Περιγραφή του API για την ανάπτυξη και Ολοκλήρωση WEB Based Εφαρμογών με την υποδομή PKI και τις Υπηρεσίες CA του ΧΑ

Έκδοση 1.1

(Microsoft Windows Environment)

Οδηγίες Ανάπτυξης Εφαρμογών

Αθήνα, Μαίος 2003

1 SmartTrust Personal – Web Signer

1.1 Εισαγωγή

Ο SmartTrust Personal είναι ένα προϊόν που φέρνει την ασφάλεια, την χρήση smart cards καθώς και ικανότητα χρήσης ψηφιακών πιστοποιητικών σε “standard” Internet software.

Το ποιο σημαντικό component του Personal είναι ο WEB Signer. Ο WebSigner είναι ένα browser add-on που επιτρέπει τη χρήση Digital Signature σε απλό κείμενο καθώς και σε αρχεία. Μπορεί δε να χρησιμοποιηθεί τόσο με Internet Explorer όσο και με Netscape Navigator.

1.2 Υπογραφή απλού κειμένου

1.2.1 Παραδείγματα ενεργοποίησης

Ο WebSigner θα ενεργοποιηθεί μέσω παραμέτρων σε μία σελίδα HTML. Το μέρος της υπογραφής απλού κειμένου του WebSigner θα «τρέξει» σαν embedded plug- in ή ActiveX όταν ενεργοποιηθεί από τον τύπο MIME «text/ x- text- to- sign». Οι παράμετροι μπορούν να περάσουν στο HTML χρησιμοποιώντας τα <EMBED> ή <OBJECT> tags.

Προσοχή: Όλες οι παράμετροι είναι “case sensitive”

Παράδειγμα ενεργοποίησης του plugin για υπογραφή απλού κειμένου:

```
<EMBED
src=" DataToSign. sgn"
Height=" 100" Width=" 100"
TYPE=" text/ x- text- to- sign"
WSXName=" MySignedData"
WSXAction=" http:// www. SmartTrustTech. com/ cgi/ scrpt1.exe"
WSXFormat=" PKCS7SIGNED"
WSXView=" ShowData">
```

Παράδειγμα ενεργοποίησης του ActiveX για υπογραφή απλού κειμένου:

```
<OBJECT CLASSID=" CLSID: 8AC8A833- 2F0F- 11D5- 845D- 0050DA2DEE56"
HEIGHT=" 100" WIDTH=" 100">
<PARAM NAME=" SRC" VALUE=" DataToSign. sgn">
<PARAM NAME=" TYPE" VALUE=" text/ x- text- to- sign">
<PARAM NAME=" WSXName" VALUE=" MySignedData">
<PARAM NAME=" WSXAction" VALUE=" http:// www. SmartTrustTech. com/
cgi/ scrpt1.exe">
```

```
<PARAM NAME=" WSXFormat" VALUE=" PKCS7SIGNED">
<PARAM NAME=" WSXView" VALUE=" ShowData">
</ OBJECT>
```

Παράδειγμα για την ενεργοποίηση είτε του plugin είτε του ActiveX:

```
<script language=" VBScript">

        Function ControlExists( objectID)
                on error resume next
                ControlExists = IsObject( CreateObject( objectID))
        End Function

</ script>

<script Language=" JavaScript">
        var plugin
        var explorer

        // Check Browser Type
        if( navigator. appName. indexOf(" Explorer")== - 1) {
                // Browser is not Explorer
                explorer = false
                plugin = navigator. mimeTypes[" text/ x- text- to- sign"] }
        else {
                // Browser is Explorer
                explorer = true
                plugin = !ControlExists(" AWebSigner. WebSignerCtl")
        }

        if (plugin) {
                document. writeln("< EMBED SRC= 'DataToSign. sgn'")
                document. writeln(" TYPE= text/ x- text- to- sign")
                document. writeln(" Height= '100' Width= '100'")
                document. writeln(" WSXName= 'MySignedData'")
                document. writeln(" WSXAction= 'http:// www. SmartTrustTech. com/
                cgi/ scrpt1. exe'")
                document. writeln(" WSXFormat= 'PKCS7SIGNED'")
                document. writeln(" WSXView= 'ShowData'>") }
                else if (explorer) {
                document. writeln("< OBJECT CLASSID= 'CLSID: 8AC8A833-
                2F0F- 11D5- 845D- 0050DA2DEE56'")
                document. writeln(" HEIGHT= '100' WIDTH= '100'>")
                document. writeln("< PARAM NAME= 'SRC' VALUE= 'DataToSign.
                sgn'>")
                document. writeln("< PARAM NAME= 'TYPE' VALUE= 'text/ x-
                text- to- sign'>")
                document. writeln("< PARAM NAME= 'WSXName' VALUE=
                'MySignedData'>")
```

```
            document. writeln("< PARAM NAME= 'WSXAction' VALUE=
            'http:// www. SmartTrustTech. com/ cgi/ scrpt1. exe'>")
            document. writeln("< PARAM NAME= 'WSXFormat' VALUE=
            'PKCS7SIGNED'>")
            document. writeln("< PARAM NAME= 'WSXView' VALUE=
            'ShowData'>")
            document. writeln("</ OBJECT>") }
      else {
            alert(" Please install Websigner for this browser")
      }

</ script>
```

1.2.2 Περιγραφή Παραμέτρων

SRC: Καθορίζει το τρέχων αρχείο που περιέχει τα δεδομένα προς υπογραφή. Τα δεδομένα πρέπει να σταλούν ως 'text/ x- text- to- sign' από τον server. Μπορεί να χρησιμοποιηθεί είτε η παράμετρο **SRC** είτε η **WSXDataToSign**. Εάν και οι δύο παράμετροι χρησιμοποιούνται τότε χρησιμοποιείται πάντα η **WSXDataToSign**

TYPE: Η παράμετρο αυτή πρέπει να έχει πάντα τιμή 'text/ x- text- to- sign'. (*Υποχρεωτική*)

Height and Width: Ορίζουν το μέγεθος του WebSigner μέσα στον browser. (*Υποχρεωτική*)

WSXName: Ορίζει το όνομα του HTML form field που θα περιέχει τα υπογεγραμμένα δεδομένα που θα σταλούν από τον WebSigner. Η WSXName είναι προαιρετική παράμετρο. Εάν δεν χρησιμοποιηθεί τότε παίρνει την default τιμή “SignedData”.

WSXAction: Ορίζει το URL στο οποίο ο WebSigner θα στείλει τα υπογεγραμμένα δεδομένα. (*Υποχρεωτική*)

WSXFormat: (*Προαιρετική*) Ορίζει τη μορφή των υπογεγραμμένων δεδομένων. Αυτή την στιγμή υποστηρίζεται μόνο ο τύπος PKCS #7 υπογεγραμμένων δεδομένων. Μετά τη δημιουργία της, η υπογραφή θα «μορφοποιηθεί με URL-encode για να σταλεί σαν στοιχείο μιας WEB form. Η τιμή “PKCS7SIGNED” ορίζει αυτή ακριβώς τη μορφή. Προαιρετικά, τα δεδομένα μπορούν να περιέχονται στην υπογραφή, χρησιμοποιώντας την τιμή "PKCS7SIGNED_ Attached" (ίδιο με το «PKCS7SIGNED»), ή να μην περιέχονται, χρησιμοποιώντας την τιμή "PKCS7SIGNED_ Detached". Η παράμετρος αυτή είναι προαιρετική και έαν παραληφθεί χρησιμοποιείται το “PKCS7SIGNED_ Detached”.

WSXView: (*Προαιρετική*) Ορίζει ποιο από τα 2 πιθανά signature dialog boxes θα εμφανιστουν για την υπογραφή. Η τιμή “HideData” έχει σαν αποτέλεσμα ο

WebSigner να ενεργοποιήσει ένα μικρό dialog box όπου τα προς υπογραφή δεδομένα μπορούν να «ειδωθούν» σε ένα άλλο ξεχωριστό dialog box. Η παράμετρος αυτή είναι προαιρετική και η default τιμή είναι “ShowData”.

WSXDataToSign Η τιμή της παραμέτρου πρέπει να περιέχει τα προς υπογραφή δεδομένα. Για να μπορέσει ο χαρακτήρας newline να δουλέψει σωστά, τα προς υπογραφή δεδομένα πρέπει να είναι URL-encoded. Ο WebSigner θα αποδικοποιήσει το μύνημα, θα το εμφανίσει στο παράθυρο της υπογραφής και θα υπογράψει το αποκωδικοποιημένο μύνημα. Μπορεί να χρησιμοποιηθεί είτε η παράμετρο **SRC** είτε η **WSXDataToSign**. Εάν και οι δύο παράμετροι χρησιμοποιούνται τότε χρησιμοποιείται πάντα η **WSXDataToSign**

WSXDataReturnName : (*Προαιρετική*) Ορίζει το όνομα του HTML form field που θα περιέχει τα ανυπόγραφα δεδομένα που θα σταλούν από τον WebSigner. Εάν παραληφθεί, τα ανυπόγραφα δεδομένα δεν θα σταλούν.

WSXButtonName: (*Προαιρετική*) Η παράμετρος αυτή μπορεί να χρησιμοποιηθεί έτσι ώστε ο WebSigner να εμφανίσει ένα συγκεκριμένο μύνημα αντί για το WebSigner logo. Παράδειγμα: “Πιέστε για υπογραφή...”. Το μέγεθος του button του WebSigner αυξομειώνεται έτσι ώστε να χωρέσει το κείμενο της παραμέτρου.

WSXWindow: (*Προαιρετική*) Ορίζει το όνομα του «παράθυρου» ή του frame στο οποίο θα εμφανιστεί η «απόκριση» του web server μετά το HTML post του WebSigner.. Εάν παραληφθεί , χρησιμοποιείται το παράθυρο, (ή frame), στο οποίο ενεργοποιήθηκε και ο WebSigner, πχ., “_ self”.
Note: Η παράμετρο αυτή είναι υποχρεωτική για browser Netscape Communicator 4. x και MS Internet Explorer 4. x.

WSXBase64 (*Προαιρετική*) Ορίζει εάν η παραγόμενη υπογραφή θα είναι base64 encoded πριν γίνει URL- encoded, (τιμή για αυτή την περίπτωση “YES”.) Εάν η παράμετρος παραληφθεί τότε λαμβάνεται η default τιμή “NO”.

WSXCustomBitmapURL: (*Προαιρετική*) Παίρνει σαν τιμή ένα URL που ορίζει ένα Windows bitmap αρχείο εικόνας, (. bmp). Η εικόνα αυτή θα εμφανιστεί στο πάνω μέρος του παραθύρου υπογραφής του WebSigner. Το μέγιστο μέγεθος της εικόνα είναι:πλάτος 390 pixels και ύψος 50 pixels.

WSXIncludeCSSD: Χρησιμοποιείται μόνο με την Oberthur GalactIC smart card. Δυνατές τιμές είναι οι "true" και "false" (default = false). Στέλνει μια νέα μεταβλητή "CSSD" στην απάντηση. Η τιμή της είναι URL encoded CSSD από την smart card. Η τιμή της είναι base64 encoded εάν η παράμετρος WSXBase64 έχει την τιμή “true”.

WSXIncludeCaCert: Δυνατές τιμές είναι οι "true" και "false" (default = false). Σε περίπτωση που έχει την τιμή "true" θα συμπεριληφθεί στην υπογραφή και το CA πιστοποιητικό που βρίσκεται στην κάρτα (εάν αυτό είναι διαθέσιμο).

1.2.3 Χρήση του ActiveX control

Ο WebSigner μπορεί να χρησιμοποιηθεί και με τη χρήση ActiveX. Σε αυτή την περίπτωση πέρα από τις παραμέτρους που μπορούν να χρησιμοποιηθούν στη δήλωση του ActiveX μπορούν να χρησιμοποιηθούν και 3 μέθοδοι του control:

- SetBuffer : Θέτει τα δεδομένα προς υπογραφή (όπως η παράμετρο **WSXDataToSign**)
- SetAction : Θέτει το URL στο οποίο θα σταλούν τα υπογεγραμμένα δεδομένα (όπως η παράμετρο **WSXAction**)
- Sign() : εμφανίζει το παράθυρο υπογραφής δεδομένων ή Sign(signBuffer as string) όπως παραπάνω αλλά θέτοντας απ' ευθείας τα προς υπογραφή δεδομένα (χωρίς χρήση της SetBuffer)

1.2.3.1 Παράδειγμα

```
<html>
<head>
<title>Smart Trust WebSigner invoked using activeX</title>
<OBJECT CLASSID=CLSID:8AC8A833-2F0F-11D5-845D-0050DA2DEE56
id="web_signer_button" style="LEFT: 0px; TOP: 0px" VIEWASTEXT>
        <PARAM NAME="TYPE" VALUE="text/x-text-to-sign">
        <PARAM NAME="WSXDataReturnName" VALUE="TheUnsignedData">
        <PARAM NAME="WSXName" VALUE="TheSignedData">
        <PARAM NAME="Hidden" VALUE=false>
        <PARAM NAME="WSXBase64" VALUE="YES">
        <PARAM NAME="WSXAction" VALUE="sendback_integration_2.asp">
        <PARAM NAME="WSXCustomBitmapURL"
        VALUE="http://webasyk/SmartTrustDemo/images/setup_visionSmall.BMP">
</object>
<SCRIPT ID=clientEventHandlersVBS LANGUAGE=vbscript>
<!--
Sub button1_onclick
        button1.disabled = true
        web_signer_button.SetAction("sendback_integration_2.asp")
        web_signer_button.SetBuffer("this are the data to be signed")
        web_signer_button.Sign
        button1.disabled = false
End Sub
-->
</SCRIPT>
</head>
<body bgcolor=#D8DFF1>
<center>
<INPUT type="button" value="Button" id=button1 name=button1>
</center></body></html>
```

1.2.4 Java Scripting

Ο WebSigner μπορεί να ενεργοποιηθεί και με χρήση JavaScript βασισμένη στην τεχνολογία LiveConnect (μόνο σε Netscape browsers). Η τεχνολογία LiveConnect φέρνει μαζί plug- ins, Java applets, και JavaScript προγράμματα και παρέχει επικοινωνία μεταξύ των 3 παραπάνω software modules επιτρέποντας άμεση πρόσβαση σε μεθόδους τους.

1.2.4.1 Attributes

Το αντικείμενο υπογραφής δεδομένων έχει 2 attributes:

- DataBuffer Περιέχει τα προς υπογραφή δεδομένα.
- SignatureBuffer περιέχει την υπογραφή

Note: Τα δεδομένα των παραπάνω attributes χρησιμοποιούνται εσωτερικά στο αντικείμενο και δεν μπορείτε να τα χρησιμποποιήσετε παρά μόνο με τις παρακάτω μεθόδους.

1.2.4.2 Methods

- void SetBuffer(String signBuffer) Θέτει τα δεδομένα προς υπογραφή (όπως η παράμετρο **WSXDataToSign**)
- int Sign()εμφανίζει το παράθυρο υπογραφής δεδομένων

- int Sign(String signBuffer) Οι 2 παραπάνω methods συνδιασμένες. Σημείωση: Σε browser Netscape 6.1, η μέθοδος ονομάζεται SignBuffer

- void SetAction(String szAction) Θέτει το URL στο οποίο θα σταλούν τα υπογεγραμμένα δεδομένα (όπως η παράμετρο **WSXAction**)

1.2.4.3 Παράδειγμα

```
<HTML>
<META HTTP- EQUIV=" Cache- Control" CONTENT=" no cache">
<META HTTP- EQUIV=" Pragma" CONTENT=" no cache"> <center> <h1> Testa script</ h1>

<embed
src=" DataToSign.sgn"
type=" text/ x- text- to- sign"
Width= 120 Height= 50
Name="Signer"
WSXName="MySignedData"
WSXAction="http:// www.SmartTrusttech.com/scripts/SmartTrustscr1.exe"
WSXView=" HideData">
```

```
<script>
        function doSignature( theBuffer) {
                document.Signer.SetBuffer(theBuffer);
                document.Signer.Sign();
        }
</ script>

<form name= form1> Native test put data here:
        <input type= text name= DataToSign size= 50>
        <input type= button value=" Click to test"
        onclick= 'doSignature( document. form1.DataToSign. value) '>
</ form>

</ center> </ HTML>
```

1.3 Περιβάλλον ανάπτυξης και χρήσης

Λειτουργικό Σύστημα: MS Windows 95, 98, ME, NT 4.0, 2000, XP
Πρόγραμμα Πλοήγησης: MS Inernet Explorer 4.X και άνω, Netscape Communicator 4.X και άνω.

2 MSCAPICOMSigner.OCX

2.1 Εισαγωγή

Το MSCAPICOMSigner.OCX αποτελεί wrapper πάνω στο CAPICOM σε μορφή ACTIVEX control. Αναπτύχθηκε από την ΑΣΥΚ και παρέχει ένα πολύ απλοϊκό API για τη δημιουργία ψηφιακών υπογραφών. Χρησιμοποιείται για τη δημιουργία ψηφιακών υπογραφών πάνω σε απλό κείμενο. Το MSCAPICOMSigner.OCX προϋποθέτει την ύπαρξη εγκατεστημένου PERSONAL γαι τη δημιουργία υπογραφών απο πιστοποιητικά που βρίσκονται σε smart cards.

2.2 Programming Interface

2.2.1 Methods

UseOnlyHERMESCertificates
Visibility:public
Return Type Expression: Boolean
Informs the control to use only HERMES Certificates or not. Returns always true.
eg
Dim ret as boolean
ret = MSCAPICOMSignerV2.UseOnlyHERMESCertificates(True)

Parameters
aValue
Type Expression: Boolean
Kind: in
DefaultValue: False
True - means use only HERMES Certificates.
False - means use all valid certificates

ViewData
Visibility: public
Return Type Expression: Boolean
Informs the control to show tha data before the signature creation or not.Returns always true
eg
Dim ret as boolean
ret = MSCAPICOMSignerV2.ViewData(True)

Parameters
aValue
Type Expression: Boolean
Kind: in
True - means show data before signature creation False - means do not show data.

Sign
Visibility: public
Return Type Expression: Byte
Signs the given data.
Returns 1 in case of success, 0 in case of failure
eg
Dim ret as byte
ret = MSCAPICOMSignerV2.Sign("data_to_be_signed")

Parameters
UnsignedData
Type Expression: String
Kind: in
contains the clear data that will be signed

2.2.2 Properties

SignedData
Visibility: public
Return Type Expression: String
PROPERTY (READ ONLY).
Returns a string that contains the signature.
eg
dim s as string
s = MSCAPICOMSignerV2.SignedData

ErrMessage
Visibility: public
Return Type Expression: String
PROPERTY (READ ONLY)
Returns a string that contains the error that occured durnig the signature creation.
eg
dim s as string
s = MSCAPICOMSignerV2.ErrMessage

OCXVersion
Visibility:public
Return Type Expression:String
PROPERTY (READ ONLY)
Returns a string that contains version of the control
eg
dim s as string
s = MSCAPICOMSignerV2.OCXVersion

2.3 Παράδειγμα (visual basic)

```
Dim ret As Boolean
Dim ClearData as string

ClearData = “Some Data To Sign”

‘c is the name of the MSCAPICOMSIGNER.OCX CONTROL.

Msgbox c.OCXVersion  ‘inrmation only

‘use only HERME certificates
Ret = c.UseOnlyHERMESCertificates (True)
‘View data before signing
Ret = c.ViewData(True)

If c.Sign(ClearData) = 1 Then
        Msgbox c.SignedData
Else
        MsgBox c.ErrMessage
End If
```

3 SmartTrust Servant – AsykCVPCom.dll (Έκδοση 1.1)

3.1 Γενικά

Το SECURITY CENTER είναι μια εφαρμογή μέσω της οποίας μπορεί να ελεγχθεί η εγκυρότητα ενός πιστοποιητικού ή και μιας ψηφιακής υπογραφής.

Μια τρίτη εφαρμογή για να «μιλήσει» με το SECURITY CENTER πρέπει να χρησιμοποιήσει το CVP (Certificate Verification Protocol). Το CVP είναι ένα εδικού τύπου text-based TCP/ IP πρωτόκολο.

Με το CVP η τρίτη εφαρμογή (client) μπορεί να στείλει ερωτήματα και να λάβει απαντήσεις από τον Servant σχετικά με την εγκυρότητα ψηφιακών πιστοποιητικών και υπογραφών.

Η εφαρμογή client μπορεί να ζητήσει οποιόδηποτε συνδιασμό από τους παρακάτω ελέγχους για ένα πιστοποιητικό:

- Time validity
- Signature verification
- Certificate revocation

Οι έλεγχοι πραγματοποιούνται με την παραπάνω σειρά. Η διεργασία ελέγχου σταματάει μόλις υπάρξει το πρώτο αρνητικό αποτέλεσμα ή όταν ολοκληρωθεί.

Time valid?	**Signature Valid?**	**CRL?**	**Result**
ΝΑΙ	ΝΑΙ	ΟΧΙ	ΕΓΚΥΡΟ
ΝΑΙ	ΝΑΙ	ΝΑΙ	REVOKED
ΝΑΙ	ΟΧΙ	(--)	ΑΚΥΡΟ
ΟΧΙ	(--)	(--)	ΛΗΓΜΕΝΟ

Εάν δεν ζητηθεί έλεγχος time validity check, τότε ο χρόνος εγκυρότητας λογίζεται ως έγκυρος. Εάν δεν ζητηθεί signature verification, τότε η υπογραφή λογίζεται ως έγκυρη. Εάν δεν ζητηθεί έλεγχος για το εάν το πιστοποιητικό δεν είναι άκυρο (δεν ανήκει σε CRL – Certificate Revocation List), τότε πιστοποιητικό θεωρείται ότι δεν έχει ακυρωθεί.

Note: Εάν δεν ζητηθεί κανένας έλεγχος τότε το αποτέλεσμα θα είναι πάντα θετικό.

3.2 AsykCVPCom.dll

Για την επικοινωνία με τον SERVANT δημιουργήθηκε ένα ActiveX dll που υλοποιεί το CVP πρωτόκολο. Το dll υλοποιήθηκε με χρήση MS J++ (από τη SMART TRUST), παρέχει δε ένα πιο φιλικό interface (methods και properties) για επικοινωνία με τον SERVANT.

3.2.1 Properties

Cert: : (Read Only - string) Επιστρέφει το περιεχόμενο του ψηφιακού πιστοποιητικού
CertIssuer :(Read Only - string) Επιστρέφει τον εκδότη του ψηφιακού πιστοποιητικού
CertResult : (Read Only - string) Επιστρέφει το αποτέλεσμα του ελέγχου του ψηφιακού πιστοποιητικού (NNN χχχχχχ πχ 200 OK)
CertSerialNumber : (Read Only - string) Επιστρέφει το serial number του ψηφιακού πιστοποιητικού
CertSubjAltName : (Read Only - string) Επιστρέφει το Alternate Name του ψηφιακού πιστοποιητικού
CertSubject : (Read Only - string) Επιστρέφει τον «Κύριο» του ψηφιακού πιστοποιητικού
CertSubjectSerial : (Read Only - string) Επιστρέφει το Subject Serial πεδίο του ψηφιακού πιστοποιητικού
CertValidFrom : (Read Only - string) Επιστρέφει την εναρκτήρια ημερομηνία ισχύς του ψηφιακού πιστοποιητικού (σε μορφή πχ 2002-10-26 10:21:24 GMT)
CertValidTo : (Read Only - string) Επιστρέφει την καταλυτική ημερομηνία ισχύς του ψηφιακού πιστοποιητικού (σε μορφή πχ 2002-10-26 10:21:24 GMT)
Message : (Read Only - string) Επιστρέφει τα «καθαρά» δεδομένα από την ψηφιακή υπογραφή.
pkcs7Result : (Read Only - string) Επιστρέφει το αποτέλεσμα του ελέγχου του ψηφιακού πιστοποιητικού στη μορφή (NNN χχχχχχ πχ 200 OK)

3.2.2 Methods

CollectCertData(Certificate as string): (Boolean) Δέχεται σαν παράμετρο τα δεδομένα ενός ψηφιακού πιστοποιητικού - **Certificate as string** - για να μπορέσουν να εξαχθούν από αυτό τα επιμέρους πεδία του καθώς και να εξακριβωθεί η εγκυρότητα του. Επιστρέφει true σε περίπτωση που η κλήση της ήταν επιτυχής, false σε αντίθετη περίπτωση.
collectPkcs7Data (TheSignedData as string): (Boolean) Δέχεται σαν παράμετρο τα δεδομένα μιας ψηφιακής υπογραφής - **TheSignedData as string** – για να μπορέσουν να εξαχθούν από αυτά το ψηφιακό πιστοποιητικό, τα καθαρά (ανυπόγαγραφα) δεδομένα καθώς και να διαπιστωθεί η

εγκυρότητα της υπογραφής. Επιστρέφει true σε περίπτωση που η κλήση της ήταν επιτυχής, false σε αντίθετη περίπτωση.
getCert(): (String) Όπως και η property **Cert**
getCertIssuer(): (String) Όπως και η property **CertIssuer**
getCertResult(): (String) Όπως και η property **CertResult**
getCertSerialNumber(): (String) Όπως και η property **CertSerialNumber**
getCertSubjAltName(): (String) Όπως και η property **CertSubjAltName**
getCertSubject(): (String) Όπως και η property **CertSubject**
getCertSubjectSerial(): (String) Όπως και η property **CertSubjectSerial**
getCertValidFrom(): (String) Όπως και η property **CertValidFrom**
getCertValidTo(): (String) Όπως και η property **CertValidTo**
getMessage(): (String) Όπως και η property **Message**
getPkcs7Result: (String) Όπως και η property **pkcs7Result**
SetServartNetwork(ServantLocation as string, Port as integer): (Boolean) Δέχεται σαν παράμετρο το όνομα ή την IP διεύθυνση του υπολογιστή όπου «λειτουργεί» το SECURITY CENTER - **ServantLocation as string** - καθώς και την «πόρτα» την οποία ο Servant χρησιμοποιεί για επικοινωνία - **Port as integer**. Επιστρέφει true σε περίπτωση που η κλήση της ήταν επιτυχής, false σε αντίθετη περίπτωση.

GetMessageBytes(): Η μέθοδος αυτή επιστρέφει τα «καθαρά» δεδομένα από την ψηφιακή υπογραφή στην μορφή byte array. Κάθε θέση του array δηλαδή, περιέχει τον αντίστοιχο ascii κωδικό του κάθε χαρακτήρα του αρχικού string.
Πχ το μύνημα «1234αβγδ» θα επιστραφεί ως
Array(0) = 49 ‘ASCII ισοδύναμο του «1»
Array(1) = 50 ‘ASCII ισοδύναμο του «2»
Array(2) = 51 ‘ASCII ισοδύναμο του «3»
Array(3) = 52 ‘ASCII ισοδύναμο του «4»
Array(4) = 225 ‘ASCII ισοδύναμο του «α»
Array(5) = 226 ‘ASCII ισοδύναμο του «β»
Array(6) = 227 ‘ASCII ισοδύναμο του «γ»
Array(7) = 228 ‘ASCII ισοδύναμο του «δ»
Στη συνέχεια είναι ευθύνη της εφαρμογής που χρησιμοποιεί το object να μετασχηματίσει το array στο αρχικό μύνημα με κώδικα όπως τον παρακάτω :

```
Dim TheByteArray As Variant
Dim TheByteArrayLength As Long
Dim iCount As Long
Dim theClearData As String

TheByteArray = mycvpcom.getMessageBytes()
TheByteArrayLength = UBound(TheByteArray)
theClearData = String(TheByteArrayLength + 1, " ")

For iCount = 1 To l + 1
   Mid$(theClearData, i, 1) = Chr$(TheByteArray (i - 1))
Next iCount
```

Στη συνέχεια το μύνημα που δημιουργήθηκε μπορεί να συγκριθεί με τα αρχικά (καθαρά) δεδομενά που έιχαν υπογραφεί (είτε για λόγους debugging είτε για λόγους που απαιτεί η εφαρμογή) πχ

```
If theClearData = TheUnshignedData Then
    Msgbox “DATA retrieved from ASYCCvpCom.dll are identical with “ & _
            “the clear data”
 Else
    Msgbox “DATA retrieved from ASYCCvpCom.dll were erroneous!”
End If
```

compareMesssage(byteArr() as integer): Η μέθοδο αυτή δέχεται ένα integer array που περιέχει όλους τους ισοδύναμους ascii κωδικούς του καθαρού μυνήματος και επιστρέφει :
1 σε περίπτωση που το μύνημα που τροφοδότησαμε το dll με αυτόν τον τρόπο είναι ίδιο με εκείνο που το dll εξήγαγε από την υπογραφή (εννοείται πως θα πρέπει να έχει κληθεί πρώτα η **collectPkcs7Data).**
0 σε αντίθετη περίπτωση.

Παράδειγμα

```
Dim byteArr() As Integer
Dim TheUnsignedDataLength As Long
Dim i As Long
Dim aRet as integer

TheUnsignedDataLength = Len(TheUnshignedData)
ReDim byteArr(TheUnsignedDataLength - 1)

For i = 0 To TheUnsignedDataLength - 1
   byteArr(i) = Asc(Mid$(TheUnshignedData, i + 1, 1))
Next i

aRet  = mycvpcom.compareMesssage(byteArr)

If aRet  = 1 Then
 Msgbox “DATA retrieved from SERVANT are identical with the clear data”
Else
  Msgbox “DATA retrieved from SERVANT were erroneous!”
End If
```

3.2.3 Κωδικοί Απόκρισης Ελέγχων SECURITY CENTER

Ο κωδικός απόκρισης είναι ένας αριθμός integer για την προσπάθεια (απο τον Servant) να καταλάβει και να ικανοποιήσει ένα request. Η περιγραφή (του) έχει σκοπό να δώσει μια σύντομη εξήγηση για τον κωδικό.

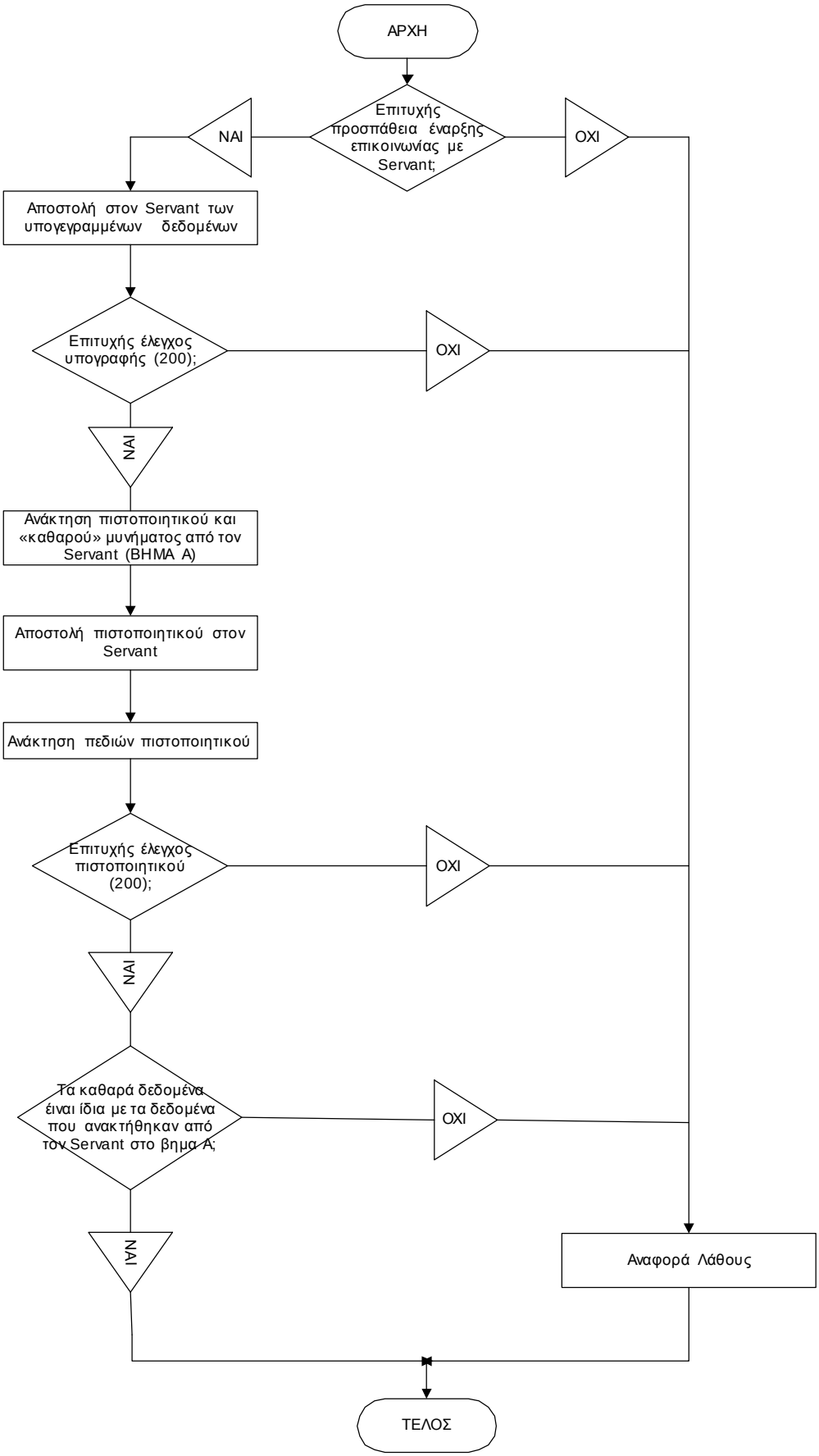
Το πρώτο ψηφίο του κωδικού ορίζει την κατηγορία της απόκρισης: 2xx Success. Η αίτηση παραλήφθηκε επιτυχώς, κατανοήθηκε και εξυπηρετήθηκε. 4xx Client error. Η αίτηση περιείχε συντακτικά λάθη ή για κάποιο λόγο δεν στάθηκε δυνατό να εξυπηρετηθεί. 5xx Server error. Ο εξυπηρετητής (Servant) δεν μπόρεσε για κάποιο λόγο να εξυπηρετήσει την αίτηση.

Success (2xx)		
200	**OK**	Η αίτηση έγινε αποδεκτή και έγιναν όλοι οι έλεγχοι επιτυχώς
201	**REVOKED**	Το πιστοποιητικό έχει ακυρωθεί (ανήκει σε CRL)
202	**INVALID**	Η ψηφιακή υπογραφή δεν είναι έγκυρη
203	**EXPIRED**	Η χρονική περίοδος εγκυρότητας του πιστοποιητικού έχει λήξει
211	**INVALID DATA**	Τα δεδομένα PKCS#7 δεν στάθηκε δυνατό να ελεγχθούν

Client Error (4xx)		
400	**Bad Request**	Η αίτηση δεν έχει τη σωστή σύνταξη
401	**Issuer Certificate Not Found**	Ο Servant δεν γνωρίζει το πιστοποιητικό του CA
402	**Certificate does not parse**	Το πιστοποιητικό δεν έχει την σωστή κωδικοποίηση

Server Error (5xx)		
500	**Server Error**	Μη αναμένομενο λάθος δεν επιτρέπει την εξυπήρετηση της αίτησης
501	**Stale revocation data**	Ο χρόνος για το επόμενο CRL έχει περάσει (άρα είναι αδύνατος ο έλεχγος για την εγκυρότητα ενός πιστοποιητικού)
502	**No CRL found for this issuer**	Δεν υπάρχουν CRLs για τον CA-εκδότη του πιστοποιητικού.

3.2.4 Λογικό Διάγραμμα Ελέγχων



3.2.5 Παράδειγμα χρήσης του AsykCVPCom.dll (Visual Basic)

Για να είναι δυνατή η χρήση του παραπάνω μέσα από Visual Basic θα πρέπει να δηλωθεί ως Reference σε ένα Project η βιβλιοθήκη AsykCVPCom. (Μενού Project **à** References)

Η παρακάτω Function δέχεται ως παραμέτρους τα ανυπόγραφα «καθαρά» δεδομένα (TheUnsignedData) καθώς και την ψηφιακή υπογραφή που προήλθε από αυτά (TheSignedData).

```
Public Function CheckPKIData(Byval TheUnsignedData as string, _
                             Byval TheSignedData as string)  As Boolean

  Const OK_RESULT = 200

  Dim b As Boolean
  Dim mycvpcom As New asykCvpCom.CvpComV2
  Dim Result As String
  Dim cert As String
  Dim message As String
  Dim certSubject As String
  Dim certIssuer As String
  Dim certSerialNumber As String
  Dim certValidFrom As String
  Dim certValidTo As String
  Dim certSubjAltName As String
  Dim subjectSerialNumber As String
  Dim ServantLocation As String

 On Error GoTo ErrHandler
 CheckPKIData = False
  If Not mycvpcom.SetServartNetwork("A_PC_NAME_OR_IP", 1234) Then
      Set mycvpcom = Nothing
      Debug.Print "ΠΡΟΒΛΗΜΑ ΣΤΗΝ ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΟΝ SERVANT"
      Exit Function
 End If

 b = mycvpcom.collectPkcs7Data(TheSignedData)
 If Not b Then
      'Network problem
      Debug.Print "ΠΡΟΒΛΗΜΑ ΣΤΗΝ ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΟΝ SERVANT"
  Else
      'Signature "quality" check
      Result = mycvpcom.getPkcs7Result()
      If Val(Left$(Result,3)) <> OK_RESULT Then
          'signature is invalid
          Debug.Print "ΠΡΟΒΛΗΜΑ ΣΤΗ ΔΙΑΔΙΚΑΣΙΑ ΕΛΕΓΧΟΥ" & _
                      " ΥΠΟΓΡΑΦΗΣ. ΑΠΟΤΕΛΕΣΜΑ=" & Result
```

```
        Else
            'retreive the certificate and the message from the signature
            cert = mycvpcom.getCert()
            message = mycvpcom.getMessage()
            b = mycvpcom.collectCertData(cert)
            If Not b Then
                'Network problem
                Debug.Print “ΠΡΟΒΛΗΜΑ ΣΤΗΝ ΕΠΙΚΟΙΝΩΝΙΑ ” & _
                “ME TON SERVANT”
            Else
                'get certificate data
                certSubject = mycvpcom.getCertSubject()
                certIssuer = mycvpcom.getCertIssuer()
                certSerialNumber = mycvpcom.getCertSerialNumber()
                certValidFrom = mycvpcom.getCertValidFrom()
                certValidTo = mycvpcom.getCertValidTo()
                certSubjAltName = mycvpcom.getCertSubjAltName()
                subjectSerialNumber = mycvpcom.getCertSubjectSerial()
                'certificate quality check
                Result = mycvpcom.getCertResult()
                Select Case val(Left$(Result, 3))
                    Case OK_RESULT 'all OK
                        'check if message extracted from signed message is identical
                        ‘with the original data
                        If message <> TheUnshignedData Then
                            'if execution ends up here means that somebody change
                            ‘the data during the net-tranfer
                            Debug.Print “ΠΡΟΒΛΗΜΑ ΣΤΗ ΔΙΑΔΙΚΑΣΙΑ
                            “ΕΛΕΓΧΟΥ ΥΠΟΓΡΑΦΗΣ“ & _
                            "Subject = " & certSubject & vbCrLf & _
                            "SerialNumber=" & certSerialNumber & vbCrLf & _
                            "SubjectSerial=" & subjectSerialNumber & _
                            "ΑΠΟΤΕΛΕΣΜΑ=" & Result
                        Else
                            CheckPKIData = True
                        End If
                    Case Else
                        Debug.Print “ΠΡΟΒΛΗΜΑ ΣΤΗ ΔΙΑΔΙΚΑΣΙΑ” & _
                        “ΕΛΕΓΧΟΥ ΥΠΟΓΡΑΦΗΣ“ & _
                        "Subject = " & certSubject & vbCrLf & _
                        "SerialNumber=" & certSerialNumber & vbCrLf & _
                        "SubjectSerial=" & subjectSerialNumber & _
                        "ΑΠΟΤΕΛΕΣΜΑ=" & Result
                    End Select
            End If  'if collectCertResult = false
        End If  'if pkcs7Result <> 200 then
    End If  'If (collectPkcs7Result = False) Then

ErrHandler:
    If Err Then
```

```
    Debug.Print “ΠΡΟΒΛΗΜΑ ΣΤΗ ΔΙΑΔΙΚΑΣΙΑ ΕΛΕΓΧΟΥ “ & _
    Vbcrlf & “Error=” & error
    Err = 0
  End If
  Set mycvpcom = Nothing
  Set x = Nothing

End Function
```

3.2.6 Περιβάλλον Ανάπτυξης

Λειτουργικό Σύστημα: MS Windows NT 4.0, 2000
WEB Server: MS IIS 4, MS IIS 5
Πρόγραμμα Πλοήγησης: MS Inernet Explorer 4.X και άνω
Εργαλεία Ανάπτυξης: MS Visual Basic, VBScript ή οποιαδήποτε άλλη γλώσσα υποστηρίζει ActiveX dlls.