Sharing data between RTE and a Partner using the HTTPS protocol

Date d'applicabilité : 03/12/2010

SOMMAIRE

1. Purpose of this document

This document is intended for partners with which RTE will be electronically sharing data using the HTTP (*Hypertext Transfer Protocol Secure*) protocol.

The document sets out the settings and configurations required for this data sharing – whether RTE is sharing data with a partner, or the partner is sharing data with RTE.

The types of data sharing described do not presuppose the use of any particular tool, nor do they require that any particular tool be used. The https protocol is a standard (see IETF RFC 2818).

2. Required network configuration

The partner must have a high-speed connection to access the Internet.

Sharing data involves establishing a connection between one of RTE's front office devices (RTE's infrastructure for accessing data externally), and a device belonging to the partner.

RTE is unable to guarantee that the connection can be effectively established, nor that data can be properly transmitted over the partner's network or over the Internet. It is noted that the method used for this data sharing is based on a connection being established between two IT systems. It is therefore possible to immediately detect whether such sharing might be impossible – irrespective of the reason. The sender can therefore decide on whatever measures they deem necessary.

Note: RTE's IT system enjoys a high level of availability. This applies to accessing the Internet, and to the devices used for data sharing.

3. RTE receiving a file sent by a partner

RTE has a module for receiving files sent by a partner via the HTTPS protocol. This module works in the following way:

- To establish an HTTPS connection, the partner needs to authenticate itself using a certificate issued by RTE (see § 6 for details of how to obtain such a certificate).
- The file is attached to a "POST multipart" request. The term "file" here means an electronic document that does not need to be converted into a file when it is sent or received. Details and an example of an http POST multipart request are given in § 5.
- If the file has been received by the module, the module returns a Technical Acknowledgement of Receipt in the form of an HTTP response the body of which

is empty and the return code which is 200[1]. The contents of the attached file are neither checked, nor validated when the response is sent.

The module will respond with a different code (not 200) if the file has not been received by RTE. There are several possible causes:

- The HTTP request received is badly formed,
- Authentication incorrect (certificate not recognised or invalid),
- Partner not authorised (authorisation),
- Incorrect URL,
- Internal error, the service is unavailable.

RTE will communicate the precise URL to the partner so it can send files when setting up the data exchange.

4. RTE sending a file to a partner

RTE has a module for sending files to a partner via the HTTPS protocol. This module works in the following way:

1) The establishment of an HTTPS connection is requested by RTE's module to the URL given by the partner.
2) A piece of front-line equipment belonging to the partner must be able to establish an HTTPS connection by requesting authentication from the party requesting the connection (i.e., RTE's module) using one of the two following methods (*to be specified when implementing the data exchange*):
   a. Authentication by using an X509 software certificate issued by the partner to RTE.
   b. Basic Authentication, i.e. a combination of a username and a password issued by the partner to RTE.
3) RTE's module authenticates the server certificate submitted by the partner's front-line equipment. In order to do this, it must identify and trust the partner's certification authority. The complete certification chain must therefore be supplied to RTE for implementation.
4) RTE's module sends the file via a "POST multipart" request and then waits for a Technical Acknowledgement of Receipt, i.e., a "200" return code, signifying that the file has been safely received, or otherwise a different code. The partner's front-line equipment must be able to receive the file and send this Technical Acknowledgement of Receipt. Details and an example of an http "POST multipart" request are given in § 5.

[1] The usual meaning of the "200" HTTP return code is "Request successfully handled".

5. “http POST Multipart” request

5.1 Definition

Using the "POST multipart" request ensures that a client software programme can send a file over HTTP to a web server.

It involves transferring any fields from a file tracking form in the body of the HTTP request (made up of several sections).

This file sharing mode is defined in the IETF’s RFC 2616 and RFC 1867 documents and is commonly used when uploading a file using HTML forms (see an example in § 5.3).

5.2 Request example

In the following example, the file sent is called "**test001.txt**". It contains the character string: “test 123456789”

Establishing the connection
https://domain.com/myserver/upload/appli/

http POST multipart request
POST /myserver/upload/appli HTTP/1.1
Host: - Domain.com
Accept: text/html,application
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
*Content-Type: multipart/form-data; boundary=**----------265001916915724***
Content-Length: 213
----------265001916915724
*Content-Disposition: form-data; name="monFichier"; filename="**test001.txt**"*
Content-Type: text/plain

test 123456789
***----------265001916915724**--*

OK response from the server
HTTP/1.1 200 OK

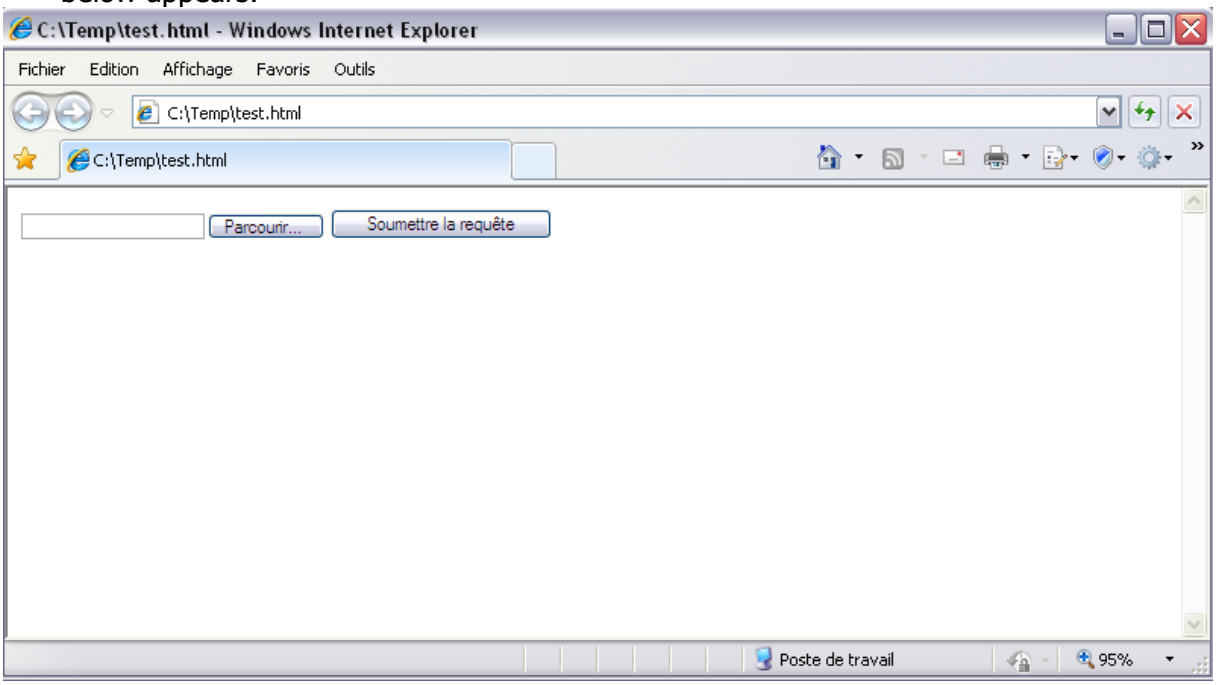
Note:
- RTE imposes no constraints on the HTTP request's headers.
- The name of the parameter defining the section (of the HTTP POST multipart request) containing the file is free. In the example given above, the name selected is “monFichier”.

5.3 Test form

It is easy to generate such an HTTP request and test that a server works properly by using a simple HTML form, coded as follows:

```
<html>
   <body>
      <form name="monFormulaire" method="post" action="
https://domain.com/myserver/appli " enctype="multipart/form-data">
         <input name="monFichier" type="file"/>
         <input type="submit"/>
      </form>
   </body>
</html>
```

When run in a browser (Internet Explorer in this case), a screen such as the one shown below appears:



Selecting a file (*Browse*) and then sending it (*Submit the request*) results in all of the steps described in § 4 being executed, starting when a server responds to the URL https://domain.com/myserver/upload/appli/. The navigator itself checks that the remote domain has been authenticated and handles the user authentication request.

The file is then sent to the server using an HTTP "POST multipart" request. A 200 return code with an empty body will result in a blank page being displayed.

5.4 Example in cURL

See appendix.

6. Certificates needed for mutual authentication when sending a file from a partner to RTE

6.1 Specifications of the client software used by the partner

The partner must access RTE's IT system using client software that can establish an encrypted connection (HTTPS protocol) with a server, and that enables mutual authentication between the client and the server. To do this, the software must meet the following conditions:

- Activate and manage the SSLv3 or TLS protocol.
- Be able to authenticate itself to RTE's server using a software certificate issued by RTE.

Requesting and collecting X509 software certificates, issued by RTE's Certification Authority, are described further down.

6.2 RTE authenticating the partner using a client certificate issued by RTE

Request submitted to RTE for a client software certificate

The partner submits a request to access RTE's IT services via RTE project manager. The project manager will give the partner a form so they can collect a so-called "client" software certificate. This must be installed with the software or the workstation used to establish the encrypted connection to RTE's IT services.

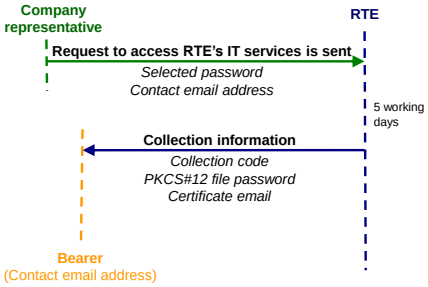
Collecting the client software certificate issued by RTE

Once it has received the form, RTE registers and approves the certificate request. A notification "Accessing RTE's IT services" email is sent to the "Contact email address" given in the form submitted to request access to RTE's IT services. This address is the address of the physical person referred to as the "Certificate bearer" who will be in charge of storing and using the certificate.

The partner collects the certificate using the "selected password" communicated to RTE in the form submitted to request access to RTE's IT services, together with the additional information given in the notification email.

To collect the certificate, the Certificate bearer logs on – using a workstation with Internet access and a browser – [2]to the software certificate collection website the URL of which is given in the notification email. They download the certificate in the form of a key pair (public and private certificate) as a PKCS#12 file (i.e., its extension is ".p12").

[2] To log onto the collection interface, RTE recommends that the client use browser versions that are supported by RTE (see General IT Appendix § 8.)



For more details about collecting, installing and using the software certificate, a user manual is available at the following address:

http://clients.rte-france.com/lang/fr/visiteurs/accueil/portail.jsp

Notes:

- It takes approximately three days starting on the date the request is submitted to issue and then integrate the certificate into the partner's IT systems.

- The client software certificate is valid for 3 years and its key is 2048 bits. Forty days before the certificate expires, an email is sent to the contact's email address, informing the Certificate bearer that their certificate is about to expire. It is therefore necessary and important to let RTE know of any changes in the Certificate bearer or the contact email address so as to avoid any technical disruptions to data being exchanged in the event of this period elapsing.

- The X509 client software certificate, issued by RTE's certification authority, must be presented by the client software used to establish the HTTPS connection with RTE's IT services. It is used to authenticate the partner to RTE. During this phase, RTE's reception module checks that the certificate has been issued by RTE's certification authority, and then checks that it is still valid and has not been revoked.

6.3 Authentication of RTE's server by the partner

The certificate presented by RTE's server when it receives a request to establish an HTTPS connection is a certificate issued by Verisign that is valid for 3 years from its issue date.

Accepting the certificate presented by RTE's server

The partner's client software must be configured to accept the certificate presented by RTE's server. Possible solutions for this are as follows:

	Solution	Advantage	How it works
A	None	Simple.	The partner does not authenticate RTE's server. There is a security risk associated with this way of doing things

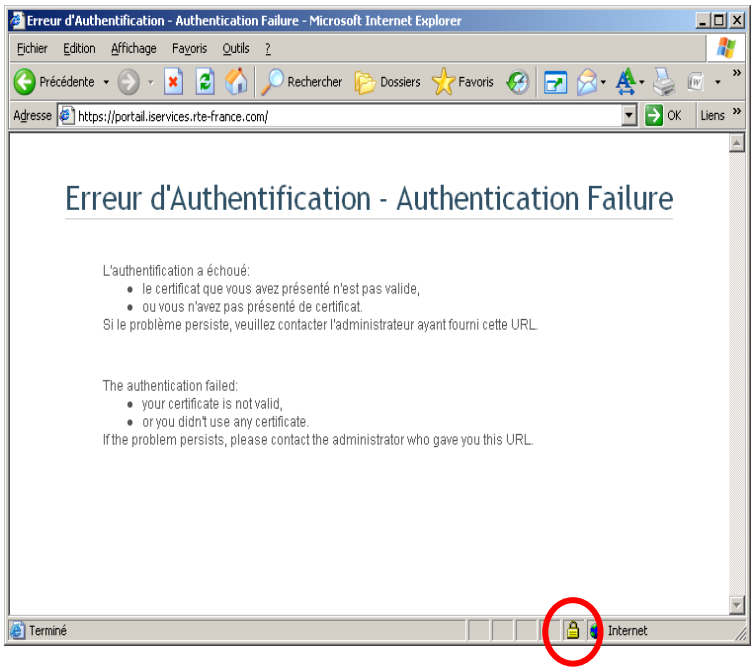
			(phishing).
B	Verification and trust in the "RTE server certificate"	The partner authenticates RTE's server.	Configuration: The partner configures their software with RTE's server certificate". How it works: The partner's software recognises that the certificate presented by the server to which it is logging on is indeed the one that it is configured to be able to access. Disadvantage: When the RTE server certificate is renewed, the partner has to configure their software with the new certificate. To avoid disruptions to the service, this operation should be performed on the day that RTE changes certificate. Note: RTE always knows a few weeks in advance when it is going to change certificate. The partner should therefore contact RTE to find out the date on which the new certificate is going to be installed.
C	Verification of RTE's server certificate and trust in one of the certificates in the chain.	The partner authenticates RTE's server. Nothing needs to be done when the RTE server certificate is renewed (if there are no changes to the certification chain).	Configuration: The partner configures their software with one of the certificates in the RTE server's certification chain. How it works: The partner verifies the certification chain presented by RTE's server, i.e., it checks that the sequenced list is correctly signed and that one of the certificates in this chain is the one that has been configured. The partner's software checks that the first certificate in the list has been issued for the URL called. Note: The intermediary or root certificates also have expiry dates. But they remain valid for much longer periods.

RTE recommends solution "C" *(this is the verification method used by web browsers).*

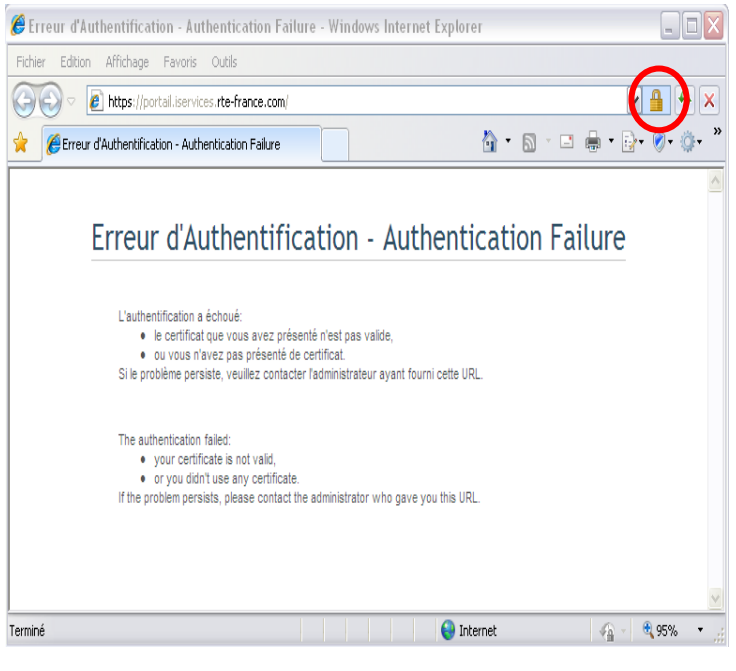
Obtaining the certification chain from RTE's server

The certification chain, i.e., all of the certificates from RTE's certificate right up to the Verisign root certificate, can be retrieved by the partner as follows:

- Log on to https://portail.iservices.rte-france.com. You are asked to authenticate yourself: click **Cancel.** In Internet Explorer, clicking on the padlock in the bottom right-hand corner of the screen, or next to the URL shows the various certificates in the certification chain used by RTE. They can also be installed in the browser and exported.



Obtaining the certification chain in Internet Explorer 6



Obtaining the certification chain in Internet Explorer 8

7. Certificates needed for mutual authentication when sending a file from RTE to a partner

For cases when RTE's issuing module needs to be authenticated by the partner's IT system with a software certificate that it issues itself, it must let RTE know what the request and collection procedure is, together with details of how long it takes for certificates to be provided.

8. Managing errors, Diagnostics procedures and Traceability of data exchanges

Each party must have procedures and means for conducting rapid diagnostics and analyses when exchanging data and in the event of an incident.

In particular, the absence of a return code or an incorrect return code after a file has been sent is an event which needs to be processed, communicated and viewable by a

management application. This is so as to rapidly identify an incident (e.g., telecommunications problem over the Internet) and decide on what measures need to be taken.

If there is intermediary equipment between the sending and receiving modules in their IT system (e.g.: firewall, proxy, reverse proxy, etc.), the partner must ensure that data exchanged can be traced so that requests leaving their IT system can be analysed and verified, together with requests received by its IT system.

RTE is capable of analysing all data received from a partner, or all requests issued by RTE's IT system over a 7-day period in the event of a major incident.

9. Appendix: cURL example

The following example is for illustrative purposes only. The scripts, software packages and any ad hoc development work done to exchange data must be carried out by competent persons. RTE does not provide any support for implementing this example.

cURL (Client URL Request Library) is an online command interface for accessing resources localised by URLs http://curl.haxx.se.

The cURL command line code below sends a file to a target server designated by a URL.

curl.exe --cacert IntCA.cer --cert userCert.pem --form upload=@test001.txt https://domain.com/myserver/upload/appli/Appli1

- "IntCA.cer" is the file that contains the root certificate from the Certification Authority which issued the target server's certificate. It is used to authenticate the target server.
- "test001.txt" is the file to send.
- "https://domain.com/myserver/upload/appli/Appli1" is the target server's URL.
- "userCert.pem" is the file which contains the client certificate (the private key) for enabling the issuer to authenticate itself to the target server.
- Notes:
  - ✓ In this example, the cURL programme and the files which are referenced are all placed in the common directory from which the command is run.
  - ✓ The "openssl" programme (http://www.openssl.org) is used to convert files containing certificates into files in ".cer" or ".pem" format.