T2S Non Repudiation of Origin (NRO)

Implementation of non-repudiation for U2A (CR 0466)

Technical Document

Author	FAM
Version	1.6
Date	16/02/2017
Status	final
Classification	
Accessible	
Classified until	

History of releases

RELEASE	DATE	ISSUES	STATUS[1]
1.0.	30/01/2015	First draft – as explained in PMG telco on 8 December 2014, relevant assessments parts, including NSP-related parts, are still ongoing and might result in noticeable changes	draft
1.0	28/02/2015	Final version – as explained in PMG telco on 8 December 2014, relevant assessments parts, including NSP-related parts, are still ongoing and might result in noticeable changes	final
1.1	09/03/2015	Updated version – results of ongoing assessment activities are included	final
1.2	27/03/2015	Updated version – issues raised during the PMG workshop on 11/03/2015 and respective clarifications are included	final
1.3	10/08/2015	Updated version – new qualified configuration.	
1.4	21/12/2015	Updated version – clarifications from user testing	
1.5	21/01/2016	Updated version – clarifications from user testing	
1.6	16/02/2017	Updated version – changes to recommended configurations	

Reference documents

REFERENCE		OBJECT
1	T2S-0466-BFD	CR for implementation of non-repudiation of origin for U2A
2	MOP	Manual of Operational Procedures V1.0

[1] Status value : Draft, Open, Final, Dismiss

1. Introduction

1.1. Context of the document

This document describes the general technical framework to help the customer to implement the requested non-repudiation of origin functionality (NRO) as enhanced security requirement. The content of the document is the guideline for the T2S participants and informs about the prerequisites to implement and use the NRO-functionality.

This document reflects the status at the time of preparation. Subsequent changes are possible, especially as relevant assessment parts are still ongoing (cf. PMG telco on 8 December 2014).

Today in the application-to-application (A2A) – instruction channel a technical and a business signature is implemented as standard feature and fulfils the NRO-requirements. T2S participants demand the same level on user to application (U2A) - side.

In September 2013 the Eurosystem presented three solutions for the implementation of non-repudiation of origin for instructions initiated in U2A -mode:

Option 1: Implementation of NRO for each transaction initiated by a user.

Option 2: Implementation of NRO for critical transactions only. Critical transactions are specified by the customers and listed in CR-0466.

Option 3: NRO achieved by “non-repudiation of emission”, i.e. a mechanism for signing digitally U2A messages at network level, and legal amendments.

After analysis of the feasibility and user-friendliness of the proposed solutions, a fourth option using existing security features in T2S and in the networks together with a legal framework similar to the framework known from TARGET2 was discussed.

In its July 2014 meeting the CSG endorsed the decision to implement option 2 ahead of the migration of wave 2. The following description refers to option 2 which is described in CR-0466.

1.2. Definition of NRO

NRO provides the recipient with the evidence NRO which ensures that the originator will not be able to (i) deny having sent the U2A instruction; or (ii) claim having sent the U2A instruction with a different content.The evidence of origin is generated by the originator and held by the recipient.

1.3. Network Service Provider responsibility

The procedures related to the management of the certificate, signature and PIN - issuance, distribution, revocation, locking after failed attempts, credential management, etc. - are individual and specific to each network service provider (NSP) and the description can be found in the technical documentation attached to the contract signed by each Directly Connected Actor and the selected NSP. In general, the rules and procedures applicable to the authentication certificates provided by the selected NSP will apply also to the digital signing certificates. However, there will be no interference between both certificates.

The certificate to be used for the digital signature will be issued by the NSP and stored on the same device used for storing the certificate for the user authentication. Both NSPs (SWIFT/SIA-COLT) have different procedures to support the T2S-participants:

- SWIFT: delivers an upload-procedure for the second certificate (and private key) for the existing e-token based on the PKCS#11-specification.
- SIA-COLT: all SIA-COLT clients have already their second certificate (and private key) on the existing e-token

SWIFT tokens will not need to be changed. A procedure is being defined to load an additional certificate + private key on the token. SIA clients should not need to change the tokens as well. In any case the authentication and the signing certificate will have the same DN as well as the same validity date.

Information regarding the availability of e-token and detailed procedures for handling e-tokens, being part of NSPs' responsibilities, will be provided by NSPs.

1.4. Readers Guide

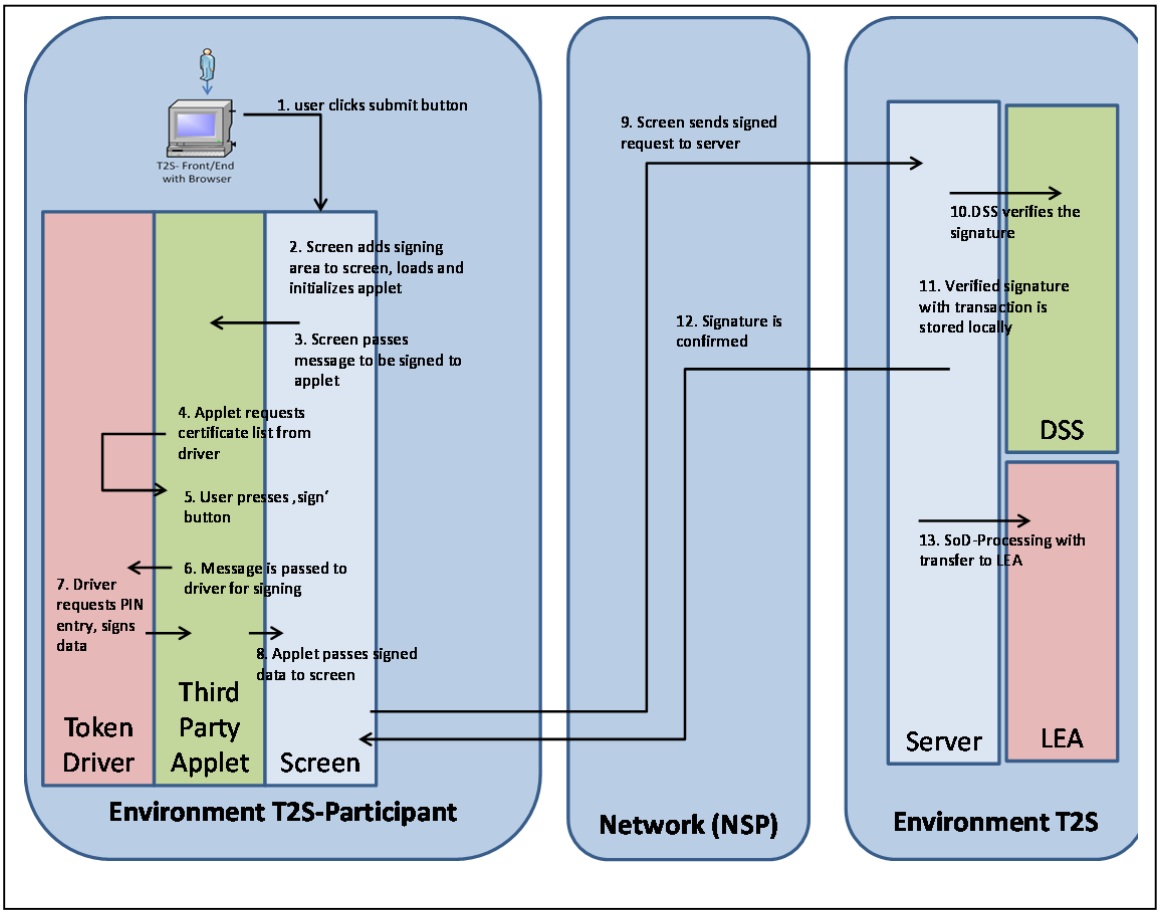
The following matrix provides a consolidated overview of the responsibility of the T2S participants.

		T2S-participants tasks	
Techn. doc. chapter	**Issue**	**SWIFT**	**SIA-COLT**
1.3	digital signature	Upload the second certificate to the existing e-token	Check availability of the second certificate on the existing e-token
2.6.3.	technical requirements	Ensure fulfilment of compatibility matrix supported by the NSPs	Ensure fulfilment of compatibility matrix supported by the NSPs
	system set up and updates*	Ensure fulfilment of the NSP recommendations	Make sure in line with the NSP recommendations

2. Process Overview

2.1. Business Process Overview (without error handling)

The following business workflow shows the interactions of the actors in the U2A signature procedure and their confirmation by the Digital Signature Server (DSS) and serves to establish a better understanding between functional and technical procedures.



2.2. Process Description (without error handling)

In order to perform a digital signature, the user needs a private key associated with a public certificate stored in a portable device (e-token) accessible by entering a PIN- code. As defined by the NSPs, the PIN code applies to the token, i.e. it is the security password for accessing the token and its content and as such, the same PIN is used for accessing all certificates stored within the token.This PIN will be then also used for signing the selected instructions initiated by the user (other than for authentication).

1) The user enters an instruction into the input fields of the U2A screen and clicks the submit-button. Working in multiple windows is not supported[2].
2) The screen adds the signing area (a screen area containing the signing applet) to the current screen, loads and initializes the applet. The applet appears on the screen, allowing for scrolling and viewing the complete data entered on the screen while signing.
3) The screen passes the message to be signed to the applet which verifies that the message to be signed is present and correct.
4) If check is ok, the applet requests a list of available certificates from the driver then displaying the signature one to the user. Steps 2 to 4 happen without user interaction in a short time span.
5) The user presses the “ok/sign” button of the applet which will call the token driver in order to establish the PKCS#11 session.
6) PIN is requested by the applet/token driver to the user (as first log-in step during PKCS #11 session establishment with the token). This is required for every signing request (in order to authenticate the application/applet to the token).
7) Once correct PIN is entered in the popup window, the applet finally establishes a PKCS #11 session with the key token, which is in turn necessary to elaborate message signature. The user can move the popup window free over his screen, i.e. the position of the popup windows is selectable.

   Only screens used for performing sensitive U2A functions (as defined by the user in CR-0466) require entering the PIN.

   The signing of the instruction is performed either in 2-eyes mode or in 4-eyes mode. The choice depends on the set-up of the T2S-GUI for the participant. If the U2A function is set up in 2-eyes mode, the user who submits the request has to sign his request.

   If the U2A function is set up in 4-eyes mode, the user who submits the request has to sign the initial request and the second user has to confirm with his respective signature.
8) The applet can finally elaborate message signature (through PKCS #11 standard functions/API available on the token), build the signed message and then transfer the signed request back to the U2A screen. As expected, the private key used for cryptographic calculation never leaves the token.
9) The screen sends the signed request to the T2S server, where a Digital Signature Server (DSS) will verify the signature as well as the validity of signature certificate.
10) DSS performs the above mentioned verification activities.
11) If both the signature and the certificate are valid, the request is forwarded to the back-end processing and the request is stored (T2S-environment).
12) Meanwhile the server sends the confirmation of the acceptance of the signature to the user.

[2] Cf. UHB Version 2.1, page 47.

13) During the SoD-processing the signed requests are extracted and transferred to the legal archiving module (LEA) for archiving.

2.3. Business Process Overview (with error handling)

As there are many different error conditions, a graphic description is not deemed adequate.

2.4. Process Description (with error handling)

The following error handling is focused on the user side and is described up to the second signature for transferring the sensitive U2A functions. Network security features are not described. For more details on login credentials see chapter 1.3. (NSP responsibility).

T2S does not provide any reporting of logging information as in the current design defined in CR-0466 error information is not transferred from the client to the server. All scenarios that do not result in a signed message are of no consequence to the backend - regardless of the cause of the error prohibiting signing (e.g. no or wrong PIN entered, certificate not shown by the applet, etc.), no processing happens as no message is signed. Thus, so far no possibility to archive or log errors on the client side exists.[3]

In case the input message is eventually missing or not well formatted, the applet will not start and will raise an error blocking in the application. User may need to open a ticket in this case.

In case user certificate is expired the applet will not show it thereby stopping the signature process. A revoked but not yet expired certificate will be eventually identified on server side during message verification; the DSS module will report accordingly the error to T2S back-end application which will, in turn, track this in the log files. It has to be noted that in compliance with A2A traffic, and according with user requirements, erroneous messages are not transferred to LEA.

In case certificate is valid, the user is demanded for PIN-entry into the popup window:

- With a positive PIN-entry a handover starts the request to the third party applet.
- With a negative PIN-entry the response leads to a request for a second attempt.
- After “wrong PIN threshold” reached, message signature process fails and the U2A screen will inform user that the instruction was not processed. Subject to further detailing by NSPs, the token will then be locked and the customer has to activate an ad-hoc procedure with NSPs to unlock it or eventually request a new one – please refer to NSPs documentation for additional details.

[3] The requirement to report login information was not raised during the scope and content definition of the CR-0466. This clarification is included as a PMG workshop (11/03/2015) follow-up.

Remarks: the minimum PIN length and its restrictions and the maximum number of failed attempts before lock off are described in chapter 2.4.1. You must be aware that the NSP´s have different specifications.

In case no certificate is available for signing the customer may need to verify correct token installation and restart the signature process.

In case required PIN input field is left empty, a message will be displayed to the user in an applet pop-up dialog window.

Finally, in case signing is cancelled (i.e. user cancels the PIN entry) or fails for technical reasons, an error will be returned by the token driver to the applet. Appropriate mechanism will be put in place to inform the user accordingly.

As described in UHB V2.0, after 10 minutes of inactivity in the live-environment T2S will automatically log you out.

2.4.1. Token login credential policies

The main credential policies are summarized in the following table:

	SWIFT	SIA/COLT
Minimum PIN Length	4	4
PIN restrictions	Alphanumeric. At least two different characters must be used	Any number of ASCII characters
Maximum number of consecutive failed attempts before the token is locked	5	3

Signatures contained in U2A instructions are created by the originator and verified by the recipient (T2S) at runtime. In case of unsuccessful verification of a signature at runtime, an error message must be generated. The user session must be terminated and an alert must be sent to the standard technical monitoring system allowing the post treatment.

2.5. Restrictions

2.5.1. Trusted Archiving

The non-repudiation functionality is obtained by associating the digital signature technique with a trusted archiving system, i.e. legal archiving, to be able in case of legal action to provide evidence and to prove that an event actually occurred in the past.

The digital signature will be stored in the inbound communication table of the Interface Domain, which is itself stored at the start of the day in the legal archiving.

The retrieval of archived transactions from legal archiving is under the responsibility of T2S Service Desk by request and described in the Manual of Operational Procedures V1.0., Chapter 3.4.4. The maximum time-frame to get the requested archived data will be three days.

No dedicated customer queries and reports on NRO related static and dynamic data will be possible.

2.5.2. Feasibility of “show what you sign” functionality

Following discussions at ISSG-level in summer 2014, the “show what you sign” functionality describes displaying all screen-data foreseen for signing before entering the PIN. CSDs requested this functionality in order to (i) avoid wrong inputs (by checking the correctness of the entered transaction); and (ii) prevent manipulation between the user´s input and the signed process.

The following paragraph provides the pertinent feasibility assessment and the way in which the issues above are addressed. In this context two options were analysed for implementation, i.e. at applet level or at T2S-level:

- An incorrect entering data into the GUI can be avoided by applying the 4-eyes principle (U2A NRO will be implemented for both 4-eyes- and 2-eyes mode). It has to be mentioned that the four eyes mode allows with the second leg to review the message sent for the first leg, i.e. if theoretically data were manipulated in the first leg, the 4-eyes principle would support the detection. If data would be manipulated in the second leg, the data of the first leg would support having evidence.
- Transactions, with their digital signature, will be stored within the T2S archive service in order to provide evidence in case of need.
- The "show what you sign" functionality, as described in summer 2014 by the ISSG, would require to be implemented at applet level.
- However, as the browser is deemed an unsafe area, the applet - which is started and controlled by the browser - is also in the unsafe area. If access to this unsafe area is compromised, it is technically very simple to replace the applet with a version that shows different data than it actually signs, behaving and looking identical to the bona fide applet in every other respect.
- Thus, “show what you sign” cannot provide protection for local data manipulation, even when implemented in the applet.
- Accordingly, the applet does not support "show what you sign" in the requested sense, and no other applet could provide this due to the inherent dependency on the browser.
- The implementation allows for scrolling and viewing the screen with the entered instruction data blocked when asked to sign.

As a follow-up to the PMG workshop on 11/03/2015 the screen print functionality has been analysed. Most T2S GUI screens provide printing functionality. Additionally, built-in browser printing is always available, as is the possibility to print screenshots.

However, a printout of a screen does not provide evidence of any kind as:

- A printout can be easily generated without ever actually sending the instruction to the server and
- It is very easy to manipulate a website to generate a printout with fake data by either:
    - Manipulating the technical contents of the page before printing or

- Using image manipulation software to modify the contents of the generated printout.

Thus, a printout is no evidence with regard to what has or has not been submitted to T2S - in terms of NRO, it does not have any value. The 4-eyes-principle, as explained above, can be used to avoid human errors. In addition, the A2A functionality can be used to retrieve information about submitted instructions, as well as the corresponding U2A screens.

2.6. Current technical requirements and recommendations

2.6.1. Third-Party Applet Requirements

The applet installation on T2S user side will be triggered with the first attempt to sign an instruction (and each time the user needs to sign one) and is transparent to users once the security warning asking for IBM applet installation[4] is explicitly accepted by the user:

For security reasons, the applet (jar) archive delivered by the provider will be code signed by a trusted certification authority (Thawte) to ensure applet integrity on customer side.

In order to properly execute Thawte certificate revocation checks, customer should ensure proper Certificate Distribution List availability; these could be downloaded from the Internet at the following URLs:

http://crl.thawte.com/ThawtePremiumServerCA.crl
http://crl.thawte.com/ThawtePCA.crl
http://th.symcb.com/th.crl

The T2S users have to ensure that the security settings of their institutions, i.e. firewalls, allow for installation of the applet as well as for code signing certificate revocation check, if not generally disabled

[4] IBM CBT thinclient applet should be accepted as trusted application.

2.6.2. Other technical requirements

For what concern terminal services solutions, as T2S provides the U2A services while NSPs take care of connectivity solution to T2S platform, customers could refer to NSPs to gather information about terminal services support for their specific U2A connectivity scenario.

2.6.3. Recommended Configurations

A specific subset of the NSPs compatibility matrix has been qualified by the Eurosystem. These configurations have been extensively tested and support on them is guaranteed.

NSP	SWIFT	SIA-COLT
OS	Windows 7	
Browser	Microsoft Internet Explorer 11[5] Mozilla Firefox v52 ESR	
JRE[6] Version	1.8.0_31 or higher. (The SWIFT solution for T2S U2A is not compatible with 64-bit version of Java)	1.8.0_31 or higher. (following the availability of the CISCO SSL VPN Gateway software)

Firefox v52 ESR will be the last version to be qualified by the 4CB, following decision by Mozilla to completely drop NPAPI support, thus directly blocking java applets on which the NRO solution is based. Firefox v52 decommissioning is currently scheduled in Q1 2018. Migration to Firefox v52 will therefore prevent the correct usage of the T2S GUI.

According to current Microsoft plans, IE11 will be supported till 2020 and the 4CB plan to keep it in the qualified configurations accordingly. The complete redefinition of the qualified browsers' set and the full replacement of the applet technology will be then defined in the context of the Eurosystem Singe Market Infrastructure Gateway, being part of the ongoing T2/T2S consolidation project in line with the agreed schedule.

The 4CB will ask customers running a software version lower than that qualified to upgrade to a qualified version in order to proceed with problem investigation.

[5] The compatibility view mode must be disabled.

[6] Following Google's decision to drop support for Java (as well as other plug-ins running the Netscape Plug-in API) all the screens changed with CR-466 „Implementation of non-repudiation for U2A" will be not functioning as of Chrome v.42.

The 4CB will investigate issues experienced by customers having issues while running a software version higher than that qualified: If the root cause is linked to the specific software version, then the 4CB will attempt to find a workaround (which may involve customers downgrading their software to a qualified version). The 4CB will evaluate whether a fix for the issue can be included in a future T2S GUI release.

Customers using totally or partially different system components or versions than those mentioned are then responsible to verify the full compatibility with the T2S GUI in the test environments and the system.

The local customer system set-up, i.e. adaption of the local firewall and security policies in order to enable the applet download, HTTPS file transfer, access to the certificate, etc. is out of control of T2S and is under the responsibility of T2S participants.
According to some customers' experience, configuration of the following windows parameters could be activated on specific workstations in order to improve responsiveness of instruction signature[7]:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\AFD\Parameters]
"DefaultSendWindow"=dword:0000fc00
"DefaultReceiveWindow"=dword:0000fc00
```

For troubleshooting, if emptying Java cache has not already solve the issue, the customer should follow the below-mentioned indications in order to collect relevant information for analysis:

- Taking step-by-step screenshots providing a brief description of each step,
- Setting the trace level to '5' in the applet JAVA Console,
- Activating in the web browser console (open in IE using F12 key, in Firefox using crtl+shift+J) the network information collection (network tab) and providing both network statistics and copy of the "console" tab.

The local windows event viewer could also be activated in order to detect any local issue linked to cryptographic activity on windows (CAPI2 events logger). The events viewer can only be activated in administrator mode.[8] Useful link: https://technet.microsoft.com/en-us/library/cc749296(v=WS.10).aspx

[7] When a PC is connected to a network, in the local PC settings a limit is set on the capacity to upload files on this network. This buffer to upload could be by default rather limited, making the upload of big files for a user more time-consuming. An IT administrator could change these local settings in order to allow overall greater responsiveness of sending and receiving traffic over the network. We recommend testing this on a test workstation first.

8 Useful link: https://technet.microsoft.com/en-us/library/cc749296(v=WS.10).aspx

2.7. Risk assessment

In line with the Information Security Risk Treatment Plan (ISRTP) PPSA-06, the successful implementation of CR-0466 will provide an adequate mitigation of the risk identified. Once T2S Release 1.1 is in production the risk will be decreased to a level that can be considered acceptable according to the Eurosystem ORM Risk Management Policy.

Risk Treatment Plan ID	Description of the: (i) Proposed Action Plan; or (ii) Risk Acceptance Proposal		Residual Risk								Planned Implementation Date	Status
			Before				After [1]					
		Threat Id	Likelihood	Business	Reputation	Financial	Likelihood	Business	Reputation	Financial		
PPSA-06	Implementation of U2A signature (CR-0466 for release 1.1)	T52	1	1	3	5	1	1	3	2	Q1 2016	Started in Q2 2014

2.8. List of used Software and Hardware

Software:

- T2S is using a Java applet from IBM (also called third-party applet), which is integrated into the existing presentation layer
    - IBM Cryptographic Based Transactions (CBT).
        - CBT XML Signer API for CBT Thin Client Applet signature verification (applet variant v1.9.0)
- PKCS#11 (Public-Key Cryptography Standards) will be delivered with the second certificate (i.e. SWIFT) or is already delivered with the existing e-token (i.e. SIA-COLT).

The new IBM applet released on 15 January 2016 to EAC (2.0.0) and foreseen for deployment to Production on 5 March 2016 contains the following change from a user perspective:

- Removal of the “more keys” button originally included in the applet.

While SIA-COLT customers have already the second certificate on their e-token, for SWIFT customers a specific download procedure has to be applied.

Hardware:

On T2S customer side no additional hardware will be necessary for the usage of the NRO-functionality.