Security Target
for

3.1

from

GmbH & Co. KG

Date	**14.09.2004**
Version No.:	**1.7**
Author:	**HOB GmbH & Co. KG**

Table of Contents

Revisions to Document

Version	Date	Changes
0.1	28.11.2003	First Draft
0.2	13.12.2003	Second Draft
1.0	22.01.2004	First published version
1.1	23.01.2004	Minor correction: Corrected WSP version number on page 10, corrected document date on first page
1.2	05.03.2004	Changes after Evaluation Report 1
1.3	05.04.2004	Changes after Review-Protocol 01 / 02.04.2004
1.4	09.07.2004	Updated OS platforms in chapter 2.3.2
1.5	05.08.2004	Updated OS platforms, added version numbering details, corrected chapter 5.1.1, replaced term “Java SSL classes” by “SSL Client Classes”
1.6	19.08.2004	Corrected 5.1.1 (FDP_ITT.EX.1, common names check) and adapted chapter 6.1.1, minor corrections in 8.2.2 and 8.2.3
1.7	14.09.2004	Changed Safari version to 1.2 and product documentation (manual) sub-version to 0405. Clarified AGD_USR.1 assurance requirement (chapter 6.2). General review of chapters 2.3.2, 2.3.3 and 2.3.4

1 Security Target Introduction

1.1 Security Target, TOE, and CC Identification

ST identification: Security Target for HOBLink Secure 3.1, Version 1.7

Keywords: SSL, TLS

TOE identification: HOBLink Secure 3.1

CC identification: CC Version 2.1

1.2 Conventions, Terminology, and Acronyms

This section identifies the formatting conventions used to convey additional information, specific terminology and acronyms used throughout the remainder of the document.

1.2.1 Conventions

This section describes the conventions used in chapter 5 to denote CC operations on security requirements. The CC allows several operations to be performed on functional requirements; *assignment*, *iteration*, *refinement*, and *selection* are defined in paragraph 2.1.4 of Part 2 of the CC.

- The selection operation is used to select one or more options provided by the CC in making a statement. Selections are denoted by *underlined italicised text*.
- The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. An assignment is indicated by showing the value in square brackets [assignment value(s)].
- The refinement operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text** and ~~strikethrough~~.
- Iteration of a component is used when a component is repeated more than once with varying operations. Iterated components are given unique identifiers by appending specific indicators (like appending " - AES" or " – RSA" to "FCS_COP.1 Cryptographic operations").

1.2.2 Terminology

Application	HOB connectivity software product that uses the TOE to transfer user data from one computer to another.
Attacker	An unauthorised user who attempts to violate the TSP.
SSL/TLS	A common denominator for the SSL protocol as specified in [SSL] and the TLS protocol as specified in [TLS].
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
User Data	Data created by and for the user, that does not affect the operation of the TSF. User data which is transferred over physically separated parts of the TOE according to the TSP is referred to as "transmitted information".
TOE Security Policy	A set of rules that regulate how assets are managed, protected and distributed within a TOE.

1.2.3 Acronyms

The following abbreviations are used in this Security Target:

AES	Advanced Encryption Standard
CM	Configuration Management
EAL	Evaluation Assurance Level
IP	Internet Protocol
LAN	Local Area Network
MAC	Message Authentication Code
MD	Message Digest
MRJ	Macintosh Runtime Java
OSP	Operational Security Policy
PKI	Public Key Infrastructure
PP	Protection Profile
RDP	Remote Desktop Protocol
RSA	Rivest, Shamir, Adleman
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SSL	Secure Socket Layer
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TLS	Transport Layer Security
TSP	TOE Security Policy
TSS	TOE Summary Specification
WSP	WebSecureProxy

1.3 Security Target Overview

HOBLink Secure is a software package for integration of SSL/TLS capability into HOBLink software products. This means that HOBLink Secure adds the capability of creating certificates along with the ability to bring up and use SSL/TLS connections to HOB software such as Terminal Server clients, emulations or data base drivers.

1.4 Common Criteria Conformance

The TOE is

- Part 2 extended,
- Part 3 conformant.

2 TOE Description

This section provides a product description in order to point out its purpose and possible fields of application. Furthermore, the scope of the evaluated configuration is defined.

2.1 Product Type

HOBLink Secure is a software package for integration of SSL/TLS capability into HOBLink software products (referred to as "application") such as HOBLink JWT, J-Term or DRDA. It was designed to build a secure communication environment based on three components:

- An administrative tool called "Security Manager" for the generation of configuration files that are required by the two other components.
- A gateway called "WebSecureProxy" located in front of the destination server (referred to as "WSP").
- A client module called "Java SSL classes", which work together with an application.

2.2 Intended Method of Use

HOBLink Secure consists of three core components that work together as follows:

Before any SSL/TLS connection can be established, the administrator uses the Security Manager to create two sets of configuration files, one for the SSL Client Classes and one for the WSP. Each set of files (referred to as "Security Unit") consists of three files: The configuration file, the certificate database file and the password file.

The Security Units are manually and securely distributed to the WSP and the client computer which will be using the SSL Client Classes.

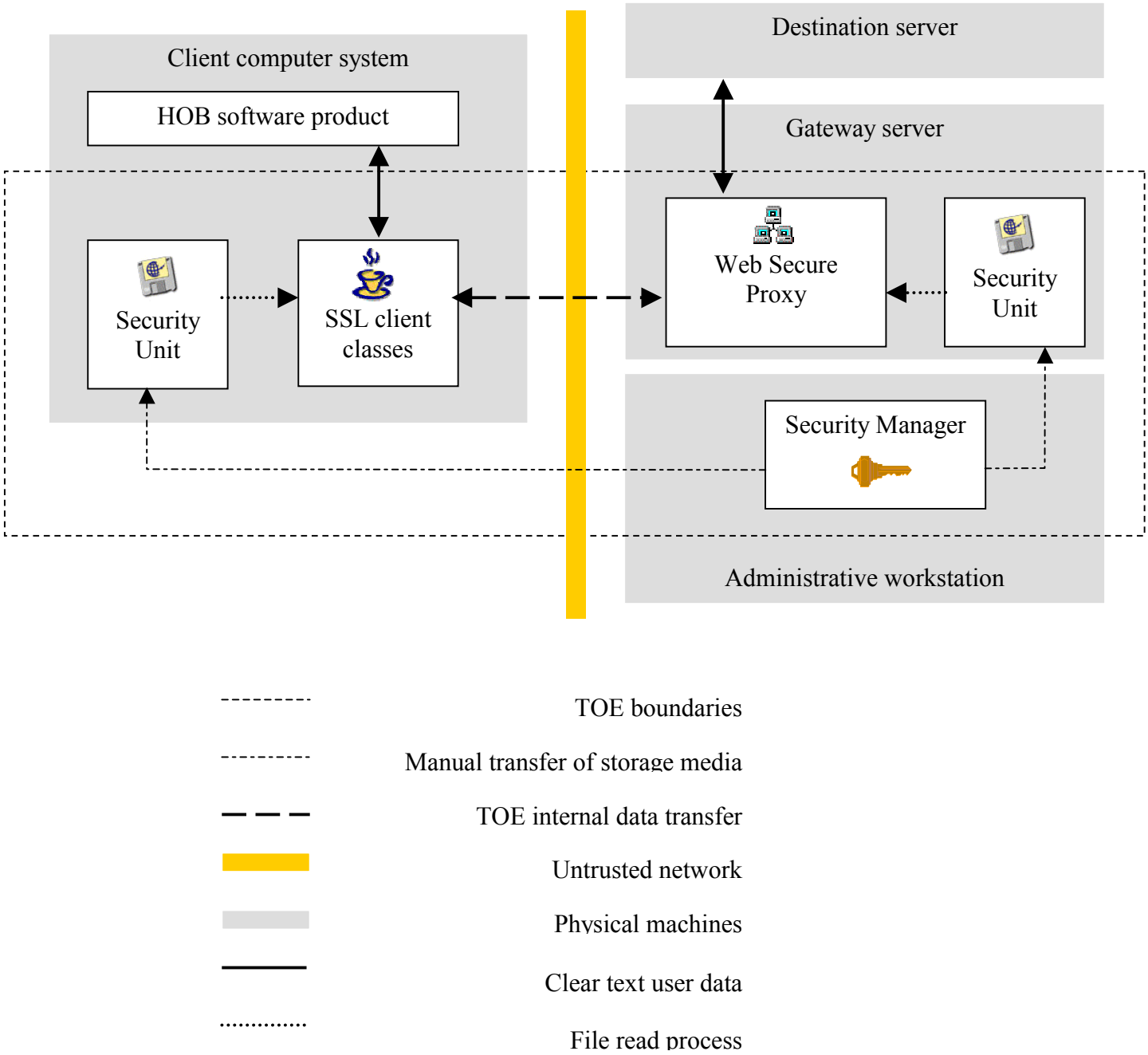
The SSL Client Classes are called as soon as an application (for example HOBLink JWT) initiates an SSL/TLS protected connection. The SSL classes read their local Security Unit and initiate an SSL/TLS handshake with the WSP.

The WSP reads its configuration data and the respective Security Units during start up.

When there is an incoming connection request, both parties go through the SSL/TLS handshake procedure and in case of success, the WSP establishes a connection to the destination server system previously defined in the gateway definition file of the WSP.

The following figure visualises the systems and components involved, the physical boundary of the TOE, and shows relevant communication paths.

Figure 1: TOE and TOE Environment Overview



2.3 Operational Environment

2.3.1 Hardware Requirements

SSL Client Classes (clients):

Intel Pentium 200 MHz or CPU with equivalent processing speed

8 Mbytes of RAM available

WebSecureProxy (gateway):

Intel Pentium III 500 MHz or CPU with equivalent processing speed

64 Mbytes of RAM available

20 Mbytes of non-volatile storage space

Security Manager (administration workstation):

Intel Pentium II 350 MHz or CPU with equivalent processing speed

64 Mbytes of RAM available

40 Mbytes of non-volatile storage space

2.3.2 Operating Systems

SSL Client Classes (clients):

MS Windows 98SE, NT 4.0 Workstation SP6a, XP Pro, 2000 Pro, 2003

Apple Mac OS 10.3.x

SuSE Linux 8.2, 9.1 (with graphical subsystem installed)

WebSecureProxy (gateway):

HP UX 11i

IBM AIX 5.1

SUN Solaris 9

Security Manager (administration workstation):

MS Windows XP Pro, 2000 Pro, 2003

Apple MacOS, 10.3.x

SuSE Linux-8.2, 9.1 (with graphical subsystem installed)

2.3.3 Java Virtual Machines

Every client has to provide a Java Virtual Machine (JVM). The JVM has to be present stand-alone (for locally installed HOB software) or integrated in a browser (for HOB software in applet-mode).
Most JVMs published since 2000 are suitable, but the use of the following versions depending on the operating systems is recommended:

Table 1: Recommended Java Virtual Machines for the TOE

Operating system	JVM (local)	Version
MS Windows	SUN MS jview	1.3.1_07 5.00.3167
Apple MacOS X	SUN SUN	1.3.1 1.4.2
Linux	SUN IBM	1.4.1_02 1.3.1

2.3.4 Browsers

Most web browsers that are able to run Java applets are suitable for use with the SSL Client Classes, but the use of the following software is recommended.

Table 2: Recommended web browsers for the SSL Client Classes

Operating system	Browser	Version
MS Windows	MS Internet Explorer MS Internet Explorer Netscape Netscape Mozilla	5.5 (not on Win XP, 2003) 6 SP1 4.77 (not on Win 2003) 7.1 1.5
Linux	Netscape Mozilla	4.77 1.2.1
Apple MacOS X	MS Internet Explorer Mozilla Safari	5.2.2 1.5 1.2

2.4 Security Services

HOBLink Secure provides two security services that are considered in this document:

- The first service (referred to as “Certificate Generation”) is provided by a utility for the generation of certificates and keys for usage in SSL connections. This software is called “SecurityManager”.
- The second service (referred to as “SSL/TLS Protocol Function”) is provided by a set of software components that is able to establish and transfer user data over an SSL connection. This set of components consists of the “SSL Client Classes” and the “WebSecureProxy”.

Along with these components, HOB provides a number of tools that fulfil the requirements of specific HOB software products, simplify the certificate distribution or analyse the set up. This software does not belong to the TOE and therefore will not be discussed further. In detail, the following programs are not a part of the TOE and do not provide security services relevant to the context of this document:

- HOBLink Certificate Generator 3.1
- HOBLink Secure Java Tools
- HOBLink SSL for Windows

2.5 Scope and Boundaries of the Evaluated Configuration

Figure 1 shows the physical and logical boundaries of the TOE. The following tables provide a short description of the software components and the respective versions for the evaluation.

Table 3: Software Components of the TOE

SSL Client Classes	Add-on module for application, has to be installed or downloaded on the client side.
WebSecureProxy	Multi-functional gateway that translates the encrypted data stream from the public side of the network into “clear-text” communication for the internal LAN and vice versa.

Security Manager	Offline PKI utility designed to create, import, export and maintain X.509v3 certificates and SSL configurations required for the operation of the two other components.

Table 4: Scope of delivery of HOBLink Secure 3.1

Component name	**Version**	**Alternate equivalent versions**	
HOBLink Secure (whole product)	3.1	040810	
SSL Client Classes	Version 01.20(9.0)	01.20 040723	
		SSL Version 1 Revision 20 Release 9.0	
		SSL Version 1.20, 23.07.2004	
WebSecureProxy	Version 2.1	On SUN Solaris systems:	2.1 Jun 21 2004
		On HP-UX (IA 64) systems	2.1-pre-02 Aug 4 2004
		On HP-UX (PA RISC) systems:	2.1 Jun 16 2004
		On IBM AIX systems:	2.1 Jun 21 2004
Security Manager	Version 3.1-00.50	3.1 0050	
Manual	Version 3.1-0406	-none-	

Additional software (tools) that is on the product CD **(not part of the TOE):**

- HOBLink Certificate Generator
- HOBLink Secure Java Tools
- HOBLink SSL for Windows

3 TOE Security Environment

In order to clarify the nature of the security problem that the TOE is intended to solve, this section describes the following:

- Any known or assumed threats to the assets against which protection within the TOE or its environment is required.
- Any organisational security policy statements or rules with which the TOE must comply.
- Any assumptions about the security aspects of the environment and/or of the manner for which the TOE is intended.

3.1 Threats

This section describes the threats to be addressed by the TOE and the threats to be addressed by the operational environment of the TOE.

3.1.1 Threats Addressed by the TOE

The **assets** to be protected are user data transmitted between physically separated parts of the TOE over an untrusted network.

The threat **agents** are attackers who have permanent access to the untrusted network and who have the ability and skill to monitor, modify, delete, re-play or re-order transmitted information or to insert information into the transmitted information. Attackers are assumed to have a knowledge of publicly known vulnerabilities and have access to commonly available tools.

The following table identifies the threats which are addressed by the TOE:

Table 5: Threats to be countered by the TOE

Name	Description
T.Untrusted-Path	An attacker may attempt to disclose, modify, delete, re-play, re-order or insert user data by monitoring, modifying, deleting, re-playing or re-ordering the information transmitted over the untrusted network or by inserting additional information in the transmitted information in an unnoticeable manner.

3.1.2 Threats Addressed by the Operational Environment

There are no threats to be addressed by the operational environment.

3.2 Organisational Security Policies

The following table identifies the organisational security policy which has to be met by the TOE:

Table 6: Policies to be met by the TOE

Name	Description
P.Certificates	The TOE must have the ability to generate certificates and the corresponding keys for its own use.

Name	Description
P.Authenticate	The TOE must enforce mutual authentication of the SSL Client Classes component and the WSP component.

3.3 Assumptions

The following table identifies the assumptions about the intended usage of the TOE and about the environment of use of the TOE:

Table 7: Assumptions

Name	Description
A.Administrators	Administrators are trustworthy, competent and follow all administrator guidance.
A.Users	Authorised users are trustworthy and follow all user guidance.
A.Malicious	All systems shall be free of malicious software such as viruses, trojan horses, worms or spyware.
A.Access	Access to the TOE and to the corresponding systems is limited to authorised persons by appropriate technical, physical and organisational means.
A.SecMgr	The Security Manager has to be installed on a separate machine that is not physically connected to any network and the Security Units generated by this tool are transferred securely between the TOE components.
A.DestroyRSA	RSA keys are securely destroyed when they are no longer needed.
A.Time	The underlying operating system provides reliable time information to the TOE.

4 Security Objectives

This section identifies the security objectives for the TOE and for the TOE environment.

4.1 Security Objectives for the TOE

The following table identifies the security objectives to address security concerns that are directly addressed by the TOE:

Table 8: Security Objectives for the TOE

Name	Description
OT.SSL/TLS	The TOE must provide functionality to protect user data against disclosure, modification, deletion, re-playing, re-ordering or insertion of additional data by the procedures specified in the SSL/TLS standard. This protection will be applied for all user data transmitted between physically separated parts of the TOE.
OT.Certificates	The TOE must provide functionality to generate certificates and the corresponding keys for its own use.
OT.Authenticate	The TOE must enforce mutual authentication of the SSL Client Classes component and the WSP component.

4.2 Security Objectives for the Environment

The following table identifies security objectives to address security concerns that are addressed by TOE environment:

Table 9: Security Objectives for the Environment

Name	Description
OE.Administrators	Those responsible for the TOE must assign trustworthy and competent personnel to the administration of the TOE who follow all administrator guidance.
OE.Users	Those responsible for the TOE must use it in an environment where authorised users are trustworthy and follow all user guidance.
OE.Malicious	Those responsible for the TOE must assure that all IT-systems used for the TOE shall be free of malicious software such as viruses, trojan horses, worms or spyware.
OE.Access	Those responsible for the TOE must assure that access to the TOE and to the corresponding IT-systems is limited to authorised persons by appropriate technical, physical and organisational means.
OE.SecMgr	Those responsible for the TOE must assure that the Security Manager is installed on a separate machine that is not physically connected to any network and that the Security Units generated by this tool are transferred securely between the TOE components.
OE.DestroyRSA	Those responsible for the TOE must assure that RSA keys are destroyed when they are no longer needed.

Name	Description
OE.Time	The underlying operating system will provide reliable time information to the TOE.

5 IT Security Requirements

This section identifies the security functional requirements for the TOE and its environment and the security assurance requirements for the TOE.

5.1 TOE Security Requirements

This section identifies the security functional requirements and the security assurance requirements for the TOE.

5.1.1 TOE Security Functional Requirements

The following table identifies the selected TOE security functional requirements. All components except the explicitly stated component FDP_ITT.EX.1 are drawn from Part 2 of the CC.

Table 10: Security Functional Requirements Overview

Component	Component Name
FCS_CKM.1	Cryptographic key generation
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1	Cryptographic operation
FDP_ITT.1	Basic internal transfer protection
FDP_ITT.3	Integrity monitoring
FIA_SOS.2	TSF Generation of secrets
FDP_ITT.EX.1	HOB SSL/TLS Policy

Family FCS_CKM Cryptographic key management

FCS_CKM.1 Cryptographic key generation - AES

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [specified in [SSL]/[TLS]] and specified cryptographic key sizes [128 Bits] that meet the following: [none].

FCS_CKM.1 Cryptographic key generation - RSA

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [random numbers proven to be prime by [RAM]] and specified **modulus length** [1536 Bits] that meet the following: [none].

FCS_CKM.4 Cryptographic key destruction - AES

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [overwriting with zeros] that meets the following: [none].

Family FCS_COP Cryptographic operation

FCS_COP.1 Cryptographic operation - AES

FCS_COP.1.1 The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES] and cryptographic key sizes [128 Bits] that meet the following: [FIPS PUB 197].

FCS_COP.1 Cryptographic operation – MD5

FCS_COP.1.1 The TSF shall perform [generation of hash values] in accordance with a specified cryptographic algorithm [MD5] and **message digests of** [128 Bits] that meet the following: [RFC 1321].

FCS_COP.1 Cryptographic operation - RSA

FCS_COP.1.1 The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [RSA] and **modulus length** [1536 Bits] that meet the following: [RSA].

FCS_COP.1 Cryptographic operation – SHA-1

FCS_COP.1.1 The TSF shall perform [generation of hash values] in accordance with a specified cryptographic algorithm [SHA-1] and **message digests of** [160 Bits] that meet the following: [FIPS PUB 180-1].

Family FDP_ITT Internal TOE transfer

FDP_ITT.1 Basic internal transfer protection

FDP_ITT.1.1 The TSF shall enforce the [HOB SSL/TLS Policy] to prevent the *disclosure*, *modification* of user data when it is transmitted between physically-separated parts of the TOE.

FDP_ITT.3 Integrity monitoring

FDP_ITT.3.1 The TSF shall enforce the [HOB SSL/TLS Policy] to monitor user data transmitted between physically-separated parts of the TOE for the following errors: [modification of data, reordering of data, deletion of data, insertion of data, replay of data].

FDP_ITT.3.2 Upon detection of a data integrity error, the TSF shall [terminate the SSL/TLS session and inform the application about the reason of the termination].

Family FIA_SOS Specification of secrets

FIA_SOS.2 TSF Generation of secrets

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet [FIPS 140-1 requirements on random number generation].

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for [SSL/TLS handshake protocol].

Explicitly Stated Requirements

FDP_ITT.EX.1 HOB SSL/TLS Policy

FDP_ITT.EX.1.1 The TSF shall enforce the HOB SSL/TLS Policy for all user data to be transferred between physically separated parts of the TSF.

FDP_ITT.EX.1.2 User data shall be transferred only after an SSL/TLS handshake has been successfully completed. All of the following conditions have to be met:

- SSL Client Classes and WSP present an X.509v3 certificate according to [X509] to each other. Each party checks the following items in both certificates (their own and the one received from the counterpart):
  - The digital signature of the certificate is valid.
  - The validity period has begun and did not end yet.
- In addition to that, the SSL Client Classes check if the transmitted common name of the server certificate matches the name stored in their list of trusted certificates.

This requirement is not hierarchical to other components.

Dependencies:		
	FCS_COP.1	Cryptographic operations
	FCS_CKM.1	Cryptographic key generation
	FIA_SOS.2	TSF Generation of secrets
	FPT_STM.1	Reliable Time Stamps

5.1.2 TOE Security Assurance Requirements

The following table identifies the selected TOE security assurance requirements. All components are drawn from Part 3 of the CC. The selected components represent assurance level EAL2.

Table 11: Security Assurance Requirements Overview

Component Identification	Component Name
ACM_CAP.2	Configuration items
ADO_DEL.1	Delivery procedures
ADO_IGS.1	Installation, generation, and start-up procedures
ADV_FSP.1	Informal functional specification
ADV_HLD.1	Descriptive high-level design
ADV_RCR.1	Informal correspondence demonstration
AGD_ADM.1	Administrator Guidance
AGD_USR.1	User guidance
ATE_COV.1	Evidence of coverage
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing - sample
AVA_SOF.1	Strength of TOE security function evaluation
AVA_VLA.1	Developer vulnerability analysis

ACM_CAP.2 Configuration items

Developer action elements:

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2C The TOE shall be labelled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.
The configuration list shall uniquely identify all configuration items that comprise the TOE.[1]

ACM_CAP.2.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.6C The CM system shall uniquely identify all configuration items.

Evaluator action elements:

ACM_CAP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

[1] This element is added as a result of Interpretation 003

ADO_DEL.1 Delivery Procedures

Developer action elements:

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

Evaluator action elements:

ADO_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1 Installation, generation, and start-up procedures

Developer action elements:

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation, and start-up of the TOE.[2]

Evaluator action elements:

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

ADV_FSP.1 Informal functional specification

Developer action elements:

ADV_FSP.1.1D The developer shall provide a functional specification.

Content and presentation of evidence elements:

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

Evaluator action elements:

ADV_FSP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.1 Descriptive high-level design

Developer action elements:

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

[2] This element is changed as a result of Interpretation 051

Content and presentation of evidence elements:

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

Evaluator action elements:

ADV_HLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_ADM.1 Administrator guidance

Developer action elements:

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

Content and presentation of evidence elements:

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_USR.1 User guidance

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_COV.1 Evidence of coverage

Developer action elements:

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

Content and presentation of evidence elements:

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

Evaluator action elements:

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent testing - sample

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

AVA_SOF.1 Strength of TOE security function evaluation

Developer action elements:

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim, the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim, the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

AVA_VLA.1 Developer vulnerability analysis

Developer action elements:

AVA_VLA.1.1D The developer shall perform a vulnerability analysis.[3]

AVA_VLA.1.2D The developer shall provide vulnerability analysis documentation.[3]

Content and presentation of evidence elements:

AVA_VLA.1.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.[4]

AVA_VLA.1.2C The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.[4]

AVA_VLA.1.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.[4]

Evaluator action elements:

AVA_VLA.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

5.2 Security Requirements for the IT Environment

Family FPT_STM Time stamps

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The **environment** shall be able to **provide reliable time stamps**.

[3] This element is changed as a result of Interpretation 051

[4] This element is replaced as a result of Interpretation 051

6 TOE Summary Specification

This section describes the TOE security functions and the TOE security measures.

6.1 TOE Security Functions

This section describes the TOE security functions.

6.1.1 SSL/TLS Protocol Function

HOBLink Secure 3.1 implements the SSL/TLS protocol as specified in [SSL] or [TLS], respectively. The product does not use third party classes to provide this functionality. The SSL Client Classes component of HOBLink Secure 3.1 reflect the “client” as specified in SSL/TLS and the WebSecureProxy (WSP) component of HOBLink Secure 3.1 implements the “server” as specified in SSL/TLS.

General characteristics of the SSL/TLS protocol: The first phase within the SSL/TLS protocol is the handshake protocol, in which a cryptographic cipher suite (consisting of an asymmetric algorithm, a bulk data encryption algorithm, key size for the bulk data encryption algorithm, a hash algorithm) and cryptographic keys (encryption/decryption keys, MAC secrets) are negotiated. Separate session keys (used for bulk data encryption) and MAC secrets are generated for each communication direction. The handshake protocol uses SHA-1 and MD5 to protect the integrity of the information exchanged. After the handshake protocol has been successfully completed, user data can be securely transferred according to the agreed cipher suite. The SSL/TLS protocol ensures the confidentiality and integrity of transmitted user data. A proper implementation of the SSL/TLS protocol allows to detect modification of data, substitution of data, reordering of data, deletion of data, insertion of data, and replay of data.

Characteristics of HOBLink Secure 3.1: In the course of handling of the SSL/TLS handshake protocol, the SSL Client Classes and the WebSecureProxy (WSP) negotiate on (RSA, AES 128 bits, SHA-1) as cipher suite and generate AES session keys of 128 bits length and MAC secrets to be used with SHA-1. User data is transferred only after an SSL/TLS handshake has been successfully completed. All of the following conditions have to be met:

- SSL Client Classes and WSP present an X.509v3 certificate according to [X509] to each other. Each party checks the following items in both certificates (their own and the one received from the counterpart):
  - The digital signature of the certificate is valid.
  - The validity period has begun and did not end yet.
- In addition to that, the SSL Client Classes check if the transmitted common name of the server certificate matches the name stored in their list of trusted certificates.

The random data generated for the SSL/TLS handshake protocol fulfil the requirements on random number generation specified in FIPS PUB 140-1. Sensitive information (session keys, MAC secrets, SSL/TLS pre_master_secret, SSL/TLS master_secret) is erased in storage when the corresponding sessions are terminated.

This security function is performed by permutational or probabilistic algorithms.

6.1.2 Certificate Generation

General characteristics of the SSL/TLS protocol: The SSL/TLS protocol allows to authenticate clients and servers. The related standards do not cover the generation of the certificates needed for authentication.

Characteristics of HOBLink Secure 3.1: HOBLink Secure 3.1 enforces mutual authentication of the SSL Client Classes component and the WebSecureProxy component of HOBLink Secure 3.1. For this purpose the Security Manager component of HOBLink Secure 3.1 can generate RSA keys and issue X.509v3 certificates according to [X509] for these keys with digital signatures based on RSA and SHA-1. In the certified configuration the Security Manager generates RSA keys with a modulus size of 1536 bits according to random numbers proven to be prime by [RAM].

This security function is also performed by permutational or probabilistic algorithms.

6.2 Assurance Measures

The following table describes how the assurance requirements of EAL2 are met by the TOE.

Table 12: Security Measures

Assurance Component	Corresponding Assurance Measures
ACM_CAP.2	HOB uses a CM system. The CM system uniquely identifies all configuration items. The CM documentation describes the method used to uniquely identify the configuration items.
ADO_DEL.1	The delivery procedures and technical measures implemented by HOB allow to detect discrepancies between the developer's master copy and the version received by a customer.
ADV_FSP.1	HOB provides an FSP document which describes the TSF and its externally visible interfaces: For each interface, it describes the purpose and method of use, and provides details of effects, error messages and exceptions.
ADV_HLD.1	HOB provides an HLD document which describes the TSF in terms of subsystems and the functionality provided by each subsystem. The HLD document identifies any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software. The HLD document identifies all interfaces to subsystems of the TSF and identifies those which are externally visible.
ADV_RCR.1	The FSP document contains a section which demonstrates that the functional specification is a correct and complete representation of the TOE security functions as specified in the TSS. The HLD document contains a section which demonstrates that the high level design is a correct and complete representation of the functional specification.
AGD_ADM.1 ADO_IGS.1	HOB provides an Administrator Reference Guide which describes how the TOE is installed, operated and administered in a secure manner.
AGD_USR.1	The administrator guidance includes a chapter that is intended to be printed separately and distributed as a leaflet to the end users. It provides instructions and guidelines for the secure use of the TOE. HOB provides an internal guidance document which describes how the security functions of the TOE can be used by the programmer of an application.

Assurance Component	Corresponding Assurance Measures
ATE_COV.1 ATE_FUN.1	HOB provides test documentation for the TOE. The test documentation consists of test plans, test procedure descriptions, expected test results and actual test results. The test plans identify the security functions to be tested and describe the goal of the tests to be performed. The test procedure descriptions identify the tests to be performed and describe the scenarios for testing each security function. The expected test results show the anticipated outputs from a successful execution of the tests. This test documentation includes an analysis of the test coverage against the functional specification.
ATE_IND.2	HOB provides to the evaluators all the required resources to perform their own tests and to repeat developer tests.
AVA_SOF.1	No strength of function claims were made.
AVA_VLA.1	HOB provides a vulnerability analysis that addresses obvious vulnerabilities that could be exploited by an attacker attempting to violate the TSP.

6.3 Strength of Function Claims

HOBLink Secure relies on probabilistic permutational algorithms (as required by FCS_CKM.1 - RSA, FCS_CKM.1 - AES, FCS_COP.1 - RSA, FCS_COP.1 - AES, FCS_COP.1 – SHA-1, FCS_COP.1 – MD5 and FIA_SOS.2.) to perform cryptographic operations and to generate secrets that ensure proper function of its encryption.

Cryptographic mechanisms are used for both security services provided by the TOE. However, in accordance with the requirements of the national scheme no strength of function claim is made for the cryptographic mechanisms.

The results of this random number generation are also used for both security services provided by the TOE. However, there is no strength of function claim made for this generator and hence no SOF claim can be made for the product itself.

The minimum strength of function level for HOBLink Secure is therefore: “not applicable”.

7 PP Claims

There are no Protection Profile Claims.

8 Rationale

This section provides the security objectives rationale, the security requirements rationale, the TOE summary specification rationale and the PP claims rationale.

8.1 Security Objectives Rationale

The purpose of this rationale is to demonstrate that the identified security objectives are:

- suitable, i.e. they are sufficient to address the security needs
- necessary, i.e. there are no redundant objectives

8.1.1 Security Objectives Coverage

The purpose of this rationale is to demonstrate that:

- all identified threats, organisational security policies and assumptions are addressed,
- there are no redundant objectives

Table 13: Security Objectives Coverage

Threats - Policies – Assumptions / Security Objectives	**OT.SSL/TLS**	**OT.Certificates**	**OT.Authenticate**	**OE.Administrators**	**OE.Users**	**OE.Malicious**	**OE.Access**	**OE.SecMgr**	**OE.DestroyRSA**	**OE.Time**
T.Untrusted-Path	X									
P.Certificates		X								
P.Authenticate			X							
A.Administrators				X						
A.Users					X					
A.Malicious						X				
A.Access							X			
A.SecMgr								X		
A.DestroyRSA									X	
A.Time										X

8.1.2 Security Objectives Sufficiency

The purpose of this rationale is to demonstrate that:

- all identified threats are countered,
- all organisational security policies are covered,
- all assumptions are properly addressed.

T.Untrusted-Path addresses the threat, that an attacker may attempt to disclose, modify, delete, re-play, re-order or insert user data by monitoring, modifying, deleting, re-playing or re-ordering the information transmitted over the untrusted network or by inserting additional information in the transmitted information. Objective **OT.SSL/TLS** ensures that the TOE uses the procedures specified in the SSL/TLS standard to protect user data against disclosure, modification, deletion, re-playing, re-ordering or insertion of additional information whenever user data is transmitted between physically separated parts of the TOE. These procedures are known to be appropriate to protect the confidentiality and integrity of user data transmitted over an untrusted network and are thus – if properly implemented – adequate to completely counter that threat.

P.Certificates is completely covered by objective **OT.Certificates**, because OT.Certificates mandates what P.Certificates specifies.

P.Authenticate is completely covered by objective **OT.Authenticate**, because OT.Authenticate mandates what P.Authenticate specifies.

A.Administrators is completely addressed by objective **OE.Administrators**, because OE.Administrators mandates what A.Administrators specifies.

A.Users is completely addressed by objective **OE.Users**, because OE.Users mandates that the TOE is only used in an environment in which A.Users is valid.

A.Malicious is completely addressed by objective **OE.Malicious**, because OE.Malicious mandates what A.Malicious specifies.

A.Access is completely addressed by objective **OE.Access**, because OE.Access mandates what A.Access specifies.

A.SecMgr is completely addressed by objective **OE.SecMgr**, because OE.SecMgr mandates what A.SecMgr specifies.

A.Time is completely addressed by objective **OE.Time**, because OE.Time mandates what A.Time specifies.

8.2 Security Requirements Rationale

The purpose of this rationale is to demonstrate that the identified security requirements are suitable and necessary to meet the security objectives:

- suitable, i.e. they are sufficient to meet the security objectives
- necessary, i.e. there are no redundant security requirements

8.2.1 Traceability of Functional Requirements

The purpose of this rationale is to demonstrate that:

- all objectives are addressed,
- there are no redundant security requirements

Table 14: TOE Security Requirements Coverage

Security Objectives / Functional Security Requirements	**FCS_CKM.1 - AES**	**FCS_CKM.1 - RSA**	**FCS_CKM.4 - AES**	**FCS_COP.1 - AES**	**FCS_COP.1 - MD5**	**FCS_COP.1 - RSA**	**FCS_COP.1 - SHA-1**	**FDP_ITT.1**	**FDP_ITT.3**	**FIA_SOS.2**	**FDP_ITT.EX.1**
OT.SSL/TLS	X		X	X	X	X	X	X	X	X	X
OT.Certificates		X				X	X			X	
OT.Authenticate											X

Security requirement FPT_STM.1 for the environment traces back to OE.TIME. It is a requirement which has to be resolved by the IT environment of the TOE.

8.2.2 Functional Requirements Sufficiency

The purpose of this rationale is to demonstrate that the requirements are adequate to meet all security objectives.

OT.SSL/TLS: Security functional requirements **FCS_CKM.1 – AES**, **FCS_CKM.4 – AES**, **FCS_COP.1 - AES**, **FCS_COP.1 - MD5**, **FCS_COP.1 – SHA-1**, **FCS_COP.1 – RSA**, **FDP_ITT.1**, **FDP_ITT.3**, **FIA_SOS.2** and **FDP_ITT.EX.1** together are appropriate to meet this objective because they require a restrictive and secure implementation of the SSL/TLS protocol. All requirements regarding security functionality which can be deducted from SSL/TLS are reflected in the functional requirements for the TOE and its environment:

- **FCS_CKM.1 - AES** contributes to this objective by requiring the generation of session keys for AES according to the procedures for session key generation specified in SSL/TLS.
- **FCS_CKM.4 - AES** contributes to this objective by requiring the erasure of session keys as soon as they are no longer needed.

- **FCS_COP.1 - AES** contributes to this objective by requiring to provide the AES algorithm. A symmetric encryption/decryption algorithm like AES is needed to perform SSL/TLS bulk data encryption of user data.
- **FCS_COP.1 – MD5** contributes to this objective by restating a requirement from the SSL/TLS standard.
- **FCS_COP.1 – SHA-1** contributes to this objective by restating a requirement from the SSL/TLS standard.
- **FCS_COP.1 - RSA** contributes to this objective by requiring to provide the RSA algorithm. An asymmetric encryption/decryption algorithm like RSA is needed to perform the SSL/TLS handshake protocol.
- **FDP_ITT.1** contributes to this objective by identifying the security function policy which is to be implemented to protect confidentiality and integrity of user data transferred between physically separated parts of the TOE over an untrusted network.
- **FDP_ITT.3** contributes to this objective by specifying the integrity errors which have to be detected and the reaction of the TSF in case an integrity error has been detected.
- **FIA_SOS.2** contributes to this objective by requiring to generate random numbers of sufficient quality which are used in the SSL/TLS handshake.
- **FDP_ITT.EX.1** contributes to this objective by mandating the use of SSL/TLS and by clearly specifying the conditions for a successful handshake.

OT.Certificates: Security functional requirements **FCS_CKM.1 – RSA**, **FCS_COP.1 – SHA-1**, **FCS_COP.1 – RSA,** and **FIA_SOS.2** together are appropriate to meet this objective because they require to provide all security functionality needed to generate certificates and the corresponding keys for its own use:

- **FCS_CKM.1 - RSA** contributes to this objective by mandating that appropriate asymmetric key pairs can be generated.
- **FCS_COP.1 – SHA-1** contributes to this objective by requiring to provide the SHA-1 algorithm. A cryptographic hash function like SHA-1 is needed to generate digital signatures.
- **FCS_COP.1 – RSA** contributes to this objective by requiring to provide the RSA algorithm. An asymmetric algorithm encryption algorithm like RSA is needed to generate digital signatures.
- **FIA_SOS.2** contributes to this objective by requiring to generate random numbers of sufficient quality which are used in the RSA key generation.

OT.Authenticate: Security functional requirement **FDP_ITT.EX.1** is appropriate to meet this objective because this security function policy requests mutual authentication of the SSL Client Classes component and the WSP component (which is an option if the SSL/TLS standard).

OE.Time: Security functional requirement **FPT_STM.1** for the IT environment is appropriate to meet this objective for the IT environment because it requires exactly what this objective calls for.

8.2.3 Explicitly Stated Requirements Rationale

Objective OT.SSL/TLS introduces the need to adhere to the SSL/TLS standard. The related "HOB SSL/TLS policy" is neither an access control policy nor an information flow control policy. Hence the existing SFRs like from the FDP class (like FDP_ACF.1 or FDP_IFC.1) are not applicable. Since no other components for modelling policies are provided in CC part 2, the ST author has found it appropriate to explicitly specify component FDP_ITT.EX.1 for that purpose. This explicitly specified functional component neither explicitly nor implicitly introduces any additional assurance requirement.

8.2.4 Rationale for Assurance Requirements

The evaluation assurance level EAL2 has been selected to give a potential customer a basic assurance that an independent third party evaluation following internationally accepted criteria has been performed.
EAL2 provides assurance by an analysis of the security functions, using a functional and an interface specification, guidance documentation and the high-level design of the TOE, to understand the security behaviour. The analysis is supported by independent testing of the TOE security functions, evidence of developer testing based on the functional specification, selective independent confirmation of the developer test results, strength of function analysis, and evidence of a developer search for obvious vulnerabilities (e.g. those in the public domain).
This is considered to be appropriate considering an attack potential of attackers who have a knowledge of publicly known vulnerabilities and have access to commonly available tools.

8.2.5 Rationale for Strength of Function Claims

No SOF-claim is made. This section does therefore not apply.

8.2.6 Mutually Supportive Security Requirements

The purpose of this rationale is to demonstrate that all dependencies are satisfied, or why specific requirements are not relevant.

Table 15: Security Functional Requirements Dependencies

SFR	Dependencies	resolved
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution *or* FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure Security attributes	**Yes**, see note 1 **Partially**, see note 2a **No**, see note 3
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes *or* FCS_CKM.1 Cryptographic key generation] FMT_MSA.2 Secure Security attributes	**Yes,** see note 4a **No**, see note 3
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes *or* FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure Security attributes	**Partially,** see note 4b **Partially**, see note 2b **No**, see note 3

SFR	Dependencies	resolved
FDP_ITT.1	[FDP_ACC.1 Subset Access Control *or* FDP_IFC.1 Subset Information Flow Control]	**No**, see note 5
FDP_ITT.3	[FDP_ACC.1 Subset Access Control *or* FDP_IFC.1 Subset Information Flow Control] FDP_ITT.1 Basic internal transfer protection	**No**, see note 5 **Yes**
FIA_SOS.2	(no dependencies)	**Yes**, implicitly
FDP_ITT.EX.1	FCS_COP.1 Cryptographic operations FCS_CKM.1 Cryptographic key generation FIA_SOS.2 TSF Generation of secrets FPT_STM.1 Reliable Time Stamps	**Yes**, see note 6

Note 1: SFR FCS_COP.1 has been selected in four iterations (FCS_COP.1 –AES, FCS_COP.1 - RSA, FCS_COP.1 – SHA-1, FCS_COP.1 – MD5).

Note 2a: This dependency is resolved for FCS_CKM.1 – AES. It is not resolved for FCS_CKM.1 – RSA, because destruction of RSA keys has to be assured by organisational measures (see OE.DestroyRSA). There is no need for the TOE to support corresponding functionality.

Note 2b: This dependency is resolved for FCS_COP.1 – AES. It is not resolved for FCS_COP.1 – RSA, because destruction of RSA keys has to be assured by organisational measures (see OE.DestroyRSA). There is no need for the TOE to support corresponding functionality. It is not applicable to FCS_COP.1 – SHA-1 and FCS_COP.1 – MD5 because no secret keys are used to generate hash values.

Note 3: The dependency is not relevant in the context of the TOE. The enforcement of the security attributes relevant for generation of keys and other cryptographic secrets (algorithms, key length) are under control of a person who is granted access to the Security Manager (administrator). Access control is enforced by technical, physical and organisational measures outside of the TOE (see OE.Access).

Note 4a: SFR FCS_CKM.1 has been selected in two iterations (FCS_CKM.1 – AES, FCS_CKM.1 - RSA).

Note 4b: This dependency is resolved for FCS_COP.1 – AES and FCS_COP.1 – RSA. It is not applicable to FCS_COP.1 – SHA-1 and FCS_COP.1 – MD5 because no secret keys are used to generate hash values.

Note 5: SFR FDP_ITT.1 requires the enforcement of the "HOB SSL/TLS" SFP to protect the transfer of user data between physically separated parts of the TOE. The "HOB SSL/TLS" SFP specified in FDP_ITT.EX.1 is neither an access control SFP nor an information control SFP. The dependency to FDP_ACC.1 respectively FDP_IFC.1 is therefore not relevant in the context of the TOE.

Note 6: SFR FDP_ITT.EX.1 and the underlying SSL/TLS protocol introduce functional requirements on symmetric key operations (satisfied by FCS_COP.1 – AES), asymmetric key operations (satisfied by FCS_COP.1 - RSA), a hashing algorithm (satisfied by FCS_COP.1 - SHA), corresponding key generation mechanisms (satisfied by FCS_CKM.1 – AES and FCS_CKM.1 – RSA), random numbers (satisfied by FIA_SOS.2) and reliable timestamps (satisfied by the environment through FPT_STM.1).

8.3 TOE Summary Specification Rationale

This section provides a mapping between TOE security functions and security functional requirements for the TOE and a mapping between TOE security measures and security assurance requirements for the TOE.

8.3.1 Mapping between TOE Security Functions and SFRs

The purpose of this rationale is to demonstrate that the identified security functions are suitable and necessary to fulfil the security requirements:

- suitable, i.e. they are sufficient to fulfil the security requirements
- necessary, i.e. there are no redundant security functions

Table 16: Security Functions Coverage

Security Function	Functional Security requirement (SFR)
SSL/TLS Protocol Function	FCS_CKM.1 – AES, FCS_CKM.4 – AES, FCS_COP.1 – AES, FCS_COP.1 - MD5, FCS_COP.1 - RSA, FCS_COP.1 – SHA-1, FDP_ITT.1, FDP_ITT.3, FIA_SOS.2, FDP_ITT.EX.1
Certificate Generation	FCS_CKM.1 – RSA, FCS_COP.1 - RSA, FCS_COP.1 – SHA-1, FIA_SOS.2

Table 17: Correspondence of SFRs to Security Functions

SFR	SSL/TLS Protocol Function	Certificate Generation
FCS_CKM.1 – AES	X	
FCS_CKM.1 – RSA		X
FCS_CKM.4 – AES	X	
FCS_COP.1 - AES	X	
FCS_COP.1 – MD5	X	
FCS_COP.1 – RSA	X	X
FCS_COP.1 – SHA-1	X	X
FDP_ITT.1	X	
FDP_ITT.3	X	
FIA_SOS.2	X	X
FDP_ITT.EX.1	X	

SFR **FCS_CKM.1 – AES** requires that AES cryptographic keys of 128 bits length are generated according to the procedure specified for generation of session keys in SSL/TLS. Such functionality is provided by the security function **SSL/TLS Protocol Function** for generating session keys for bulk data encryption when handling the handshake protocol.

SFR **FCS_CKM.1 – RSA** requires that RSA keys with a modulus size of 1536 bits according to [random numbers proven to be prime by [RAM]] length are generated. Such functionality is provided by the security function **Certificate Generation** for generating certificates for the SSL Client Classes and the WebSecureProxy (WSP).

SFR **FCS_CKM.4 – AES** requires that AES keys are destroyed by overwriting with zeroes. Such functionality is provided by the security function **SSL/TLS Protocol Function**. This security function erases the AES session keys and other sensitive data (MAC secrets, SSL/TLS pre_master_secret, SSL/TLS master_secret) in storage when the corresponding SSL/TLS session has been terminated.

SFR **FCS_COP.1 – AES** requires that AES encryption/decryption using keys with a key length of 128 bits is used. Such functionality is provided by the security function **SSL/TLS Protocol Function**. It uses AES for bulk data encryption.

SFR **FCS_COP.1 – MD5** requires that the message digests according to MD5 are used. Such functionality is provided by the security function **SSL/TLS Protocol Function**. It uses MD5 to generate message authentication codes to protect the authentication of data exchanged in the SSL/TLS handshake protocol. The same messages are also protected by message authentication codes based on SHA-1.

SFR **FCS_COP.1 – RSA** requires that RSA encryption/decryption using keys with a modulus size of 1536 bits is used. Such functionality is provided by the security function **SSL/TLS Protocol Function**. It uses RSA encryption/decryption in a phase of the SSL/TLS handshake protocol to protect the confidentiality of information and to verify certificates. The encryption functionality is also provided by the security function **Certificate Generation**. It uses RSA to generate X.509v3 certificates for keys it has generated.

SFR **FCS_COP.1 – SHA-1** requires that the message digests according to SHA-1 are used. Such functionality is provided by the security function **SSL/TLS Protocol Function**. It uses SHA-1 to generate message authentication codes to protect the authenticity of data exchanged in the SSL/TLS handshake protocol. The same messages are also protected by message authentication codes based on MD5. Such functionality is also provided by the security function **Certificate Generation**. It uses SHA-1 to generate X.509v3 certificates for keys it has generated.

SFR **FDP_ITT.1** requires that user data is protected against disclosure or modification when it is transmitted between physically-separated parts of the TOE. Such functionality is provided by the security function **SSL/TLS Protocol Function** that implements the SSL/TLS protocol.

SFR **FDP_ITT.3** requires that the TSF monitor user data transmitted between physically-separated parts of the TOE and detect modification of data, reordering of data, deletion of data, insertion of data, replay of data. Upon detection of a data integrity error, the TSF have to terminate the SSL/TLS session and inform the application about the reason of the termination. Such functionality is fully provided by the security function **SSL/TLS Protocol Function**.

SFR **FIA_SOS.2 – SSL** requires that the TSF shall provide a mechanism to generate secrets that meet FIPS 140-1 requirements on random number generation and that the use of these TSF generated secrets is enforced for the SSL/TLS handshake protocol and for RSA key generation. Such functionality is provided by the security functions **SSL/TLS Protocol Function** and **Certificate Generation.** The random numbers generated fulfil the test requirements specified in FIPS PUB 140-1. They are used by the security function **SSL/TLS Protocol Function** during the handling of the SSL/TLS handshake protocol to generate the protocol elements ServerHallo.random, ClientHallo.random and pre_master_secret and by the security function **Certificate Generation** to generate RSA keys.

SFR **FDP_ITT.EX.1** requires that the TSF shall enforce the HOB SSL/TLS Policy for all user data to be transferred between physically separated parts of the TSF. User data is transferred only after an SSL/TLS handshake has been successfully completed. SFR **FDP_ITT.EX.1** specifies the requirements under which a handshake is considered to be successful. This policy is imple-

mented by security function **SSL/TLS Protocol Function** which handles the SSL/TLS protocol. It prevents user data from being transferred during a session unless a successful handshake has been successfully completed. It strictly obeys the conditions specified in the HOB SSL/TLS Policy.

8.3.2 Mapping between Security Measures and Assurance Requirements

The table in section 6.2 demonstrates that all assurance requirements are addressed by adequate assurance measures.

8.4 PP Claims Rationale

This Security Target does not claim conformance to any Protection Profile.

Annex Related Standards and Documents

[AES] FIPS PUB 197, National Institute of Standards and Technology, Advanced Encryption Standard, November 26, 2001

[CC] Common Methodology for Information Technology Security Evaluation, CCIMB-99-031, Version 2.1, August 1999, Part 1 to 3

[CEM] Common Criteria for Information Technology Security Evaluation, CEM-99/045, Part 2 – Evaluation Methodology, Version 1.0, 1999

[MD5] RFC 1321, Rivest, The MD5 Message-Digest Algorithm, January 1992

[RAM] A.J. Menezes, Handbook of applied cryptography, p.139, p.165, 1996 (CRC Press LLC)

[RSA] R. Rivest, A. Shamir, and L. Adleman, On Digital Signatures and Public-Key Cryptosystems, MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212, January 1979

[SHA-1] FIPS PUB 180-1, National Institute of Standards and Technology, Secure Hash Standard, April 17, 1995

[SSL] Freier, Karlton, Kocher, The SSL Protocol Version 3.0, November 18, 1996

[TLS] RFC 2246, Diercks, Allen, The TLS Protocol Version 1.0, January 1996

[X509] Housley, R., Ford, W., Polk, W. and D. Solo, Internet Public Key Infrastructure: Part I: X.509 Certificate and CRL Profile, RFC 2459, January 1999