SERTIT-002-CR Certification Report

Issue 1.0 12. April 2005

Sospita License Protection QX Operating System, version 3.2

Certification Report - SERTIT standard report template Sd 009 version 0.3 29.10.2002

SERTIT, Postboks 14, N-1306 Bærum postterminal, NORWAY
Phone: +47 67 86 40 00 Fax: +47 67 86 40 09 E-mail: post@sertit.no Internet: www.sertit.no

Contents

Design Subsystems **24**

1 Certification Statement

Sospita License Protection QX Operating System is a micro-controller operating system that controls the execution of portions of a software application that is uploaded for execution on a token (e.g. a smart card or an USB-token).

Sospita License Protection QX Operating System version 3.2 has been evaluated under the terms of the Norwegian IT Security Certification Scheme and has met the specified Common Criteria Part 3 Evaluation Assurance Level EAL 3 requirements for the specified Common Criteria Part 2 conformant functionality when running on the platforms specified in Annex A.

Author	Arne H. Rage Certifier
Quality Assurance	Lars Borgos Quality Assurance
Approved	Kjell W. Bergan Head of SERTIT
Date approved	12. April 2005

2 Abbreviations

ACR	Access Control Rights
CC	Common Criteria
CCRA	Common Criteria Recognition Arrangement
CEM	Common Criteria Evaluation Methodology
EAL	Evaluation Assurance Level
EOR	Evaluation Observation Report
ETR	Evaluation Technical Report
EUL	End-User License. A Sospita license type that does not have the right to perform protection. An EUL is a mirror of a certain ML where both are have same identification values, except that the EUL only allows execution of code protected with the corresponding ML.
EVIT	Evaluation Facility under the Norwegian Certification Scheme for IT Security
EWP	Evaluation Work Plan
ML	Master License. A Sospita license type that allows the process of protection and execution source code. ML must be purchased from Sospita.
PIN	Personal Identification Number
POC	Point of Contact
PUK	Personal Unblocking Key
QP	Qualified Participant
QX	seQure eXecution, the Sospita token operating system, a component of SLP.
QXBlock	A sub-element of a QXCode. One QXBlock corresponds to one "function" call to QX.
QXCode	The collection of all QX virtual machine code which may be uploaded to and executed on the token. One QXCode is associated with exactly one license, and vice versa.
QXToken	A secure micro controller, external to the host, with CPU, memory, I/O interfaces and crypto capabilities, where the QXCode is executed.
SCP	Sospita Company Procedure
SDK	Sospita Development Kit, a component of SLP.
SERTIT	Norwegian Certification Authority for IT Security
SLM	Sospita License Manager, a component of SLP.
SLP	Sospita License Protection

SoF	Strength of Function
SPM	Security Policy Model
SRS	Sospita Runtime System
ST	Security Target
TL	Transport License. A Sospita license type that is required to transfer licenses between tokens.
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	TOE Security Policy

3 References

[1] Sospita License Protection The QX Operating System Security Target. Document number 1.2.2.2 of 11. September 2002.

[2] Sospita License Protection The QX Operating System Security Target. Document number 1.2.2.2 of 11. September 2002 PUBLIC.

[3] Common Criteria Part 1, CCIMB-99-031, Version 2.1, August 1999.

[4] Common Criteria Part 2, CCIMB-99-032, Version 2.1, August 1999.

[5] Common Criteria Part 3, CCIMB-99-033, Version 2.1, August 1999.

[6] Om sertifiseringsordningen, SD001, Version 2.3C, 23. August 2002.

[7] Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology, CEM-099/045, Version 1.0, August 1999.

[8] Int. 008 Evaluation of ST Introduction (ASE_INT.1) 31. July 2001.

[9] Int. 084 Evaluation of IT Security Requirements (ASE_REQ.1) 16. February 2001.

[10] R1-127 Evaluation of TOE Summary Specification (ASE_TSS.1) 29. October 2001.

[11] Int. 116 Evaluation of delivery (ADO_DEL.1) 31. July 2001.

[12] Int. 074 Evaluation of coverage (ATE_COV.2) Evaluation of depth (ATE_DPT.1) 15. October 2000.

[13] Int. 075 Evaluation of functional tests (ATE_FUN.1). Evaluation of independent testing (ATE_IND.2). 15. October 2000.

[14] Evaluation Technical Report S-1418/20.06 Version 1.1 of 2. October 2002.

[15] 1.4.7.1 Sospita License Protection 3.2 Developer's Manual of 16. August 2002.

[16] 1.8.7.1 Sospita License Protection 3.2 Administrator's Manual of 28. August 2002.

[17] Sospita License Protection Development Kit of October 2001.

[18] 1.2.6.1 Sospita License Protection QX Test Specification Integration Test Specification of 03. September 2002.

[19] 1.2.3.1 Sospita Licence Protection System Overview of 16. October 2001.

[20] 1.2.3.3 Sospita License Protection QX Security Manager System Design of 28. August 2002.

[21] 1.2.3.4 Sospita License Protection QX Virtual Machine System Design of 17. June 2002.

[22] 1.2.3.6 Sospita License Protection The QX Operating System Functional Specification of 09. July 2002.

[23] 0.1.7 SCP Software Release Procedure of 28. January 2002.

[24] 5.5 SCP Production Procedure – Logistics. Product version 3.0 of 19. November 2001.

[25] 1.4.8.1 Sospita License Protection White Paper of 12. July 2002.

4 Executive Summary

4.1 Introduction

This Certification Report states the outcome of the Common Criteria IT-security evaluation of Sospita License Protection QX Operating System version 3.2 to the Sponsor, Sospita, and is intended to assist prospective consumers when judging the suitability of the IT security of the product for their particular requirements.

Prospective consumers are advised to read this report in conjunction with the Public version of the Security Target [2], which specifies the functional, environmental and assurance evaluation requirements.

4.2 Evaluated Product

The version of the product evaluated was Sospita License Protection QX Operating System version 3.2.

This product is also described in this report as the Target of Evaluation (TOE). The developer was Sospita, Øksa 18, 4505 Mandal, Norway.

The TOE is a component of Sospita License Protection (SLP), a product that provides protection of software applications against unauthorised usage. The TOE is a micro-controller operating system that controls the execution of protected portions of a software application that are uploaded for execution on a hardware token (e.g. a smart card or a USB-token).

Details of the evaluated configuration, including the TOE's supporting guidance documentation, are given in Annex A.

An overview of the TOE's security architecture can be found in Annex B.

4.3 TOE scope

The scope of the TOE is limited to the QX operating system.

The following components are outside of the scope of the evaluation:

- Sospita Development Kit
- Sospita License Manager
- Sospita Runtime System

4.4 Protection Profile Conformance

The Security Target [1][2] did not claim conformance to any protection profile.

4.5 Assurance Level

The Security Target [1][2] specified the assurance requirements for the evaluation. Predefined evaluation assurance level EAL3 was used. Common Criteria Part 3[5] describes the scale of assurance given by predefined assurance levels EAL1 to EAL7. An overview of CC is given in CC Part 1[3].

4.6 Strength of Function

The minimum Strength of Function (SoF) was SoF-High. This was claimed for the TOE security functions listed in the ST [1][2] section 8.1.

The cryptographic mechanism contained in the TOE is publicly known (triple DES), and its appropriateness and strength is outside of the evaluation scope. Its implementation and operation is however within the scope of the evaluation.

4.7 Security Policy

There are no Organizational Security Policies or rules with which the TOE must comply.

4.8 Security Claims

The Security Target [1][2] fully specifies the TOE's security objectives, the threats, which these objectives meet, and security functional requirements and security functions to elaborate the objectives. All of the SFR's are taken from CC Part 2 [4]; use of this standard facilitates comparison with other evaluated products. An overview of CC is given in CC Part 1 [3].

4.9 Threats Countered

The threats that the TOE counter are as follows:

- An attacker may successfully modify a licensed software application or components of a licensed software application, resulting in execution of an application that they are not licensed to use.
- An attacker may successfully get security sensitive attributes of a license.
- An attacker may successfully modify non-writeable attributes of a license, causing the license to be illegally duplicated or extending the original limitations.
- An attacker may successfully execute a software application outside of the time limits imposed by the license without tampering with the license itself.
- An attacker may successfully execute a software application more times than allowed by the limits imposed by the license without tampering with the license itself.
- An attacker may successfully execute a software application with different ACR than allowed by the limits imposed by the license without tampering with the license itself.
- Information may leak between QX applications or between licenses and QX applications, allowing an attacker to breach license protection.
- An attacker may successfully illegally duplicate a license.
- An attacker may obtain a master license with an id reserved for somebody else. This impersonation can cause application errors if a user tries to execute

an application with the attacker's license or simplify unauthorized duplication if the attacker has access to cryptographic information used in original master license.

- A non-authorized user may break the lock protection of a PIN/PUK or password locked license or token, and perform unauthorized usage.
- An attacker can stress the physical limitations on write cycles of EEPROM and get a modification of bits in licenses.

4.10 Threats Countered by the TOE's environment

The following threat is to be countered by the TOE's environment:

- A sophisticated attacker possessing the necessary skills and appropriate hardware and software tools may perform hardware attacks in an attempt to modify or read security sensitive data in the TOE.

4.11 Threats and Attacks not Countered

- If a user by mistake setting the host's internal clock to a future time it can disable valid licenses. Setting the host's internal clock backwards can make a license last longer than it is expected to, if no start date is set as a constraint. The QX keeps track of last used time, so manually setting the clock back in time requires the user to keep manually track of the last used time.
- The License ID can be manipulated because the License ID is not protected by Sospita's QX code.

These vulnerabilities are described in the guidance documents [15], [16] and [17] and in the developer's vulnerability analysis.

4.12 Environmental Assumptions and Dependencies

The following assumptions are assumed to exist in the environment:

- The TOE must be installed on a secure hardware token that provides resistance against physical attacks appropriate to the threat environment.
- The QX application program interface [SRS] will provide security metrics for the authorizing mechanisms (i.e. PIN/PUK and password).
- The ability to generate master licenses must be limited to a master license generation facility.

4.13 IT Security Objectives

The IT security objectives in the ST [1][2] are as follows:

- One or more licenses control the execution of QX applications.
- An end-user must not be able to view security sensitive attributes of a license stored on a token, but may be able to view those that are not security sensitive.

- A user must not be able to use modified QX applications by which he/she is not the legal owner of, i.e. possesses a corresponding master license.
- The execution of a QX application on a token must not interfere with any other QX application or license.
- QX applications must not be readable from the host computer.
- Execution of a QX application must be permitted only within the time constraints of a license stored on a token.
- The number of executions of a QX application must be controlled in accordance with a counter of license stored on a token.
- The execution of QX applications must be controlled in accordance with the ACR field stored in licenses.
- The ability to securely (integrity and confidentiality protection) transfer a license between tokens must be provided.
- It must be possible to control the ability to propagate a license from one token to other tokens.
- A user wanting to use a PIN/PUK or password locked license or token must provide a PIN/PUK or password in order to be authorized.
- Modification of attributes of a license must be controlled.
- Mechanisms for detecting hardware faults because of imposed stress must be provided by the TOE.
- Mechanisms for providing a proof of ownership for a specific master license must exist.
- The TOE must be installed on a secure hardware token that provides resistance against physical attacks appropriate to the threat environment.
- The QX application program interface [SRS] will provide security metrics for the authorizing mechanisms (i.e. PIN/PUK and password).
- The ability to generate master licenses must be limited to a master license generation facility.

4.14 Non-IT Security Objectives

The non-IT security objectives in the ST [1][2] are met by procedural or administrative measures in the TOE's environment and are as follows:

- Protected software applications and the corresponding cryptographic information must be generated and stored in a secure manner, such that no sensitive data is disclosed or possible to modify for unauthorized persons.

4.15 Security Functional Requirements

The TOE provides security functions to satisfy the following Security Functional Requirements (SFRs). Iteration (the use of a component more than once with varying

operations) is indicated by use of (*n*) following the component designator, where *n* is the number of the iteration:

- Cryptographic key generation FCS_CKM.1
- Cryptographic operation FCS_COP.1 (1)
- Cryptographic operation FCS_COP.1 (2)
- Cryptographic operation FCS_COP.1 (3)
- Subset access control FDP_ACC.1 (1)
- Subset access control FDP_ACC.1 (2)
- Security attribute based access control FDP_ACF.1 (1)
- Security attribute based access control FDP_ACF.1 (2)
- Basic data authentication FDP_DAU.1
- Import of user data with security attributes FDP_ITC.2 (1)
- Import of user data with security attributes FDP_ITC.2 (2)
- Subset residual information protection FDP_RIP.1
- Stored data integrity monitoring FDP_SDI.1
- Basic data exchange confidentiality FDP_UCT.1 (1)
- Basic data exchange confidentiality FDP_UCT.1 (2)
- Data exchange integrity FDP_UIT.1 (1)
- Data exchange integrity FDP_UIT.1 (2)
- Authentication failures FIA_AFL.1 (1)
- Authentication failures FIA_AFL.1 (2)
- Authentication failures FIA_AFL.1 (3)
- Timing of authentication FIA_UAU.1
- Re-authenticating FIA_UAU.6
- Replay detection FPT_RPL.1
- Non-bypassability of the TSP FPT_RVM.1
- TSF domain separation FPT_SEP.1

The IT-environment is required to satisfy the following SFRs:

- Basic data authentication FDP_DAU.1
- Verification of secrets FIA_SOS.1
- Passive detection of physical attack FPT_PHP.1
- Resistance to physical attack FPT_PHP.3
- Reliable Time Stamps FPT_STM.1

4.16 Security Function Policy

The TOE has a licence access control security function policy defined in FDP_ACC.1 (1), FDP_ACF.1 (1), FDP_ITC.2 (1), FDP_UCT.1 (1) and FDP_UIT.1 (1).

The license access control SFP is enforced on:

- The following subjects: User
- The following objects: Master license, Server license, End-user license, Transport license.
- The following operations: View a license, Generate a request for a master license, Generate a transport license, Move a license between tokens, Derive a specific number of end-user licenses, Merge end-user and server licenses, Backup writable licenses, Edit writable license attributes, Lock and unlock a license, Delete a license.

The TOE has a QX application access control security function policy defined in FDP_ACC.1 (2), FDP_ACF.1 (2), FDP_ITC.2 (2), FDP_UCT.1 (2) and FDP_UIT.1 (2).

The QX application access control SFP is enforced on:

- The following subjects: User.
- The following objects: QX application.
- The following operations: Upload a QX application, Encrypt and decrypt a QX application, Execute a QX application.

4.17 Evaluation Conduct

The evaluation was carried out in accordance with the requirements of the Norwegian Certification Scheme as described in SERTIT Document SD001[6]. The Norwegian Certification Authority for IT Security (SERTIT) manages the Scheme.

The purpose of the evaluation was to provide assurance about the effectiveness of the TOE in meeting its Security Target [1][2], which prospective consumers are advised to read. To ensure that the Security Target [1][2] gave an appropriate baseline for a CC evaluation, it was first itself evaluated. The TOE was then evaluated against this baseline. Both parts of the evaluation were performed in accordance with CC Part 3 [5] and the Common Evaluation Methodology (CEM) [7]. Interpretations used are [8], [9], [10], [11], [12], [13] and [14] listed in the references section.

SERTIT monitored the evaluation, which was carried out by the Secode Norge[1] IT-Security Evaluation Facility (ITSEF/EVIT). The evaluation was completed when the EVIT submitted the Evaluation Technical Report (ETR) [14] to SERTIT in 02.10.2002. SERTIT then produced this Certification Report.

[1] At the time of the evaluation, Secode Norway AS was called System Sikkerhet ASA

4.18 General Points

The evaluation addressed the security functionality claimed in the Security Target [1][2] with reference to the assumed operating environment specified by the Security Target [1][2]. The evaluated configuration was that specified in Annex A. Prospective consumers are advised to check that this matches their identified requirements and give due consideration to the recommendations and caveats of this report.

Certification does not guarantee that the IT product is free from security vulnerabilities. This Certification Report and the belonging Certificate only reflect the view of SERTIT at the time of certification. It is furthermore the responsibility of users (both existing and prospective) to check whether any security vulnerabilities have been discovered since the date shown in this report. This Certification Report is not an endorsement of the IT product by SERTIT or any other organization that recognizes or gives effect to this Certification Report, and no warranty of the IT product by SERTIT or any other organization that recognizes or gives effect to this Certification Report is either expressed or implied.

5 Evaluation Findings

5.1 Introduction

The Evaluators examined the following assurance classes and components taken from CC Part 3 [5]. These classes comprise the EAL3 assurance package.

Assurance Class	Assurance Components	
Configuration management	ACM_CAP.3	Authorization Controls
	ACM_SCP.1	TOE CM coverage
Delivery and operation	ADO_DEL.1	Delivery Procedures
	ADO_IGS.1	Installation, generation and start-up procedures
Development	ADV_FSP.1	Informal Functional Specification
	ADV_HLD.2	Security enforcing high-level design
	ADV_RCR.1	Informal correspondence demonstration
Guidance documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life cycle support	ALC_DVS.1	Identification of security measures
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing – sample
Vulnerability assessment	AVA_MSU.1	Examination of Guidance
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis

All assurance classes were found to be satisfactory and were awarded an overall "pass" verdict.

5.2 Delivery

The delivery procedures are described in the Sospita Company Procedure (SCP) documents [23] and [24] listed in Section 3 "References".

Upon receipt of the TOE, the consumer is recommended to check that the evaluated version has been supplied, and to check that the security of the TOE has not been compromised in delivery.

5.3 Installation and Guidance Documentation

The SLP Developer's Manual [15], the SLP Administrator's Manual [16] and the SLP Development Kit [17] (which are release notes and readme file) provide guidance and details of installation of the TOE in the evaluated configuration.

The main document to read for installation in the evaluated configuration is the SLP Administrator's Manual[16].

5.4 Misuse

Users should follow the guidance in the Administrator's Manual [16] in order to ensure that the TOE operates in a secure manner.

5.5 Vulnerability Analysis

The evaluators were satisfied that the developer's vulnerability analysis describes all obvious vulnerabilities and that it gives a rationale for why they are / are not exploitable in the intended environment for the TOE.

5.6 Developer's Tests

Sospita's testing is mostly based on automatic tests performed by a test program (software), which is developed by Sospita. The test procedures of this test program are documented in SLP QX Test Specification Integration Test Specification [18].

The test program is installed on a PC, which has the necessary smart card readers and USB interfaces. Two unused tokens (Philips P16Wx064 or Atmel AT90SC6464C PCS, smart card or USB interface) with the QX OS version 3.2 and a STL (Sospita Transport License) are inserted in the connections. The PC is connected to the network at Sospita's location in Mandal, which has the facility of transferring a Sospita master license.

Automatic tests requirements:

- Software:
  - Windows 2000
  - QXApi32.dll
  - Testrunner.exe, version 3.2
  - 0QXApiTest.dll, version 3.2, build 3.2.6
  - QxSystemTest.exe, version 3.2
- Hardware with Master license generation server via TCP/IP:
  - 2 QX tokens with Philips hardware (and the necessary readers).
  - 2 QX tokens with Atmel hardware (and the necessary readers).
- Hardware without Master license generation server via TCP/IP:

- Prerequisite: Initial access to a Master license server via TCP/IP for preloading of necessary master license to one token.
- 3 QX tokens with Philips hardware (and the necessary readers), where one token holds all the necessary master licenses.
- 3 QX tokens with Atmel hardware (and the necessary readers), where one token holds all the necessary master licenses.

Manual test requirements for License constraints, Execution and Block integrity:

- Development tools (Sospita Development Kit, version 3.2 [17])
- 1 Master license
- 2 tokens with the necessary readers
- Additional for tests in section 13: An editor

Pre-coded test applications are available.

Developer's testing approach

The developer has thoroughly tested all security functions, which are addressable via the smart card or USB interface. Tests are performed both for negative and positive results. All security functions except F14, F26 and F25 have been tested with different test procedures.

The Security functions:

F14 One QX application on a token is not able to interfere with the execution path of any other QX application on the token, nor use memory areas used for licenses. When a memory area used by one QX application is allocated to another, this memory is cleared before the allocation.

F26 The TOE shall detect if integrity errors of licenses and memory range of QX applications stored on tokens occur

are internal QX functions and have been tested during the development process.

The security function:

F25 The TOE shall detect write failures in EEPROM when license are uploaded

is executed every time a new license is generated and is thereby verified.

Amount of testing performed

Approximately 150 different tests are performed. These tests covers the functions specified in document SLP QX Operating System Functional Specification [22], except for the security functions listed above.

Testing results

The evaluators have received test reports both for the automatic and manual tests from the developer. The test reports include both expected and actual results and

have been generated by the automatic test programs. For the manual tests, test reports with individual test results have been generated. The evaluators have witnessed both the automatic and manually performed tests during the independent testing. All tests have been successfully performed with the expected results.

5.7 Evaluators' Tests

The evaluation team decided to test all security functions related to the user of a protected application (end-user), which are:

F16: The following attributes of licenses in transfer are protected from disclosure:

- i The symmetric encryption key
- ii The initialisation vector
- iii Lock specifications

F17: End-user and server licenses stored on QX tokens are non-writeable.

F21: A lock can be assigned to a license or token to ensure that only authenticated users can use this license or token. The user must provide a PIN or password in order to unlock a license or token and the unlocked modus is only active in a continuous session.

F22: After 3 successive authentication failures a PIN-locked license or token will be blocked until a correct PUK is provided. Wrong PUK can be given 6 times. After 6 successive PUK failures, the license or token will be permanently blocked.

F23: After 10 successive authentication failures a password-locked license or token will be permanently blocked.

By examining the developers test specification [18] the evaluation team discovered that the developer has thoroughly tested all security function related to the user of a protected application, and therefore decided to witness these tests at the developer's site.

Most of the developers testing are performed automatically, and the evaluation team decided to witness the tests described in the developers test specification [18]. The testing was performed at Sospita in Mandal and at Secode Norge[2] in August 2002.

All test results were according to the expected results in the developers test specification [18].

[2] At the time of the evaluation, Secode Norway AS was called System Sikkerhet ASA

6 Evaluation Outcome

6.1 Certification Result

After due consideration of the ETR [14], produced by the Evaluators, and the conduct of the evaluation, as witnessed by the Certifier, SERTIT has determined that Sospita License Protection QX Operating System version 3.2 running on the hardware tokens Philips P16WX064, Atmel AT90SC6464C and Atmel AT90SC6464C-USB micro controllers meets the specified Common Criteria Part 3 conformant requirements of Evaluation Assurance Level EAL 3 for the specified Common Criteria Part 2 conformant functionality in the specified environment, when running on platforms specified in Annex A.

The minimum Strength of Function was SoF-High.

6.2 Recommendations

Prospective consumers of Sospita License Protection QX Operating System version 3.2 should understand the specific scope of the certification by reading this report in conjunction with the Security Target [1][2]. The TOE should be used in accordance with environmental considerations as specified in the Security Target [1][2].

Only the evaluated TOE configuration should be installed. This is specified in Annex A with further relevant information given above under Section 4.3 "TOE Scope" and Section 5 "Evaluation Findings".

The TOE should be used in accordance with the supporting guidance documentation included in the evaluated configuration.

Annex A: Evaluated Configuration

TOE Identification

The TOE consists of:

The Sospita License Protection QX Operating System version 3.2.

TOE Documentation

The supporting guidance documents evaluated were:

[a] SLP Developer's Manual [15]

[b] SLP Administrator's Manual [16]

[c] SLP Development Kit [17]

Further discussion of the supporting guidance material is given in Section 5.3 "Installation and Guidance Documentation"

TOE Configuration

The following configuration was used for testing:

The test program for automatic tests on the TOE is installed on a PC, which has the necessary smart card readers and USB interfaces. Two unused tokens (**Philips P16Wx064** or **Atmel AT90SC6464C** PCS, smart card or USB interface) with the QX OS version 3.2 and a STL (Sospita Transport License) are inserted in the connections. The PC is connected to the network at Sospitas location in Mandal, which has the facility of transferring a Sospita master license.

Automatic tests requirements:

- Software:
  - Windows 2000
  - QXApi32.dll
  - Testrunner.exe, version 3.2
  - QXApiTest.dll, version 3.2, build 3.2.6
  - QxSystemTest.exe, version 3.2
- Hardware
  - With Master license generation server via TCP/IP
    - 2 QX tokens with Philips hardware (and the necessary readers)
    - 2 QX tokens with Atmel hardware (and the necessary readers)
  - Without Master license generation server via TCP/IP
    - Prerequisite: Initial access to a Master license server via TCP/IP for preloading of necessary master license to one token

- 3 QX tokens with Philips hardware (and the necessary readers), where one token holds all the necessary master licenses
- 3 QX tokens with Atmel hardware (and the necessary readers), where one token holds all the necessary master licenses

Manual test requirements for section 8, License constraints, section 12, Execution, and section 13, Block integ:

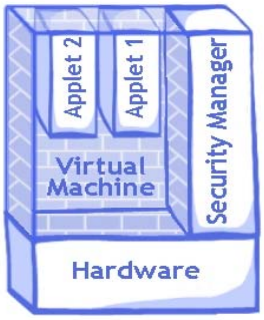
- Development tools (Sospita Development Kit, version 3.2)
- 1 Master license
- 2 tokens with the necessary readers
- Additional for tests in section 13: An editor

Pre-coded test applications are available.

Annex B: Product Security Architecture

This annex gives an overview of the main product architectural features that are relevant to the security of the TOE. Other details of the scope of evaluation are given in the main body of the report and in Annex A.

Architectural Features



The figure above shows the two main (high level) components of the QX operating system, the security manager and the virtual machine. Documents [25], [20] and [21] and give an overview of QX, describe the QX security manager and the QX virtual machine respectively.

The security manager performs license management operations (e.g. license upload/download, changing license attributes and so on) and runtime license enforcement. The virtual machine handles execution of QX applications, enforcing non-interference between different QX applications (i.e. fire-wall mechanisms) and runtime memory bounds verification.

Design Subsystems

The QX Virtual Machine

As the QX Virtual Machine supports 32 bits instructions, independently of the hardware platform, most PC application code using data types supported by QX can be executed with equal precision in QX. QX is also highly dynamic by allowing for automatic reallocation of memory areas that must be used by a specific QX application. In this way, an application with many big fragments can be executed by dynamically uploading the wanted fragment and reallocate memory for this particular block. The virtual machine employs a RAM stack for internal function calls. The stack size for QX version 3.2 is typically some 700 bytes, depending on underlying hardware characteristics.

Multiple QX applications can be active ("selected" in the Java card terminology) simultaneously. Calls to QX applications are queued on the host, and executed one at the time on the card. An "unlimited" number of QX applications can be executed simultaneously. (The actual limit is 128 applications, the size of the license table, where each license is associated with one QX application.) In QX, EEPROM memory is not a limitation: QX offers a dynamic uploading of QX applications, which means that QX applications are swapped in and out of the QX operating system. This mechanism

also applies to a single QX application, which means that the QX application size may be greater than the EEPROM size.

The QX Security Manager

The QX Security Manager is concerned with the following functions:

- Controlling all I/O to/from host.
- License generation, storage, transfer (out of and in to token) and deletion.
- Enforcing license restrictions during runtime, only allowing applications with a valid license to be executed.
- Memory management, keeping information about open connections, licenses and applications stored on token and garbage collection (including object reuse, i.e. re-initialising memory returned to the OS).