USA+1:16:*** This USA segment indicates that SHA-1 is the hash algorithm used to compute RSA signature.

***USC+81+3:KEYUSER1TS:222222222222:::USER1TS+4:CAKEY1:1234567891234:::CANAME1+3+2+2+1+++++2:20041217:181101+3:20041217:181101+4:20051217:181101'USA+3:0:1:10:1'USA+4:0:1:16:1'USA+6:0:1:10:1+12:E177E00EADFB037E580D543ACAF9D004FE2A8BBA78D3F0FCA8AE62998F08E6DCB0A83248BC003817C791A9C1DEF4BDC281D489A991F1C0271E5CE9E4BD8FAC1F1BADAF47859B591407266613925B568922B8E1E80B309E44646D723053E9C140429E1D427D73F9604549565940D01391263769E0AAC93724C4AB099FA84F39E1+13:03+14:1024'USR+1:8B8D38F85A11C3286099B773D493190EA398ED7635C3A07A2D7E906B0886F19731FDB62126BE292492CCC5A3D0D32327465F9F880689252A0530AC8B87831D5DC37B7C2569E1F98834C6977FB414B3F0CE0FF67EDC0D9696873CA22E97C926A5B597FC628F8D897A022259156CD5E08BC4D2250A6BEC2089E2E0C8EEDFD6CD1F:*** This is the segment group 2. It encapsulates the requester EDIFACT certificate.

The second bold-italic text paragraph is the segment group n: security trailer and security result. It encapsulates the RSA digital digital signature of the timestamp request



### 4.3. TIMESTAMP TOKEN EDI MESSAGE

The timestamp token returned by the PKITS EDITSA Server is as follows:

```
UNH+1+INVOIC:D:93A:UN:EAN007'USH+10+2+1+4+2+2++1::EDITSA:::TS
AEDI2++9+1:19981223:100213'USA+1::16'USC+82+3:KEYSERVER1TS:111
1111111111::SERVER1TS+4:CAKEY1:1234567891234::CANAME1+3+2+2
+1+++++2:20041217:181213+3:20041217:181213+4:20051217:181213'USA+
3:0:1:10:1'USA+4:0:1:16:1'USA+6:0:1:10:1+12:BFC762E1E0C4AABE3577E
B5D454411EB2F2FDAB652B2DD211E7203B0343379D2C2B5F989819CA61
A6FB11A6A705F4F7995BD738C7404905CFD48F267BC3DEF82850EBB07C
0815735F5C4A16C9BDAE9E27DC31C274D141675A6C3686137BC5645C5C
9E44CC4F3D6E0650E52F75CC77DE110AA2DF05EE7313CFDA9F944D0A3
54D9+13:03+14:1024'USR+1:24FF95371B9F865C48E3B38E6DFB1CB69E26
84DC00184413E67D9FA84D8E12824FFB80971A8F2404D870B4F8D0FDFD0
4A20369F7D8FD0DDC37F4F8FCF3A57C1D1081B3172FF2A7085F223F552
EE1546F96B896BDDC8BC1FA43B6AD7ABB0A7F6A92C9BA8C0238E390E
FCF15840099ACBD779628E8BE111075FD56AF3884DEDDF1'USH+10+1+1
+4+2+2++1::EDITSA:::TSAEDI2++8+1:19981223:095815'USA+1::16'BGM+
380+INV1'DTM+137:19960702:102'RFF+DQ:A87564'RFF+ON:P7564'NAD+SU
+8456789000007::9++AECOC::reg. merc. 37/82:pag. 6782
Barcelona+MALLORCA 288
ENTLO.+BARCELONA++08037'RFF+VA:G08557985'NAD+BY+845678990000
0::9++SERVIDOR DE MENSAJES+--+
++00000'RFF+VA:B23456765'PAT+21'DTM+13:19961015:102'MOA+23:98600'
PAT+21'DTM+13:19961115:102'MOA+23:98600'ALC+A++++TD'MOA+8:20000
'LIN+1++8456789900000:EN'IMD+F+M+::AECOQUITOS 100
GR.'QTY+47:2000'MOA+66:95000'PRI+AAB:50'TAX+7+VAT+++::16'MOA+12
4:15200'ALC+A+++1'PCD+1:5'MOA+204:5000'LIN+2++8456789567890:EN'IM
D+F+M+::AECOQUITOS 200
GR.'QTY+47:1000'MOA+66:95000'PRI+AAB:100'TAX+7+VAT+++::16'MOA+1
24:15200'ALC+A+++1'PCD+1:5'MOA+204:5000'UNS+S'CNT+2:2'MOA+125:17
0000'MOA+176:27200'MOA+79:190000'MOA+139:197200'MOA+131:-
20000'TAX+7+VAT+++::16'MOA+176:27200'UST+1+3'UST+2+9'USR+1:BA0
C636DC80F35A6DBB7927C201FDDAEACE6C32F21198BFB766D3A8778C
AF435F600A6B14C51324601ABE19D58E4D54BDD2CC15C96FE7D88811B9
E74A343E7527BB60BF45DB66486139F193EFA5684A287FA977CF970C5B6
A4EFE533EE2AC7A0C19CE010C777776D3C12F79F783806D7F6A6D62FF2
AA15EE8CFB7C9CDF934C94'UNT+59+20'
```

Because the user requested a new timestamp under linking protocol scheme, the resulting timestamp has two segment groups 1, 2 and n in order to encapsulate the backward link as specified on previous [PKITS D5]. These additional segment groups are underlined on the previous pretty-print timestamp.

As we have done with the timestamp request in the previous chapter, we detail the most important fields of the resulting timestamp:

**USH+10+2+1+4+2+2++1::EDITSA:::TSAEDI2++9+1:19981223:100213:** This is the USH segment of the the segment group 1. It has essential information:

- 10 Specified that this USH is related to time-stamping service information
- 4 It indicates that the timestamp was issued in linking protocol scheme EDITSA and TSAEDI2: Identifies PKITS EDITimeStamping Server
- 9 Timestamp serial number
- 19981223 is the stamped date. 23th December 1998-12-23
- 100213 is the stamped UTC time. 10 hours 02 minutes 13 seconds (UTC time)

**USA+1:::16'** This USA segment indicates that SHA-1 is the hash algorithm used to compute RSA signature.

**USC+82+3:KEYSERVER1TS:1111111111111:::SERVER1TS+4:CAKEY1:1234567891234:::CANAME1+3+2+2+1+++++2:20041217:181213+3:20041217:181213+4:20051217:181213'USA+3:0:1:10:1'USA+4:0:1:16:1'USA+6:0:1:10:1+12:BFC762E1E0C4AABE3577EB5D454411EB2F2FDAB652B2DD211E7203B0343379D2C2B5F989819CA61A6FB11A6A705F4F7995BD738C7404905CFD48F267BC3DEF82850EBB07C0815735F5C4A16C9BDAE9E27DC31C274D141675A6C3686137BC5645C5C9E44CC4F3D6E0650E52F75CC77DE110AA2DF05EE7313CFDA9F944D0A354D9+13:03+14:1024'USR+1:24FF95371B9F865C48E3B38E6DFB1CB69E2684DC00184413E67D9FA84D8E12824FFB80971A8F2404D870B4F8D0FDFF04A20369F7D8FD0DDC37F4F8FCF3A57C1D1081B3172FF2A7085F223F552EE1546F96B896BDDC8BC1FA43B6AD7ABB0A7F6A92C9BA8C0238E390EFCF15840099ACBD779628E8BE111075FD56AF3884DEDDF1:** This is the segment group 2. It encapsulates the PKITS EDITimeStamping Server EDIFACT certificate.

**USH+10+1+1+4+2+2++1::EDITSA:::TSAEDI2++8+1:19981223:095815':** This is the additional USH that encapsulates the linking information. It is the USH segment group 1 of the previous timestamp issued by the PKITS EDITimeStamping Server under linking protocol scheme. This is:

- 10 Timestamping related security header
- 4 Linking information
- EDITSA and TSAEDI2: Identifies PKITS EDITimeStamping Server
- 8 Serial number of the previous timestamp in the chain
- 19981223 the previous timestamp was issued on 23th December 1998-12-23
- 095815 the previous timestamp was issued at 09 hours, 58 minutes 15 seconds (UTC time)

**UST+1+3'** This is the security trailer related to the additional segment group 1 and 2. In the resulting prototype, there is no security result (on the [PKITS D5] que specified that it shall have the same value of the security result of the previous timestamp)

**UST+2+9'USR+1:BA0C636DC80F35A6DBB7927C201FDDAEACE6C32F21198BFB766D3A8778CAF435F600A6B14C51324601ABE19D58E4D54BDD2CC15C96FE7D88811B9E74A343E7527BB60BF45DB66486139F193EFA5684A287FA977CF970C5B6A4EFE533EE2AC7A0C19CE010C777776D3C12F79F783806D7F6A6D62FF2AA15EE8CFB7C9CDF934C94:** Security trailer and the security result of the resulting timestamp. It encapsulates the RSA digital digital signature the PKITS EDITimeStampingServer applied on the resulting timestamp token

## 5. APPENDIX B: BIBLIOGRAPHY

- [EDIINT-1] EDIINT: "Propuesta de implantación de servicios de seguridad a mensajes EDIFACT". March 98
- [PKITS D3] Architecture of Time-Stamping Service and Scenarios of Use: Service and Features, Deliverable D3 of ETS Project 23.192 Public Key Infrastructure with Time-Stamping Authority, May, 1998.
- [PKITS D5] Time-Stamping Service Functional Specification and Protocols for Structured Data: EDI documents, Deliverable D5 of ETS Project 23.192 Public Key Infrastructure with Time-Stamping Authority, August, 1998.
- [RFC 1305] D.L. Mills, "1305 Network Time Protocol (Version 3) Specification, March 1992.  
<Ftp://ds.internic.net/rfc/rfc1305.txt>
- [RFC 1319] B. Kaliski, "The MD2 Message-Digest Algorithm", Apr. 1992.  
<Ftp://ds.internic.net/rfc/rfc1319.txt>
- [RFC 1321] R. Rivest, "The MD5 Message-Digest Algorithm", Apr. 1992.  
<Ftp://ds.internic.net/rfc/rfc1321.txt>
- [RFC 781] Su, Z. A specification of the Internet protocol (IP) timestamp option. DARPA Network Working Group Report RFC-781. SRI International, May 1981.  
<Ftp://ds.internic.net/rfc/rfc781.txt>
- [SJWG 1] SJWG: "EDIFACT CD-9735-1: Application level syntax rules. Part 1: Syntax rules common to all parts, together with syntax service directories for each of the parts." 1997.
- [SJWG 2] SJWG: "EDIFACT CD-9735-2: Application level syntax rules. Part2: Syntax rules specific to batch EDI security rules for batch EDI." 1997.
- [SJWG 3] SJWG: "EDIFACT CD-9735-5: Application level syntax rules: Security rules for batch EDI. Part 5: (authenticity, integrity and non-repudiation of origin)". 1997.
- [SJWG 4] SJWG: "EDIFACT CD-9735-6. Application level syntax rules: Part 6: Secure authentication and acknowledgement message (message type AUTACK) ". 1997.
- [SJWG 5] SJWG: "Recommendations for UN/EDIFACT message level security from the UN/EDIFACT Security JWG" 1993.
- [SJWG 6] SJWG: "TRADE/WP4/R1026: EDIFACT SECURITY IMPLEMENTATION GUIDELINES". 1994.
- [SJWG 7] SJWG: "EDIFACT CD-9735-9. Application level syntax rules: Part 9: Security key and certificate management message (message type KEYMAN) ". 1997.