RSA-SecurID
ActiveX-Control
für Internet Explorer 7.0

Version 1.0

Content

1 Description 3
1.1 Purpose 3
1.2 Process flow 3
2 Installation and Embedding 4
2.1 Environment 4
2.2 Installation 4
2.3 Embedding 4
2.4 Calling the embedded ActiveX-Control 4
3 Functions 5
3.1 GetTokenID 5
3.1.1 Input parameters 5
3.1.2 Output parameters 5
3.1.3 Embedding example 5
3.2 ResetToken 6
3.2.1 Input parameters 6
3.2.2 Output parameters 6
3.2.3 Embedding example 6
3.3 ResetPIN 7
3.3.1 Input parameters 7
3.3.2 Output parameters 7
3.3.3 Embedding example 7
3.4 ChangePIN 8
3.4.1 Input parameters 8
3.4.2 Output parameters 8
3.4.3 Embedding example 8
3.5 VerifyPIN 9
3.5.1 Input parameters 9
3.5.2 Output parameters 9
3.5.3 Embedding example 9
4 Return values 10
4.1 Return value list 10
4.2 POST examples: 11
5 PUK encryption 12
5.1 Example 12
6 Localisation 13
6.1 Parameter list 13
6.2 Example 14

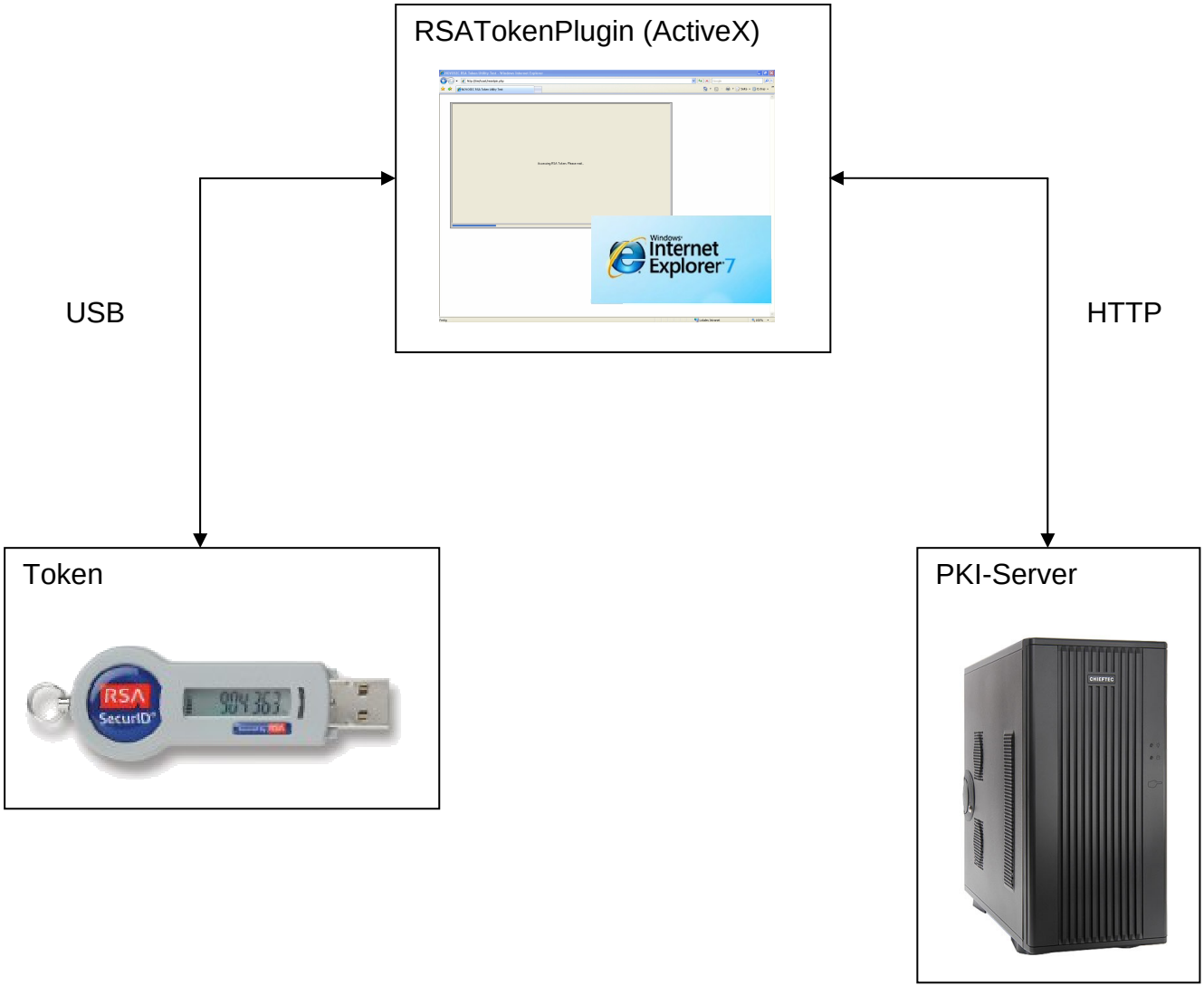
1 Description

1.1 Purpose

The ActiveX-Control is a supplement to the RSA Control Center for use with Microsoft Internet Explorer. Running the control in IE, an RSA SecurID Token can be wiped for assignment to a different user or the PIN can be reset in case the user cannot remember the old PIN. For both functions, the PIN unlocking key (PUK) is required.
As additional functions, the current token owner can verify or change the user PIN without knowledge of the PUK.

1.2 Process flow

1. User starts RSATokenPlugin: Retrieve TokenID from Token (GetTokenID)
2. RSATokenPlugin posts TokenID to URL of PKI-Server
3. PKI-Server calls RSATokenPlugin: ResetPIN or ResetToken with token specific PUK
4. RSATokenPlugin posts result to URL specified when called



2 Installation and Embedding

2.1 Environment

The control is designed to run under Windows XP with Microsoft Internet Explorer 7.0.
The RSA Security Center or RSA Smart Card Middleware must be installed to provide the PKCS #11 library.

2.2 Installation

The RSATokenPlugin.dll must be copied to the user's hard drive. It must registered calling

```
regsvr32 INSTALLPATH\RSATokenPlugin.dll
```

2.3 Embedding

To be loaded by the Internet Explorer, the control must be embedded in a Website (HTML, PHP etc.). The following tag is used:

```
<object id="RSATokenPlugin"
               width="70%"
               height="60%"
               classid="clsid:4D41494B-7355-4337-834F-4E4F564F5345"

         <param name="Request" value="FUNCTION_TO_BE_CALLED">
         <param name="PostURL"
          value="CORRESPONDING_POST_URL">
         <param name="PUK" value="TOKEN_SPECIFIC_ENCRYPTED_PUK">

</object>
```

2.4 Calling the embedded ActiveX-Control

To call the ActiveX Control, simply load the site in the Internet Explorer. Note that ActiveX must be activated in Internet Explorer, otherwise the control will not be loaded.

3 Functions

3.1 GetTokenID

GetTokenID retrieves the unique TokenID and posts this ID to the URL given in the input parameter “PostURL”.

3.1.1 Input parameters

Param name	Param value / description
Request	“GetTokenID”, string value
PostURL	where to post the TokenID, string value of valid URL

3.1.2 Output parameters

See parameter list under 4.1.

3.1.3 Embedding example

```
<object id="RSATokenPlugin"
               width="70%"
               height="60%"
               classid="clsid:4D41494B-7355-4337-834F-4E4F564F5345"

         <param name="Request" value="GetTokenID">
         <param name="PostURL"
          value="http://TokenIDResponseURL.php">

</object>
```

3.2 ResetToken

The token is cleared; all information on the token is deleted and the user must assign a new PIN. During the deletion, the user PIN is changed to a random value. When the deletion is complete, the user is presented with the standard RSA PIN dialog. The PUK must be given as input parameter when calling this function.

3.2.1 Input parameters

Param name	Param value / description
Request	"ResetToken", string value
PostURL	where to post the TokenID, string value a of valid URL
PUK	the token specific AES-256-encrypted PUK, 16 byte hex value without blanks

3.2.2 Output parameters

See parameter list under 4.1.

3.2.3 Embedding example

```
<object id="RSATokenPlugin"
                width="70%"
                height="60%"
                classid="clsid:4D41494B-7355-4337-834F-4E4F564F5345"

         <param name="Request" value="ResetToken">
         <param name="PostURL"
          value="http://ResetTokenResponseURL.php">
         <param name="PUK" value=" d5a809fe651c2ba605ac98e614f29d99">

</object>
```

3.3 ResetPIN

The token PIN is reset without clearing the token. To enter the new PIN, the user is presented with the standard RSA PIN dialog. The PUK must be given as input parameter when calling this function.

3.3.1 Input parameters

Param name	Param value / description
Request	"ResetPIN", string value
PostURL	where to post the TokenID, string value of a valid URL
PUK	the token specific AES-256-encrypted PUK, 16 byte hex value without blanks

3.3.2 Output parameters

See parameter list under 4.1.

3.3.3 Embedding example

```
<object id="RSATokenPlugin"
                width="70%"
                height="60%"
                classid="clsid:4D41494B-7355-4337-834F-4E4F564F5345"

         <param name="Request" value="ResetPIN">
         <param name="PostURL"
          value="http://ResetPINResponseURL.php">
         <param name="PUK" value="d5a809fe651c2ba605ac98e614f29d99">

</object>
```

3.4 ChangePIN

The token PIN is changed without clearing the token. The old user PIN is needed. To enter the new PIN, the user is presented with the standard RSA PIN dialog.
By trying to log on repeatedly with the wrong PIN, the token may be locked and need resetting.

3.4.1 Input parameters

Param name	Param value / description
Request	"ChangePIN", string value
PostURL	where to post the TokenID, string value of a valid URL

3.4.2 Output parameters

See parameter list under 4.1.

3.4.3 Embedding example

```
<object id="RSATokenPlugin"
               width="70%"
               height="60%"
               classid="clsid:4D41494B-7355-4337-834F-4E4F564F5345"

         <param name="Request" value="ChangePIN">
         <param name="PostURL"
          value="http://ChangePINResponseURL.php">
</object>
```

3.5 VerifyPIN

The user can verify his PIN by logging on to the token. To verify the PIN, the user is presented with a customizable PIN dialog.
By trying to log on repeatedly with the wrong PIN, the token may be locked and need resetting.

3.5.1 Input parameters

Param name	Param value / description
Request	"VerifyPIN", string value
PostURL	where to post the TokenID, string value of a valid URL

3.5.2 Output parameters

See parameter list under 4.1.

3.5.3 Embedding example

```
<object id="RSATokenPlugin"
                width="70%"
                height="60%"
                classid="clsid:4D41494B-7355-4337-834F-4E4F564F5345"

         <param name="Request" value="VerifyPIN">
         <param name="PostURL"
          value="http://VerifyPINResponseURL.php">
</object>
```

4 Return values

The return POST is composed of up to three parameters:
Result (mandatory) – Reason (optional) –TokenID (optional)

4.1 Return value list

Result	Reason	TokenID	Description	GetTokenID	ResetToken	ResetPIN	ChangePIN	VerifyPIN
SUCCESS	–	**X**	The function terminated as planned.	**X**	**X**	**X**	**X**	**X**
ERROR	LibraryError	**–**	The control could not load the PKCS #11-library.	**X**	**X**	**X**	**X**	**X**
	PKCS11InitFailed	**–**	The initialisation of the PKCS #11library failed.	**X**	**X**	**X**	**X**	**X**
	TokenIDFailure	**–**	The control could not read the TokenID.	**X**	**X**	**X**	**X**	**X**
	WrongToken	**–**	The token found is no RSA SecureID token.	**X**	**X**	**X**	**X**	**X**
	TokenInternalError	**X**	An unspecified internal error occurred on the token. The TokenID field may be empty, depending on where the error occurred	**X**	**X**	**X**	**X**	**X**
	PUKInvalid	**X**	The control could not parse the PUK (wrong format). **The PUK was NOT sent to the token.**		**X**	**X**		
	PUKError	**X**	Login with PUK failed. **Caution: Repeatedly using the wrong PUK will permanently lock the token!**		**X**	**X**		
	TokenDeleteError	**X**	One or more token objects could not be deleted.		**X**			
	PINResetFailed	**X**	The token could not be initialized.		**X**	**X**		
	PINChangeError	**X**	The user PIN could not be changed.		**X**	**X**	**X**	
	WrongPINError	**X**	The login failed due to entering a wrong user PIN					**X**
	UnknownPluginCommand	**–**	Unsupported request value.	**X**	**X**	**X**	**X**	**X**
CANCEL	UserCancelledPINInput	**X**	A PIN dialog was cancelled.		**X**	**X**	**X**	**X**

4.2 POST examples:

```
"Result=SUCCESS&TokenID=1A01146404123618"

"Result=ERROR&Reason=PKCS11InitFailed"

"Result=ERROR&Reason=PUKInvalid&TokenID=1A01146404123618"

"Result=ERROR&Reason=TokenInternalError&TokenID="

"Result=CANCEL&Reason=UserCancelledPINInput&TokenID=1A01146404123618"
```

5 PUK encryption

In order to hamper eavesdropping, the PUK is encrypted before transmission from server to plugin. The algorithm used is AES-256 with a static key stored in the plugin. The PUK, which consists of 8 byte hex is padded to 16 byte hex. The padding algorithm is irrelevant because only the first 8 bytes of the decrypted hex value will be used by the plugin.

5.1 Example

Token PUK (8 bytes):
0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF

Padded token PUK (16 bytes):
0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

Note: the padding algorithm is irrelevant for the correct decryption since only the first 8 bytes of the decrypted key will be used.

AES-256 key (32 bytes):
0xCA, 0xFE, 0xCA, 0xFE, 0xCA, 0xFE, 0xCA, 0xFE,
0xCA, 0xFE, 0xCA, 0xFE, 0xCA, 0xFE, 0xCA, 0xFE,
0xCA, 0xFE, 0xCA, 0xFE, 0xCA, 0xFE, 0xCA, 0xFE,
0xCA, 0xFE, 0xCA, 0xFE, 0xCA, 0xFE, 0xCA, 0xFE

Encrypted token PUK (16 bytes):
0xB9, 0xEA, 0x86, 0xD2, 0x3A, 0xA3, 0x59, 0x96,
0xC9, 0x30, 0x35, 0x66, 0xDA, 0xF0, 0xD9, 0x2D

6 Localisation

The static texts, and the VerifyPIN dialog can be modified using parameters when loading the plugin. All parameters are optional, if not explicitly changed, the standard values will be used.

6.1 Parameter list

Param name	Location	Standard value
STATUS_LOADING	main window, plugin status text	RSA Token Utility
STATUS_INSERT_TOKEN	main window, plugin status text	Please insert RSA Token.
STATUS_ACCESS_TOKEN	main window, plugin status text	Accessing RSA Token. Please wait...
STATUS_TRANSFER_DATA	main window, plugin status text	Transfering data...
STATUS_USER_INPUT	main window, plugin status text	Waiting for user input...
TITLE_VERIFYPIN	dialog, window title	Verify PIN
STATIC_USERPIN	dialog, static text	User PIN:
BUTTON_OK	dialog, button text	OK
BUTTON_CANCEL	dialog, button text	Cancel

6.2 Example

```
<object id="RSATokenPlugin"
               width="70%"
               height="60%"
               classid="clsid:4D41494B-7355-4337-834F-4E4F564F5345"
               codebase="RSATokenPlugin.dll#version=1,0,0,1">

         <param name="Request" value="ResetPIN">
         <param name="PostURL"
            value="http://tim/toast/ResetPINResponseURL.php">
         <param name="PUK" value="ccf3aa24e96db2711da905ec52699bb3">

         <param name="STATUS_LOADING" value="RSA Token Plugin">
         <param name="STATUS_INSERT_TOKEN"
            value="Bitte stecken Sie den RSA Token ein.">
         <param name="STATUS_ACCESS_TOKEN"
            value="Zugriff auf den RSA Token. Bitte warten...">
         <param name="STATUS_TRANSFER_DATA" value="Übertrage Daten...">
         <param name="STATUS_USER_INPUT"
            value="Warte auf Eingaben des Benutzers...">
         <param name="TITLE_VERIFYPIN_TITLE" value="Token PIN eingeben">
         <param name="STATIC_USERPIN" value="Token PIN:">
         <param name="BUTTON_OK" value="OK">
         <param name="BUTTON_CANCEL" value="Abbrechen">

</object>
```