FUJ00078691

A&TC
Enterprise
Solutions

ICL Pathway Horizon Project

KM TeamWARE Crypto Test Specification

Author: Keith Simons

Reference:

Issue:

Date:

Abstract: This specification document captures the acceptance testing of the TWC release to be used as part of the Key Management System introduced at NR2+. TWC provides protection of filestore on PO counters controlled by the PMMC Agent.

Approver:

Signature & Date:

PCMS Reference:

Last Saved: 31 March 1999

This is a controlled document.

Check with the document controller (below) that this is the latest issue. An out-of-date issue or a non-approved issue **is not definitive**.

Controlled by: Pauline Grice

Location: BRA01
Phone: GRO

Electronic repository: Source Safe db = \\nt025\Agent_dev\vss\$/Pathway/Cryptography/Documents/Key Mgt Service (R2+ -KMS)/PMMC Agent/CRY077TWCTestSpec.doc

Distribution:

BRA01	Will Dawson	BRA01	Glynn Morgan
BRA01	Anjan Ghosh	BRA01	Sarah Munns
BRA01	Richard Glanville	BRA01	Anthony St.John
BRA01	Dave Johns	BRA01	Mark Scardifield
BRA01	Nick Lawman	BRA01	Mark Taylor
BRA01	Peter McMahon		

FUJ00078691

0. DOCUMENT CONTROL

0.1 Contents List

0.2 Changes in this Version

Issue 1.0 Approved version incorporating comments received.

0.3 Document History

0.1	12/03/99	First draft for internal review; sections 4 & 5 are incomplete
0.2	24/03/99	Sections 4 & 5 completed; comments incorporated
1.0	31/03/99	Approved version incorporating comments received

FUJ00078691

0.4 Cross References

KMPMMC	TSC/CRY/059	KM PMMC Agent Detailed Design	0.7
KMPMMC[U]	TSC/CRY/072	KM PMMC Agent Detailed Design Update Proposals	0.2
KMTERM	TSC/CRY/057	KM Management Terminology	0.3

0.5 Terminology and Abbreviations

See [KMTERM] and [KMPMMC].

TWG TeamWARE Group (Finland)

0.6 Conventions

[Describe any conventions used throughout the document]

1. INTRODUCTION

1.1 Scope

The purpose of this document is to present the details of the testing of TeamWARE Crypto to be carried out under the NR2+ KMS Development Plan during the 3RD PARTY INTEGRATION AND ACCEPTANCE phase. Note that this testing is not to be confused with any testing carried out during link testing which is separate to the KMS Development.

1.2 Background

TeamWARE Crypto is used to protect the filestore and swapfile on PO counters. At NR2+ the PMMC Agent [KMPMMC and KMPMMC[U]] puts in place this protection. The software component to carry out this role is PoLo.exe which existed at 1c and NR2.

TeamWARE Crypto is a product produced by the TeamWARE Group in Finland. It is an off-the-shelf product which is tailored by Crypto Development Team in Bracknell for use in Pathway. The productised version provides API interfaces without any of the GUI components.

The TWC used at NR2 is constructed from version 4.0-5 build 303 (25th August 1998). There have been two official versions since that used at NR2. The current official version of TWC is 4.0-8 build 322. The Crypto Development Team have submitted a short list of enhancements for inclusion in a new version.

2. TWC RELEASE APPROACH

The TeamWARE Group plan and thus the delivery date of the TWC release with enhancements is currently looking doubtful in the KMS timescale. If the release is not available in time then we have to decide to move to the latest TWC or possibly stay at the version used at NR2.

There is a known bug in TWC version 4.0-5 build 303 which does not allow the test on the encryption status of a file to distinguish between a file that is encrypted under one key from that under another. The NR2 PoLo code thus just tests for encrypted and non encrypted files. If a file encrypted under the wrong FEK (i.e. the wrong exchangeable disc had been inserted in a SCO) then the fault is only found later when the file is accessed. The information available at this point is very general (a response code of

FUJ00078691

ERROR_ACCESS_DENIED). Problems of this sort are unlikely in the field. However providing the development, test and integration teams (who are more likely to be moving files around) with better diagnostics will be time saving. The requirement specification for mirrored exchangeable disc working was watered down in this respect in the light of this deficiency in the TWC release.

It is expected that there would be more code changes between versions 4.0-5 and 4.0-8 than between versions 4.0-8 and the enhancement version. While the enhancement version is not available the early exposure of 4.0-8 will allow any faults (which will probably also exist in the enhancement) to be found early and reduce the chances of finding faults late on.

It is understood that version 4.0-5 is no longer supported by TWG apart from for Pathway.

It is proposed to use version 4.0-8 build 322 on development machines and for testing until the enhancement version is available. It will also be the fall back version should the enhancement not get delivered in time or is not accepted. If version 4.0-8 build 322 is the TWC used for NR2+ then the NR2 workaround methods (carried out by waiter.exe, PoloHold.exe) will need to be carried forward at NR2+. Note that the workarounds will be packaged differently and waiter.exe, PoloHold.exe etc will still be able to be withdrawn at NR2+.

It has been agreed with Alan D'Alvarez that we should go ahead on the assumption that the enhancement version will not be available in the KMS timescales. The testing described in this specification will make use of TWC version 4.0-8 build 322.

3. TESTING VIEWPOINTS

This section describes the testing of TWC from different viewpoints.

3.1 Regression Tests

This first viewpoint takes the NR2 software and runs with a later TWC version (4.0-8). A large proportion of the TWC functionality used at NR2+ was used at NR2. The NR2 software in this case is being used as a testing tool.

Note that at NR2+ migration the software will be changed including TWC. Should it be necessary to run the NR2 PoLo software to construct a lost NR2 PMMC or PIN then it is hoped that it will not be necessary to change the TWC version. It is currently assumed that the new TWC release (4.0-8) is backward compatible and that it can be used for both NR2 & NR2+ PoLo.

3.1.1 NR2 PoLo

The following table shows the different PoLo modes and the TWC API used.

	PoLo mode	Comments	TWC API
A1	all	get TWC keystore status	OpenAcmContext() GetPersonalKeyStatus() CloseAcmContext()
A2	all	test file encryption status	GetCryptoStatus()
A3	new counter	encrypt protected filestore	OpenAcmContextEx() CreateCryptoKey() SetCryptoInfo() I_StartTreeCryptTask() CloseAcmContext()

FUJ00078691

A4	normal reboot	unlock protected filestore	OpenAcmContextEx() CloseAcmContext()
A5	recovery reboot[1]	re-encrypt protected filestore	OpenAcmContextEx() ChangeCryptoKeyEx() CloseAcmContext()

Table 3-1 TWC API used by NR2 PoLo

[1] At NR2, PoLo recovery carries out a re-encryption of the filestore. Normally the NR2+ PoLo recovery will not carryout a re-encryption.

FUJ00078691

3.1.2 Post PoLo

When PoLo exits it leaves the TWC environment in place to provide registered applications with access to the protected filestore and the transparent underlying encryption / decryption operations.

Part of this TWC environment consists of shared memory buffers which must be left in position if TWC is to function correctly. The mechanisms put in place at NR2 to ensure that these shared memory buffers do not get unloaded are as follows:

	TWC shared memory buffer	NR2 Software	Comments
B1	Crypto Lock Settings	PoloHold NT service started by PoLo	It causes the reference count on the buffer to be incremented and the software sleeps. As long as the service is not stopped the reference count is therefore not zero and thus the buffer does not get unloaded.
B2	Crypto Keys	waiter.exe part of System Management daily activities	Similar to above.

Table 3-2 NR2 software and TWC shared buffers

Without these mechanisms various situations caused the reference counts to become zero and the buffers got unloaded causing problems. This NR2 software above will be checked out on the new TWC release to establish that the TWC shared memory buffer still operate in the same way.

From recent prototyping experiments it has become known that just registering the software with TWC is insufficient to declare that that software is a Crypto application. Such an application is able to access the protected filestore with transparent encryption / decryption operations. A set of rules governing the definition of a Crypto application has been established and these needs to be checked out at the new TWC release (4.0-8).

The set of rules (**identified as B3**) allowing an application to act as a Crypto Application are:

a) the application must be declared as a Crypto Application (NT registry TeamWARE\Crypto\CryptoApps),

a) the appropriate parts of filestore are declared as protected (NT registry TeamWARE\Crypto\Permissions),

a) the TWC shared memory buffers must not be unloaded, and

a) the application should be a windows application[2]. (containing a load of USER32.DLL).

[2] There has been some unreliability with console applications under certain circumstances such that TWG have recommended that it is safer to use window applications.

FUJ00078691

3.2 Expected NR2+ TWC API usage differences

This viewpoint looks at the differences in the TWC API used at NR2 and that expected to be used at NR2+.

The first four subsections are named after the planned C++ methods to be used at NR2+

3.2.1 IsEncrypted()

This method is comparable to and is expected to make use of the same API as in row A1 of table 3-1.

3.2.2 Encrypt()

This method encrypts the protected filestore and allows Crypto applications to handle encrypted files. Its functionality is similar to that at NR2. The TWC APIs for NR2 are shown in row A3 of table 3-1.

At NR2 only NT directories (and therefore their files) are protected. For NR2+ the protection is extended to cover individual files. The TWC API differences are to do with the declaration and encryption of directories and files. The following API is expected to used:

	PoLo release	Comments	TWC API
C1	NR2+ (same as NR2)	declare directory for protection and encrypt any directory files	SetCryptoInfo() - declare directory; no encryption I_StartTreeCryptTask() - encrypt
C2	NR2+	declare a file for protection and encrypt if it exists	SetCryptoInfo() - declare file; encrypt

Table 3-3 TWC API to encrypt protected filestore at NR2+

3.2.3 Unlock()

This method unlocks the protected filestore and allows Crypto applications to handle encrypted files. Its functionality is similar to that at NR2. It is comparable to and is expected to make use of the same API as in row A4 of table 3-1.

3.2.4 ReEncrypt()

This method re-encrypts the protected filestore and allows Crypto applications to handle the encrypted files. It is comparable to and is expected to make use of the same API as in row A5 of table 3-1.

3.2.5 File Encryption Status

Tests are carried out on the status of files that exist in the areas of protected filestore when unlocking and encrypting. As noted in section 2 there is a bug in the TWC version (4.0-5) used at NR2 such that no checks are carried out by NR2 PoLo to distinguish between a file encrypted under the correct key to that encrypted under the wrong key. The API is expected to be the same as in row A2 of table 3-1. If the bug is cleared then it will be possible to make more use of one of the output parameters than previously and detect inconsistent situations earlier.

3.3 TWC Enhancements

The TeamWARE Group have been requested to consider making enhancements to the TeamWARE Crypto product. A formal request was made on the 25/02/99. These are identified as Enhancement No 1 to TeamWARE Crypto.

The proposed enhancements are described in the following sections.

3.3.1 ERROR_ACCESS_DENIED

This return value is to be produced when a registered application (as defined in the registry

FUJ00078691

HKLM\Software\TeamWARE\Crypto\CryptoApps) creates a file in protected filestore (as defined in the registry HKLM\Software\TeamWARE\Crypto\Permissions) and the Crypto Lock Settings shared memory buffer has disappeared.

The reason is to protect against a Crypto application writing to protected filestore unknowingly in clear.

Measures to ensure that the shared memory buffer does not get unloaded in the first place are in the next section.

3.3.2 Crypto Lock Settings shared memory buffer

A new registry flag provides a means to force every Crypto application to reference the Crypto Lock Settings shared memory buffer when the application is loaded.

The reason is to simplify the mechanisms to maintain the Crypto Lock Settings shared memory buffer in a loaded state.

With this facility the NR2+ KeyStore NT service (declared as a Crypto application) started by PoLo will increment the reference count. The NT service contains a sleep action and will therefore maintain the buffer in memory permanently after PoLo exits. This would then make the NR2 PoloHold NT service redundant at NR2+.

Without this facility it is still proposed to withdraw the NR2 PoloHold NT service at NR2+ by incorporating the incrementing reference count code into the NR2+ KeyStore NT service.

3.3.3 Crypto Keys shared memory buffer

A new registry flag provides a means to force every Crypto application to reference the Crypto Keys shared memory buffer when the application is loaded.

The reason is to simplify the mechanisms to maintain the Crypto Keys shared memory buffer in a loaded state.

With this facility the NR2+ KeyStore NT service (declared as a Crypto application) started by PoLo will increment the reference count. The NT service contains a sleep action and will therefore maintain the buffer in memory permanently after PoLo exits. This would then make the System Management provided Crypto application waiter.exe and TivoliWait.dat redundant at NR2+. This clears PinICL 6211 which introduced waiter.exe to overcome the problem caused by the shared memory buffer getting unloaded.

Without this facility it is still proposed to make it unnecessary for System Management to run waiter.exe and also to clear PinICL 6211. This involves incorporating the incrementing reference count code from waiter.exe into the NR2+ KeyStore NT service.

3.3.4 Encrypted DLLs

TWC facilities do not handle transparently encrypted DLLs for Crypto applications. This restriction is documented in the TWC release notices with a workaround. This part of the enhancement requested the TW group to investigate introducing new facilities. If new facilities are introduced then they will need to be tested on behalf of other KM teams that will make use of them. L&G code includes a DLL which is to be stored in protected filestore.

Any new facilities will be tested as will the workaround if no new facilities are introduced.

3.4 Known Bugs

This section views the new TWC version (4.0-8) from the viewpoint of known bugs that affect PoLo. These can be classified into:

- those that exist in the previously used version that are understood to be cleared, and
- those that exist in the previously used version that are understood not to be cleared.

FUJ00078691

Both need to be checked in particular the second where the behaviour may have changed and workarounds may be affected.

3.4.1 TWC Cleared

	Area	Comments	TWC API affected
D1	File encryption status	see 3.2.5	GetCryptoStatus

Table 3-4 TWC cleared bugs

3.4.2 TWC Uncleared

None known.

3.4.3 PinICLs

PinICL 6211 see 3.3.3

FUJ00078691

4. TESTS

This section describes the tests to be carried out on the new TWC version (4.0-8). Note this does not include the TWC enhancements from section 3.3.

	Test description	Environment	Pre & post test checks	Views
T1	Regression test PoLo Rollout mode (encrypts directory files)	NR2 PoLo; new TWC; Riposte; SCO (generates exchangeable disk also used in T4)	Pre: TWC registry items (KeyFile missing and Permissions missing 'C:\Riposte' and 'F:\Ripostemirror') Post: TWC registry items (KeyFile present and Permissions contains 'C:\Riposte' and 'F:\Ripostemirror'); encryption status checks using PoLo normal reboot	3.1.1 table 3-1 A1, A2, A3 3.2.1 3.2.2 table 3-3 C1 3.2.5 3.4.1 table 3-4 D1
T2	Regression test PoLo normal reboot mode (unlocks directory files)	NR2 PoLo; new TWC; Riposte		3.1.1 table 3-1 A1, A2, A4 3.2.1 3.2.3 3.2.5 3.4.1 table 3-4 D1
T3	Regression test PoLo recovery reboot mode[3] (Lost PMMC or PIN)	NR2 PoLo; new TWC; Riposte; RecApp on PRS	Post: encryption status checks using PoLo normal reboot	3.1.1 table 3-1 A1, A2, A5 3.2.1 3.2.4 3.2.5 3.4.1 table 3-4 D1
T4	Test GetCryptoStatus() for correct	PoLo var1; new TWC; Riposte;		3.2.5

[3] This also tests out the full NR2 PoLo in Recovery mode which is required to support the migration of PoLo to NR2+.

FUJ00078691

	combination of output parameters PoLo var1 (see 5.2) displays GetCryptoStatus() parameter settings Run PoLo var1 Rollout & Reboot with wrong exchangeable disk	SCO with exchangeable disk (blank); second encrypted exchangeable disk (i.e. from T1)		3.4.1 table 3-4 D1
T5	Test SetCryptoInfo() for encrypting a file PoLo var1 encrypts directory on hard disk & encrypts file on exchangeable disk (both containing messagestores) Run PoLo var1 Rollout through to Riposte desktop allowing messagestores to synchronise	PoLo var1; new TWC; Riposte; SCO	Pre: TWC registry items (KeyFile missing and Permissions missing 'C:\Riposte' and anything starting with 'F:\Ripostemirror') Post: TWC registry items (KeyFile present and Permissions contains 'C:\Riposte' and 'F:\Ripostemirror\mirror.dat'); check messagestores synchronised using Rckms encryption status checks using PoLo normal reboot	3.2.2 table 3-3 C2
T6	Test crypto application rules update registry items; run PoLo normal reboot to unlock protected filestore; run PoloHold and waiter to keep buffers loaded; run application cryptocp (see 5.3) (clear to encrypted, encrypted to encrypted, encrypted to clear) (a form of cryptocp is used with the System Management compression activities)	NR2 PoLo; new TWC; Riposte; SCO; NR2 PoloHold and waiter; cryptocp	Post: check size of copied files; view header of encrypted files encryption status checks using PoLo normal reboot	3.1.2 B3

FUJ00078691

T7	Test shared memory buffers run PoLo normal reboot to unlock protected filestore; run PoloHold and waiter to keep buffers loaded; stop/close all other software with links to buffers; restart Riposte desktop software; login to update messagestore (comparable with System Management overnight activities) include failure case where memory buffers get unloaded	NR2 PoLo; new TWC; Riposte; NR2 PoloHold and waiter;	Post: check messagestore updated by login using Rckms encryption status checks using PoLo normal reboot	3.1.2 table 3-2 B1, B2 3.3.2 3.3.3
T8	Test Encrypted DLL run PoLo normal reboot to unlock protected filestore; run utility to open the encrypted DLL file before use for the first time as a DLL include failure case of using DLL before opened by utility	NR2 PoLo; new TWC; Riposte; dlltest1 and dlltest2		3.3.4
T9	Experiment to see if it is possible to upgrade TWC from 4.0-5 to 4.0-8 allowing Riposte to continue working on the old TWC until a reboot. The result may impact on the PMMC Agent migration strategy	NR2 PoLo; new & old TWC; Riposte		

Table 4-1 Test descriptions

FUJ00078691

The following table displays the TWC API and facilities used at NR2+ against the tests above.

	TWC API/facility	Tests
	OpenAcmContext()	T1, T2, T3
	OpenAcmContextEx() - new master key	T1
	OpenAcmContextEx() - existing master key	T2, T3
	CloseAcmContext()	T1, T2, T3
	GetPersonalKeyStatus() - no TWC keystore	T1
	GetPersonalKeyStatus() - TWC keystore exists	T2, T3
	GetCryptoStatus() - file unencrypted	T1, T4
	GetCryptoStatus() - file encrypted correct FEK	T2, T3, T4
	GetCryptoStatus() - file encrypted wrong FEK	T4
	CreateCryptoKey()	T1
	SetCryptoInfo() - declare directory; no encryption	T1
	SetCryptoInfo() - declare existing file & encrypt	T5
	I_StartTreeCryptTask()	T1
	ChangeCryptoKeyEx()	T3
	Crypto application	T6
	Crypto Lock Settings buffer - not unloaded	T7
	Crypto Keys buffer - not unloaded	T7
	Encrypted DLL	T8

Table 4-2 Matrix of TWC API/facility against tests

FUJ00078691

5. TOOLS

The tools used to check out this release of TWC are categorised into:

- existing software,
- software written for this release only and
- software written for this and possible future releases.

5.1 Existing Software Used

This section covers existing pieces of software used directly for testing. The software concerned is:

- NR2 PoLo for regression testing and migration checks
- Reset counter software
- RegEdit for checking registry values
- NR2 PoloHold to keep the Crypto Lock Settings shared memory buffer loaded
- NR2 waiter (System Management owned software) to keep the Crypto Keys shared memory buffer loaded
- Riposte services
- Rckms Riposte DOS utility which displays the counts of the node messages within the Riposte messagestore

5.2 Written Software Used - this Release only

This section covers those pieces of software used for testing on this occasion and not required at a later release. These items will not be stored in Visual Source Safe. The software concerned is:

- PoLo var1 - this is a build of NR2 PoLo with
  - additional screen messages displaying the values of the output parameters from GetCryptoStatus(). This checks that the bug in this API is cleared at this TWC release. The NR2+ PoLo makes use of this knowledge (see 3.2.5)
  - change the code paths for processing the second (exchangeable disk) directory to encrypt a file using SetCryptoInfo(). This checks API changes to be used by NR2+ PoLo (see 3.2.2 row C2 of table 3-3)

This software is not required after NR2+. All the functionality checked by PoLo var1 will exist in NR2+ PoLo.

5.3 Written Software Used - more than this Release

This section covers those pieces of software used for testing this release of TWC and that may be used for subsequent releases. These items will be stored in Visual Source Safe. The software concerned is:

- cryptocp - this is a piece of code which was provided by a TWC developer which is similar to code used as part of the System Management compression actions. It carries out a copy files operation run from a DOS prompt and takes two parameters the source and destination files. The program is constructed as a crypto application and allows the transparent handling of encrypted files without the use of caching which had caused problems.
- dlltest1 - opens for read the Microsoft Debug DLL software ”mfc42d.dll”; displays a message and sleeps. This crypto application together with utility dlltest2 tests the handling of an encrypted DLL file. Note ”mfc42d.dll” is not part of the NT loaded on PO counters.

FUJ00078691

- dlltest2 - a Windows application built in Microsoft Developers Studio in Debug mode. When run it will attempt to load the ”mfc42d.dll”. This utility together with dlltest1 tests the handling of an encrypted DLL file.

At NR2+ there will exist code to handle the encrypted L&G dll (not part of PMMC Agent) thus strictly eliminating the need to retain dlltest1 and dlltest2. However the code for handling the encrypted L&G dll may only get exercised properly at full integration time. To ensure that a subsequent release of TWC has not regressed in this area, the dlltest1 and dlltest2 software is retained to expose such problems at an earlier stage.