BALTIMORE
www.baltimore.com

Security Target for Baltimore Timestamp Server version 2.0.2 Patch 1

Release	2.0.21
Status	Release
Date	3rd Oct 2003

Table of Contents

1. Introduction

1.1 Security Target Identification

1.1.1 This section provides the labelling and descriptive information necessary to control and identify the Security Target and the TOE to which it refers.

1.1.2 It is assumed that the reader of this document is familiar with the concept of PKI and timestamps [RFC3161].

Title:	Security Target for Baltimore Timestamp Server version 2.0.2 Patch 1 version 2.0.2l
Authors:	CMG Admiral / Baltimore Technologies
ISO 15408 (CC) Version:	2.1 Final
EAL:	3
ST Evaluation:	AISEP, CMG Admiral AISEF
TOE Description	Baltimore Technologies, Timestamp Server 2.0.2 Patch 1
Keywords:	Timestamp, Public Key Infrastructure

Table 1.1 ST Information

1.2 Security Target Scope

1.2.1 Baltimore's Timestamp Server is a PKI/Cryptography standards compliant server for generating digital timestamps in response to requests received from remote clients. It uses Public Key technology to digitally sign the timestamp tokens as evidence that a specified data item existed at a particular point in time.

1.2.2 The Timestamp Server implements software cryptographic functions that are used for digitally signing timestamp tokens, preserving the integrity of the server's configuration data and events log, and for identifying and authenticating TOE Administrators. The server can be configured to use hardware cryptography implemented by third party PKCS#11 compliant devices, the device must keep the signing key secure, and provide SHA-1 and RSA/DSA signing for the Timestamp Server to use. These products must be assured to at least CC EAL3 or equivalent. The following cryptographic algorithms are included in the evaluation: Signature Generation and Verification using RSA (1024 and 2048 bit), DSA (1024 bit) and SHA-1.

1.2.3 The Timestamp Server supports two key storage formats: PKCS#12 and PKCS#11. All keys and certificates used with the server must be generated externally and conform to the requirements set by the National Authority.

1.2.4 The Timestamp Server will be evaluated with using internal software cryptographic functions. Timestamp Server will be tested with hardware cryptographic functions provided by Baltimore's Sureware Keyper version 2.1 Professional.

1.2.5 The Timestamp Server runs on Windows NT Server version 4.0. The interaction between the Timestamp Server and a client application conforms to the Internet X.509 Public Key Infrastructure Time Stamp Protocol [RFC3161] using a socket interface. No IT requirements have been specified for the reliability or accuracy of the time source; it is the responsibility of the Timestamp Server owners to ensure that a time source for timestamping is available, and that its reliability and accuracy is acceptable to them.

1.2.6 Timestamp Server 2.0.2 Patch 1 is distributed with Timestamp Server 2.0.2. Timestamp Server 2.0.2 must be installed prior to applying Patch 1. Patch 1 updates components of Timestamp Server 2.0.2 that contain incorrect timezone definition for (GMT+0) Casablanca, Monrovia. Timestamp Server is to be set to the (GMT+0) Casablanca, Monrovia Timezone, to prevent the case where an administrator changes the system time during a daylight saving transition causing a double transition as NT4 does not allow specifying if the new time entered is DST time or normal time.

1.3 Security Target Organisation

1.3.1 The main sections of the Security Target are its TOE description, TOE Security Environment, Security Objectives, IT Security Requirements, TOE Summary Specifications, Protection Profile Claims and Rationale.

1.3.2 The *TOE Description* provides general information about the TOE, serves as an aid to understanding its security requirements, and provides context for the ST's evaluation.

1.3.3 The *TOE Security Environment* describes security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes descriptions of a) assumptions regarding the TOE's intended usage and environment of use, b) threats relevant to secure TOE operation, and c) organisational security policies with which the TOE must comply.

1.3.4 The *Security Objectives* reflect the stated intent of the ST. They pertain to how the TOE will counter identified threats and it will cover identified organisational security policies and assumptions. Each security objective is categorised as being for the TOE, or for the environment.

1.3.5 All of the requirements in this ST apply to the TOE itself, as opposed to the TOE environment. The IT security requirements are subdivided as follows: (a) TOE Security Functional Requirements, including strength-of-function requirements for TOE security functions realised by a probabilistic or permutational mechanism, and (b) TOE security assurance requirements.

1.3.6 The *TOE Summary Specification* defines the instantiation of the security requirements of the TOE. This specification describes the security functions and assurance measures of the TOE that meet the TOE security requirements. The TOE Summary Specification section covers the IT security functions and specifies how these functions satisfy the TOE security functional requirements. It includes a bi-directional mapping between functions and requirements that shows which functions satisfy which requirements and that all requirements are met.

1.3.7 The *Protection Profile claims* section contains the Protection Profile conformance claim statements. Although there are no Protection Profile conformance claims, this section is provided for completeness.

1.3.8 The *Rationale* presents evidence that the ST is a complete and cohesive set of requirements and that a conformant TOE would provide an effective set of IT security countermeasures within the security environment.

1.3.9 The Rationale is factored into two main parts. First, a Security Objectives Rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them. Then, a Security Requirements Rationale demonstrates that the security requirements (TOE and environment) are traceable to the security objectives and are suitable to meet them.

1.4 CC Conformance Claim

1.4.1 The TOE conforms to the Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408), Version 2.1, Parts 2 and 3 as follows:

a) Part 2 conformant - the security functional requirements are only based upon functional components identified in Part 2 of the CC

b) Part 3 conformant - the security assurance requirements are in the form of an evaluation assurance level (EAL 3) that is based only upon assurance components in Part 3 of the CC.

1.5 Glossary

Name	**Description**
AKVT	Audit Key Vault Token
API	Application Programming Interface
DLL	Dynamically Linked Library. These are software modules that are loaded by the program when it executes. These modules may be developed and compiled separately to the main program.
Keytools	Keytools is software library used by the TSS to perform cryptographic functions.
Key Token	A key token is the information used by the TSS to define the audit and reference keys that it uses. It includes information on the location of the key, a name and key identification information (eg certificate details).
National Authority	The National Authority is the body responsible for approving the measures used to meet specified environment security objectives. Typically this is a government organisation that is responsible for approving IT security

	solutions for government work (eg NSA)
OID	Object Identifier – These are used by the TOE to uniquely identify policies.
PKI	Public Key Infrastructure
Salted hash	The result of a non-standard Hash Algorithm using internal code word(s) to mask the algorithm used.
ST	Security Target
Security Related Events	Security related events refers to those security events recorded in the events log (see table 5.2)
Timestamp Server	The product name
TOE	Target of Evaluation
TS	Abbreviation of Timestamp
TSA	Timestamp Authority
TSS	Abbreviation of Timestamp Server
TSS Client	The client component of the product
TSS Client Configuration GUI	TSS Client configuration and administration GUI
TSS Server	The server component of the product.
TSS Server Administration GUI	TSS Server configuration and administration GUI

1.6 References

1.6.1 The following documents were referenced in the preparation of this Security Target:

[CC] Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408), version 2.1, Parts 1, 2 and 3

[PPST_G] Guide for production of Protection Profiles and Security Targets, version 0.8, ISO/IEC WD 15446, M. Donaldson, July 1999

[TSS_O] Time Stamp Server Overview, Baltimore Technologies, version 1.0, August 2000

[TSS_FS] Time Stamp Server Functional Specification, Baltimore Technologies, version 2.0.1h, December 2000

[IETF_TS] Internet X.509 Public Key Infrastructure Time Stamp Protocol (TSP), Internet Draft PKIX Working Group, October 2000, <draft-ietf-pkix-time-stamp-11.txt>.

[RFC3161] Adams et al, Internet X.509 Public Key Infrastructure, Time-Stamp Protocol (TSP). RFC 3161 August 2001.

[PKCS11] PKCS #11 Cryptographic Token Interface Standard, RSA Laboratories, v2.11 Draft 1, November 2000

[PKCS12] PKCS#12 Personal Information Exchange Syntax, RSA Laboratories, v1.0, June 24, 1999

[SHA-1] Digital Signature Algorithm, Federal Information Processing Standards Publication 180-1, 17 April 1995.

[DSA] Digital Signature Algorithm, Federal Information Processing Standards Publication 186-1, 19 May 1994.

[PKCS1] PKCS#1 RSA Encryption Standard, RSA Laboratories, v1.5, November 1, 1999

2. TOE Description

2.1 Product Type

2.1.1 The Timestamp Server is a PKI/Cryptography standards compliant server for generating digital timestamps. Timestamp tokens are produced in response to requests from clients. These timestamp tokens provide evidence that a data item existed at a particular point in time. The Timestamp Server utilises Public Key technology to digitally sign the timestamp tokens for the purposes of non-repudiation and integrity.

2.1.2 The interaction between the Timestamp Server and a client application conforms to the Internet X.509 Public Key Infrastructure Time Stamp Protocol [RFC3161] over a TCP/IP socket interface. Timestamp Server 2.0.2 was designed to be compliant to the draft Timestamp Protocol for sockets [ITEF-TS] but is still compliant to the current timestamp protocol [RFC3161].

2.2 General TOE Functionality

2.2.1 The purpose of a timestamping service is to provide evidence to third parties of the existence of a data item, with an indication of the time at which the data existed. This evidence can be used for time-critical applications such as the submission of a tender, or to indicate the time of transaction for entries in a log. A timestamp server also complements a Public Key Infrastructure by providing the means to verify that a digital signature was applied before the corresponding certificate was revoked. This facilitates the use of a revoked public key certificate for verifying signatures created prior to the time of revocation.

2.2.2 A timestamp transaction involves two entities: a Timestamp Client, which requests a timestamp, and a trusted Timestamp Server, which provides the timestamping service. The principal flow of a transaction is illustrated in Figure 2.1 below and can be summarised as follows:

1. A transaction is always initiated by a Timestamp Client sending a request to the Timestamp Server. The request message contains a hash of the data to be timestamped and a reference to a policy specifying the parameters (such as time source and key strength) according to which the data is to be timestamped. The set of policies is developed and published by the Timestamp Server administration.

2. The request is processed by the Timestamp Authority (TSA) within the Timestamp Server that implements the policy specified in the request. The TSA generates a Timestamp Token containing time information, the original hash, and a signature generated using a private key associated with the TSA.
3. The Timestamp Token is sent to the requestor in a response message. The Timestamp Client is responsible for validating and storing the token. The token is then available for use in a non-repudiation framework to prove to a third party that the data existed at a particular point in time.

2.2.3 It should be noted that it is the responsibility of TOE Administrators to configure the TSAs to implement the policies correctly.

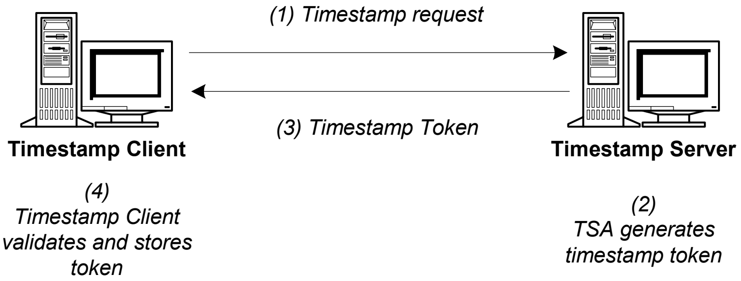
(1) Timestamp request
(3) Timestamp Token
Timestamp Client
Timestamp Server
(4) Timestamp Client validates and stores token
(2) TSA generates timestamp token

Figure 2.1 – Timestamp Transaction Flow

2.3 TSS Definition

2.3.1 Baltimore's Timestamp Server (TSS) is sold as a product that comprises the following components:

a) TSS Server (including software cryptographic functions) - the server can be configured to use hardware cryptography implemented by third party PKCS#11 [PKCS11] compliant devices

b) TSS Client development kit which includes the TSS Client Timestamp API in C++

c) Product documentation.

2.3.2 The TSS Client Timestamp API enables a developer to build a standards compliant TSS Client application that will submit requests to a Timestamp Server.

2.4 TSS Components

2.4.1 The Timestamp Server consists of the following major components as shown in Figure 2.2:

a) Timestamp Server run-time engine (TSS engine), which contains one or more Timestamp Authorities (TSA). The number of TSAs is configurable

b) Cryptographic modules for key storage and cryptographic operations. Note that PKCS#11 devices are represented as being external to the Timestamp Server hardware, however they may in practice be incorporated with the Timestamp Server hardware (e.g. connected to PCI bus) or connected externally on a hardware port or via an internal network

c) Transaction Log database, Oracle Database and Database client software

d) Events log

e) Configuration Parameters File

f) Audit Key Vault Token (AKVT)

g) Administration Utility/GUI.

2.4.2 The TSS components are shown below.

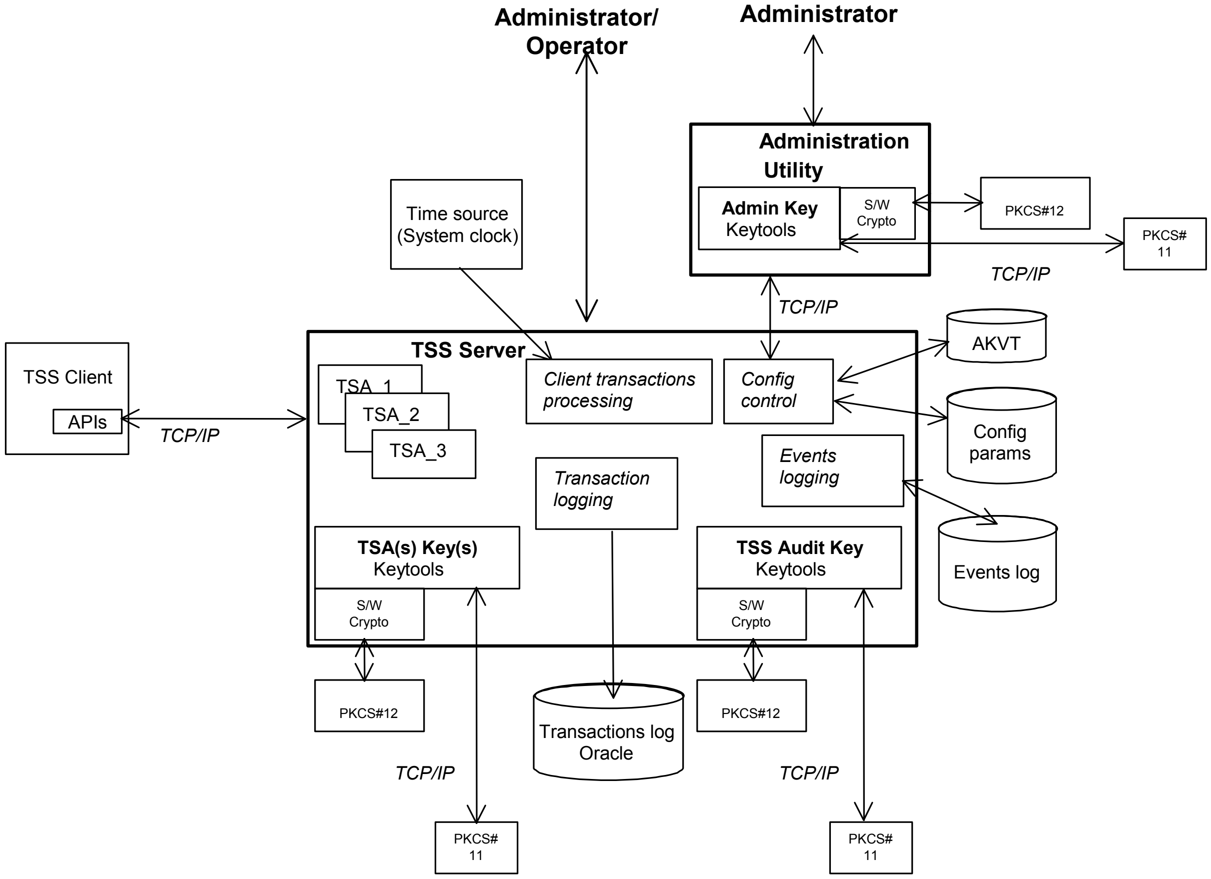

Figure 2.2 – Timestamp Server

2.4.2.1 The TOE may be installed on a single workstation or spread across a number of devices. However, the hardware configuration used must comply with the assumptions A.Location and A.Connectivity.

2.4.3 Timestamp Authorities (TSA)

2.4.3.1 A Timestamp Authority (TSA) is an entity that represents a point of service for the client. A physical Timestamp Server contains one or more logically independent TSAs. TSAs process timestamp requests and generate signed timestamp tokens. Each TSA can have its own signing key, or can share a key with another TSA.

2.4.3.2 A single TSS may implement one or more TSAs each of which implements a particular policy defined by an Object Identifier (OID). This policy is used to identify the TSA in a timestamping request. Thus, within the TSS each TSA must have a unique policy. Each TSA is associated with its own configuration parameters that specify the policy it will implement by reference to an OID. An OID must be supplied in a timestamp request.

2.4.3.3 A Timestamp Server listens for incoming timestamping requests on a user defined IP port (the port number is configurable by the TOE Administrator). All TSAs share the same port.

2.4.3.4 Each record in the transaction log contains a reference to the TSA that executed the transaction.

2.4.4 Cryptographic Modules

2.4.4.1 PKI keys are used by the Timestamp Server for three purposes, they are denoted as:

a) Audit Key - for signing the Configuration Parameters File and records in the Transaction Log and Events Log

b) TSA Key(s) - for signing timestamps

c) Administrator Key(s)- for authenticating administrators (signing administration requests).

2.4.4.2 Each key is contained inside a secure cryptographic module. The cryptographic module can be a software-based encrypted file a PKCS#12 [PKCS12], or a hardware security module PKCS#11 [PKCS11]. The same cryptographic module can contain all keys, or separate modules can be used for different keys.

2.4.4.3 The Timestamp Server is configured with parameters, except the passphrases, necessary to access select and unlock a key in each cryptographic module, be it software-based or hardware based. The TSS Administration GUI provides means for specifying the required parameters, and associating a key with its intended use.

2.4.4.4 In order to perform signing operations, the Timestamp Server engine has to unlock a corresponding cryptographic module containing the necessary key. A cryptographic module will be unlocked when first required, and kept open for the duration of the Timestamp Server operations (i.e. until the service is stopped)

2.4.4.5 The Timestamp Server interacts with the cryptographic modules via the Baltimore Technologies KeyTools Library vault.

2.4.5 Transaction Log Database

2.4.5.1 All successful timestamping transactions performed by the Timestamp Server engine are logged in the Transaction Log. The log is maintained in an Oracle database. The interface from the Timestamp Server to the Oracle database is via embedded SQL.

2.4.5.2 Note that the TOE security functionality does not provide any facilities for viewing the contents of the Transactions Log.

2.4.6 Events log

2.4.6.1 The Timestamp Server Engine is responsible for logging system and Security related events to the Events Log. The log is maintained as a single ASCII file.

2.4.6.2 In order to preserve the integrity of the records, each security related event record in the Events Log is signed with the Audit Key. In order to preserve the integrity of the whole events log, each security related record in the Events Log, except the first, contains a hash value of the previous record's data.

2.4.6.3 Note that the TOE security functionality does not provide any facilities for viewing the contents of the Events Log.

2.4.7 Configuration Parameters File

2.4.7.1 The Configuration Parameters File is an XML file that contains all configuration details for the Timestamp Server. Whenever the configuration file is changed, the Timestamp Server (re)signs the file with the Audit key. This signature is checked every time the Timestamp Server engine reads the configuration file at start-up.

2.4.7.2 While the Configuration Parameters File is capable of storing the passphrases for the audit and TSA keys this is not part of the evaluated configuration.

2.4.8 Audit Key Vault Token (AKVT)

2.4.8.1 The Audit Key Vault Token (AKVT) is a salted hash of the Audit Key certificate created during the system bootstrap process. It is used to check the integrity of the Audit Key during system start-up.

2.4.9 Administration Utility/GUI

2.4.9.1 The TSS Server Administration GUI enables administrators to view and modify the configuration parameters of the Timestamp Server. The configuration parameters include global configuration parameters that are applicable to the Timestamp Server as a whole, and TSA configuration parameters specific to individual TSAs. All configuration parameters are stored in a dedicated Configuration Parameters File.

2.5 TSS Scope and Boundary

2.5.1 TSS Boundary

2.5.1.1 The TSS consists of the components identified in 2.4. The primary interface to the TSS is a TCP/IP network that connects the TSS Server to the Oracle Database server, the Administration Utility to the TSS Server, the TSS Server to the PKCS#11 Device, and the timestamp requestor to the TSS. The TCP/IP link is established via a WAN/LAN that the TOE accesses via a communication protocol stack. The nature of the underlying hardware is hidden from the operation of the TSS. While the Administrator utility uses TCP/IP to access the TSS Server it can only be configured to be on the same physical device.

2.5.1.2 Assumption A.Connectivity requires that the network segment that the TSS (and all associated components of the Timestamp Server as set out in Section 2.4) reside on be on a protected network. This would normally be achieved through measures such as a boundary firewall and other security measures. The timestamp requestor is external to the protected network and only has access to the TSA port to request and receive timestamps.

2.5.1.3 The PKCS#12 entity is a software file residing on a hard disk. It is accessed using standard operating system calls, using the PKCS#12 [PKCS12] protocol. The system clock is used as the time source and it is accessed using standard operating system calls.

2.5.2 Physical and Software Components

2.5.2.1 The Timestamp Server is installed on a hardware platform that meets the minimum requirements specified below.

Hardware	Pentium II 350Mhz processor 128 MB RAM;
Operating System	Microsoft Windows NT Server 4.0, Service Pack 6a
Network Interface	TCP/IP

2.5.2.2 Additional software required by the Timestamp Server are listed below:

Transaction Log Database	Oracle version 8.1.6

2.5.2.3 Key storage formats supported by the Timestamp Server are detailed below:

Audit Key	PKCS#12, PKCS#11 token
TSA Key(s)	PKCS#12, PKCS#11 token
Administrator Key	PKCS#12, PKCS#11 smartcard

2.5.2.4 The following key usage parameters for Certificates used with the Timestamp Server must be set:

Audit Key	key usage includes: “digitalSignature” and/or “nonRepudiation”
TSA Key(s)	key usage includes: “digitalSignature” extended key usage set to “timestamping”
Administrator Key	no key usage necessary.

2.5.3 Evaluated Configuration

2.5.3.1 The Timestamp Server evaluated configuration will be as per figure 2.5.3.1. The TOE requires that Timestamp Server 2.0.2 Patch 1 must be added after Timestamp Server 2.0.2 is installed. The system timezone must be set to (GMT+0) Casablanca, Monrovia.

The unshaded items in figure 2.5.3.1 are within the scope of the evaluation. Further in reference to figure 2.5.3.1:

a) The TSS Server and TSS Administration Utility are part of the TOE.

b) The configuration file, events log, and AKVT are part of the TOE.

c) When used with PKCS#12 [PKCS12] compliant key storage file formats, the protection of the file, loading and cryptographic operations performed are included in the evaluation. The format and interface specifications are outside the scope of the evaluation. The level of protection offered by using PKCS#12 file(s) is outside the scope of the evaluation. The PKCS#12 file(s) can be used for Administrator(s) Key(s), Audit Key and TSA Key(s) storage.

d) When used with PKCS#11 [PKCS11] compliant key storage device(s), the protection of the physical device, public key and certificate loading, and type of cryptographic operations are covered by the assumptions A.Connectivity, A.Location and the policy P.Cryptography. The format and interface specifications are outside the scope of the evaluation. The level of protection offered by the PKCS#11 device is outside the scope of the evaluation. The cryptographic operations performed (signing and verifying) are outside the scope of the evaluation. The PKCS#11 device(s) can be used to hold the Administrator(s), Audit Key, and TSA Key(s). Only PKCS#11 smartcards that hold only one signing key and certificate can be used for Administrator(s) Key(s).

e) The Time Source, and System Clock Synchronisation utilities are outside the scope of the evaluation.

f) Key generation and certificate generation environment and tools are outside the scope of the evaluation.

g) The Client and TSS Client API are not part of the TOE.

h) The Transactions Log is part of the TSS but not part of the TOE.

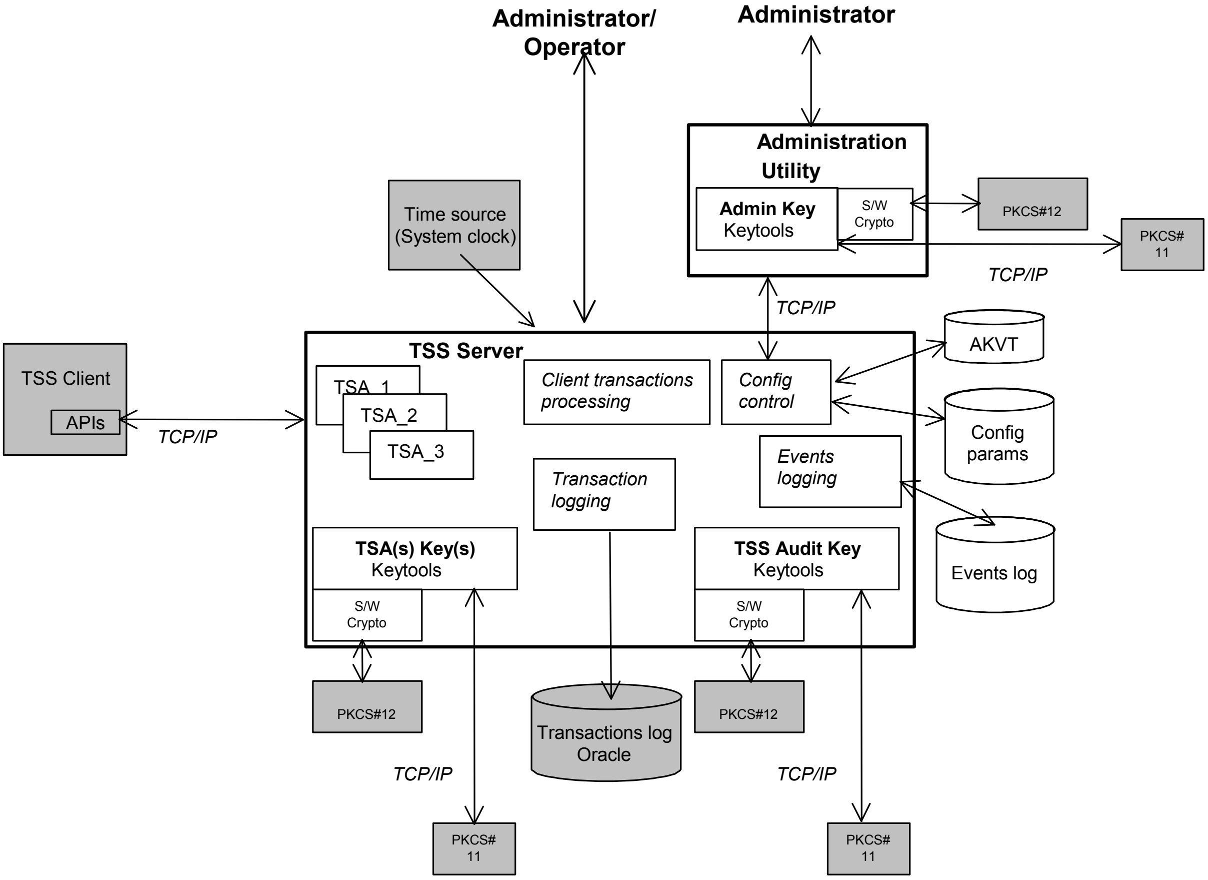

Figure 2.5.3.1 Timestamp server evaluated configuration.

2.5.3.2 Software cryptographic functions are performed by the Timestamp Server engine. If used with PKCS#11 cryptographic hardware devices, the cryptographical functions, signing, verification, and creating secure hashes functions are performed by the PKCS#11 cryptographic hardware devices. The software cryptographic functions will be used if the PKCS#11 cryptographic device does not provide the functions. The PKCS#11 cryptographic device must be assured to at least CC EAL3, or equivalent and or approved by the National Authority.

2.5.3.3 The following information applies to the evaluated configuration:

a) Each TSA can have its own signing key, or can share a key with another TSA

b) Passphrases for TSA and Audit keys are not to be stored as configuration parameters. They are to be entered manually each time the Timestamp Server is started.

c) The Transactions Log (Oracle database) may be installed on the same hardware platform as the Timestamp Server engine, or on a separate machine connected via an internal network.

3. TOE Security Environment

3.1 Introduction

3.1.1 This section contains a statement of the TOE Security Environment. It describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be employed.

3.2 Assumptions

3.2.1 This section describes the security aspects of the environment in which the TOE is intended to be used. It includes information about the intended usage of the TOE, including such aspects as the intended application, potential asset value, and possible limitations of use; and information about the environment of use of the TOE, including physical, personnel, and connectivity aspects.

Identification	Description
A.Time_Source	It is assumed that TOE owners are responsible for ensuring that a time source for timestamping is available, and that its reliability and accuracy is acceptable to the TOE owner.
A.PKI	It is assumed that the TOE operates within a securely managed PKI such that all keys and certificates associated with the TOE are issued and revoked securely and that the status of all keys and certificates are checked prior to their use.
A.Key_Storage	It is assumed that all private keys used in the operation and administration of the TOE are securely stored to prevent access by persons other than authorised TOE Administrators.
A.Location	The TOE (and all associated components of the Timestamp Server as set out in Section 2.4) is assumed to be located within controlled access facilities that will prevent unauthorised physical access
A.Connectivity	It is assumed that the TOE (and all associated components of the Timestamp Server as set out in Section 2.4) resides on a dedicated network and that measures are in place to protect this network from attacks from external networks.

Identification	Description
A.System_Administrator	It is assumed that one or more authorised persons are assigned the responsibility for securely installing and maintaining the TOE in its evaluated configuration, but are not given access to any of the keys associated with the TOE. These are the System Administrators. System Administrators are assumed to be trusted, competent and possess sufficient knowledge and training to carry out their duties. The duties of System Administrators include such things as: 1. Ensuring that no malicious software is running on the same platform as the TOE or has access to the TOE 2. Ensuring that there is adequate disk space for the TOE's requirements 3. Ensuring that the TOE's databases are properly maintained
A.TOE_Administrator	It is assumed that one or more authorised persons are assigned the responsibility to securely configure and manage the TOE. These are the TOE Administrators. There are the following classes of TOE Administrators (a single person may play one or more of these roles): 1. **Bootstrap Administrator** – initially configures the system during the system BOOTSTRAP mode, requires access to the Audit Key and Administrator certificates 2. **TSS Administrator** – administers the TSS Server, requires access to a TSS Administrator Key 3. **TSS Operator** – starts up the TSS Server. May need to type in passphrase for Audit and/or TSA Keys. TOE Administrators are assumed to be trusted, competent and possess sufficient knowledge and training to carry out their duties.

Identification	Description
A.TS_Requestor	It is assumed that the Timestamp User (Timestamp Requester) validates and retains the timestamp token produced. This includes checking, using out-of-band methods, that the TSA certificate has not been revoked, and that the timestamp token was signed by the correct TSA. The Timestamp User is assumed to retain the timestamp token for non-repudiation evidence.
A.P11_Device	If used, it is assumed that the TSS owners will select a hardware cryptographical device that can: • Perform RSA [1024 or 2048 bit] and DSA [1024 bit] signing and verification; • Produce a SHA-1 secure hash. • And is compliant with PKCS#11 [PKCS11] standard. The device must be approved by the National Authority or evaluated to Common Criteria EAL3 or equivalent for these requirements.

Table 3.2 – Assumptions

3.3 Threats

3.3.1 This section describes all threats to the assets against which specific protection within the TOE, or its environment is required. Each threat is described in terms of an identified threat agent, the attack, and the asset that is the subject of the attack. Threat agents are described by addressing their required expertise, available resources, and motivation. Attacks are described by addressing the attack methods, any vulnerabilities exploited and opportunity.

Identification	Description
T.Hack_Imperson_Admin	Persons other than an authorised TOE Administrator gain access to the administration utility by impersonating an authorised TOE Administrator. The attacker would be required to forge or steal an identity and authenticate themselves to the TOE. The required level of expertise and resources, is high. The motivation of such a hacker depends upon the value they assign to the data they wish to timestamp.
T.Hack_Imperson_TOE	A hacker located on a network external to the TOE's network, and possessing a high level of expertise and resources, impersonates the TOE by forging timestamp tokens. The motivation of such a hacker depends upon the value they assign to the data they wish to timestamp.
T.Hack_Mod_Timestamp	The contents of the timestamp produced by the TOE are altered by either the intended recipient of the timestamp or a hacker who intercepts the timestamp destined for its recipient. The required level of expertise and resources to accomplish this is high. The motivation of such a hacker depends upon the value they assign to the timestamp.
T.Client_Refute_Origin	The intended recipient of a timestamp refutes the origin of the timestamp by claiming that it did not come from the TOE. The Client would need to modify the signature of the Timestamp token. The required level of expertise and resources to accomplish this is high. The motivation of such a client depends upon the value they assign to the existence of the timestamp.
T.Config_Mod_Undetect	A TOE Administrator makes changes to the configuration of the TOE that is undetected by the TOE. It is subsequently not possible to determine the configuration of the TOE at a given point in time. The motivation of such an Administrator depends upon the value they assign to the existence of the timestamp. This requires a high level of expertise and resources to impersonate the TOE. The motivation of such an attack depends upon the value they assign to the data they wish to timestamp.

Identification	Description
T.Log_Mod_Undetect	A TOE Administrator inadvertently makes changes to the Events log. It is subsequently not possible to verify the integrity of the log. This attack would require a lack of knowledge of the TOE, but would require a high level of resources and expertise to mask the changes.
T.Mod_Config_Data	Changes to the Configuration Parameters File - either unintentionally by a TOE Administrator or deliberately by an attacker (persons other than TOE Administrators) - alters the configuration of the TOE. Such changes will not be recorded in the events log. This would require the impersonation of the TOE. This requires high level of resources and expertise to mask the changes. The motivation of such a hacker depends upon the value they assign to the timestamp.
T.Replace_Audit_Key	Replacement of the Audit key - either unintentionally by a TOE Administrator or deliberately by an attacker (persons other than TOE Administrators) - goes undetected. For the Administrator this would require a lack of knowledge of the TOE. For an attacker this would require access to the TOE or TOE network. The required level of expertise and resources to accomplish this is high. The motivation of such a hacker depends upon the value they assign to the timestamp.
T.Replace_TSA_Key	Replacement of TSA key(s) - either unintentionally by a TOE Administrator or deliberately by an attacker (persons other than TOE Administrators) - goes undetected. For the Administrator this would require a lack of knowledge of the TOE. For an attacker this would require access to the TOE or TOE network. The required level of expertise and resources to accomplish this is high. The motivation of such a hacker depends upon the value they assign to the timestamp.

Table 3.2 – Threats

3.4 Organisational Security Policies

3.4.1 This section identifies the organisational security policy statements or rules with which the TOE must comply.

Identification	Description
P.Cryptography	All cryptographic operations (signing and verification) must be implemented by algorithms approved by the National Authority.
P.Key_Generation_Destruction	All cryptographic keys and certificates (TOE Administrator and system related) must be produced and destroyed externally to the TOE by a method approved by the National Authority.
P.Passphrases_PINs	All passphrases and PINs used to access private keys associated with the TOE (TOE Administrator and system related) must be kept confidential, changed regularly and conform to the requirements set by the National Authority.

Table 3.3 – Organisation Security Policies

4. Security Objectives

4.1 Introduction

4.1.1 This section defines the security objectives to be satisfied by the TOE and the security objectives to be satisfied by IT and non-IT measures within the TOE environment. It addresses all of the identified aspects of the security environment.

4.2 Security Objectives for the TOE

4.2.1 The following security objectives for the TOE trace back to aspects of identified threats to be countered by the TOE and/or organisational security policies to be met by the TOE.

Identification	Description
O.Ident_Authent	The TOE will ensure that each TOE Administrator is uniquely identified and that the claimed identity is authenticated before the TOE Administrator is granted access to the TOE's security related data.
O.Timestamp	The TOE will provide the means for generating and issuing timestamp tokens by binding time information[1] with some given data. The TOE will provide the means for ensuring that timestamp tokens are associated with the identity of the TOE.
O.Audit	The TOE will provide the means of recording security related events so as to assist a TOE Administrator in the detection of potential attacks or misconfiguration of the TOE security features that would leave the TOE susceptible to attack. The TOE will provide the means for generating evidence in each security related event record of the log that allows the TOE Administrator to verify the integrity of the record and to detect if the previous record has been deleted or modified. This does not prevent the case where the last record or records are deleted before a new record is added.
O.Integrity_Config	The TOE will provide the means of preserving the integrity of the TOE's configuration. This includes the configuration data stored in the Configuration Parameters File, and protection of the audit and TSA keys.

Table 4.1 – Security Objectives for the TOE

[1] This is the time that the timestamp token is generated.

4.3 Security Objectives for the Environment

4.3.1 The following security objectives for the environment trace back to aspects of identified threats not completely countered by the TOE and/or organisational security policies or assumptions not completely met by the TOE.

Identification	Description
OE.Time_Source	Those responsible for the TOE are responsible for ensuring that a time source for timestamping is available, and that its reliability and accuracy is acceptable to the TOE owner.
OE.PKI	Those responsible for the TOE must ensure that the TOE operates within a securely managed PKI such that all keys and certificates are issued and revoked securely and that the status of all keys and certificates are checked prior to their use.
OE.Cryptography	Those responsible for the TOE must ensure that all cryptographic operations (signing and verification) have been implemented by algorithms approved by the National Authority.
OE.P11_Device	Those responsible for the TOE must ensure that any hardware device used with the TSS shall: • Perform RSA [1024 or 2048 bit] and DSA [1024 bit] signing and verification; • Produce a SHA-1 secure hash; and • Be compliant with PKCS#11 [PKCS11] standard. Be approved by the National Authority or evaluated to Common Criteria EAL3 or equivalent for these requirements.
OE.Key_Generation_Destruction	Those responsible for the TOE must ensure that all cryptographic keys and certificates used in the operation and administration of the TOE have been produced and destroyed by a source approved by the National Authority.
OE.Passphrases_PINs	Those responsible for the TOE must ensure that procedures exist for the selection and management of passphrases and PINs to conform to the requirements set by the National Authority.
OE.Key_Storage	Those responsible for the TOE must ensure that all private keys used in the operation and administration of the TOE are securely stored to prevent access by persons other than TOE Administrators.

Identification	Description
OE.Physical	Those responsible for the TOE must ensure that those parts of the TOE (and all associated components of the Timestamp Server as set out in Section 2.4) that are critical to security policy enforcement are protected from physical attack which might compromise IT security.
OE.Connectivity	Those responsible for the TOE must ensure that no connections are provided from the TOE (and all associated components of the Timestamp Server as set out in Section 2.4) to outside systems that would undermine IT security.
OE.System_Administrator	Those responsible for the TOE must ensure that all System Administrators are appropriately vetted, possess the required knowledge and receive sufficient training to carry out their duties. System Administrators must ensure that the TOE is installed and maintained in a manner which maintains IT security, as specified in the product documentation.
OE.TOE_Administrator	Those responsible for the TOE must ensure that all TOE Administrators are appropriately vetted, possess the required knowledge and receive sufficient training to carry out their duties. TOE Administrators must ensure that the TOE is configured and run in a manner which maintains security, as specified in the product documentation.

Identification	Description
OE.TS_Requestor	The Timestamp Requestor is responsible for validating and retaining the timestamp token produced by the TOE. This includes checking, using out-of-band methods, that the TSA certificate has not been revoked, and that the timestamp token was signed by the correct TSA. The Timestamp Requestor is assumed to retain the timestamp token for non-repudiation evidence.
OE.Audit_Log	Those responsible for the TOE must provide some management and protection of the audit log, as follows: 1. They must ensure that adequate space is available by archiving and removing old records 2. While the TOE provides some protection for the Audit Log, there is also a requirement for the environment to provide some protection. This is because audit records can be deleted starting from the most recent without detection.

Table 4.2 – Security Objectives for the environment

5. IT Security Requirements

5.1 Introduction

5.1.1 This section defines the detailed IT security requirements that shall be satisfied by the TOE or its environment.

5.2 TOE Security Functional Requirements

5.2.1 This section defines the Security Functional Requirements (SFRs) of the TOE as functional components drawn from the Common Criteria (CC) Part 2. The SFRs are summarised in the following table.

Functional Class		Functional Component	
FAU	Audit	FAU_GEN.1	Audit Data Generation
FCO	Communication	FCO_NRO.2	Enforced Proof of Origin
FCS	Cryptographic Support	FCS_CKM.3	Cryptographic Key Access
		FCS_COP.1	Cryptographic Operation
FIA	Identification and Authentication	FIA_UAU.1	Timing of Authentication
		FIA_UID.1	Timing of Identification
FPT	Protection of the TSF	FPT_ITT.1	Basic Internal TSF Data Transfer
		FPT_ITT.3	TSF Data Integrity Monitoring

Table 5.1 – TOE Security Functional Requirements

5.2.2 The remainder of this section contains the functional components from Part 2 of the CC with operations completed. The standard CC text is in regular font and the text inserted by the author is in italic font enclosed in brackets.

5.2.3 Audit data generation (FAU_GEN.1)

5.2.3.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [*not specified*] level of audit; and

c) [*all security related events listed in Table 5.2*].$^{\text{FAU\_GEN.1.1}}$

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*other audit relevant information as listed in Table 5.2*] FAU_GEN.1.2

ID	Security Related Event	Other Audit Information
1	Change global configuration parameter: Event Log name and location	new configuration parameters
2	Stop TSA	none
3	Create TSA	new TSA parameters
4	Modify TSA	new TSA parameters
5	Delete TSA	none
6	Create Key Token	new Key token parameters
7	Modify Key Token	new Key token parameters
8	Delete Key Token	name of deleted Key token
9	Start TSS	entire configuration parameter set
10	Shutdown TSS	none
11	<Not Used>	
12	<Not Used>	
13	<Not Used>	
14	<Not Used>	
15	<Not Used>	
16	<Not Used>	
17	<Not Used>	
18	<Not Used>	
19	TSA private key cannot be accessed	error message
20	<Not Used>	
21	<Not Used>	
22	<Not Used>	
23	Enter Configuration mode	none
24	TSS Crash	none

Table 5.2 – Security Related Events

5.2.4 Enforced proof of origin (FCO_NRO.2)

The TSF shall enforce the generation of evidence of origin for transmitted [*administration requests*] at all times.$^{\text{FCO\_NRO.2.1}}$

The TSF shall be able to relate the [*digital signature*] of the originator of the information, and the [*entirety (i.e. the whole administration request)*] of the information to which the evidence applies.$^{\text{FCO\_NRO.2.2}}$

The TSF shall provide a capability to verify the evidence of origin of information to [*recipient (Timestamp Server engine),*][*TOE administrators*] given [*no limitations on the evidence of origin*].$^{\text{FCO\_NRO.2.3}}$

5.2.5 Cryptographic key access (FCS_CKM.3)

The TSF shall perform [*key loading from a PKCS#12 file or PKCS#11 device*] in accordance with a specified cryptographic key access method [*Keytools vault library API*] that meets the following: [*PKCS#12[PKCS12] and PKCS#11[PCKS11] standards*].$^{\text{FCS\_CKM.3.1}}$

5.2.6 Cryptographic operation (FCS_COP.1A)

The TSF shall perform [*digital signing and verification*] in accordance with a specified cryptographic algorithm [*RSA digital signature*] and cryptographic key sizes [*1024, 2048 bits*] that meet the following: [*PKCS #1[PCKS1] standard*].$^{\text{FCS\_COP.1.1}}$

5.2.7 Cryptographic operation (FCS_COP.1B)

TSF shall perform [*digital signing and verification*] in accordance with a specified cryptographic algorithm [*DSA digital signature*] and cryptographic key sizes [*1024 bits*] that meet the following: [*DSA[DSA] standard*].$^{\text{FCS\_COP.1.1}}$

5.2.8 Cryptographic operation (FCS_COP.1C)

The TSF shall perform [*hashing*] in accordance with a specified cryptographic algorithm [*SHA-1 hash algorithm*] and cryptographic key sizes [*160 bit message digest*] that meet the following: [*SHA-1[SHA-1] standard*].$^{\text{FCS\_COP.1.1}}$

5.2.9 Timing of Authentication (FIA_UAU.1)

The TSF shall allow [*Startup of TSS Server, Events log Creation, Audit Key Registration, Administrator User Entries*] on behalf of the user to be performed before the user is authenticated.$^{\text{FIA\_UAU.1.1}}$

The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.$^{\text{FIA\_UAU.1.2}}$

5.2.10 Timing of Identification (FIA_UID.1)

The TSF shall allow [*Startup of TSS Server, Events log Creation, Audit Key Registration, Administrator User Entries*] on behalf of the user to be performed before the user is identified.[FIA_UID.1.1]

The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. [FIA_UID.1.1]

5.2.11 Basic internal TSF data transfer protection (FPT_ITT.1)

The TSF shall protect TSF data from [*modification*] when it is transmitted between separate parts of the TOE.[FPT_ITT.1.1]

5.2.12 TSF data integrity monitoring (FPT_ITT.3)

The TSF shall be able to detect [*modification of administration request, substitution of administration request*] for TSF data transmitted between separate parts of the TOE.[FPT_ITT.3.1]

Upon detection of a data integrity error, the TSF shall take the following actions: [*deny administration request*].[FPT_ITT.3.2]

5.3 TOE Security Assurance Requirements

5.3.1 This section defines the Security Assurance Requirements (SARs) of the TOE as Evaluation Assurance Level (EAL) 3, specified in terms of assurance components in the Common Criteria (CC) Part 3. The SARs are summarised in the following table.

Assurance Class		**Assurance Component**	
ASE	Security Target	ASE_DES.1	TOE Description
		ASE_ENV.1	Security Environment
		ASE_INT.1	ST Introduction
		ASE_OBJ.1	Security Objectives
		ASE_PPC.1	PP Claims
		ASE_REQ.1	IT Security Requirements
		ASE_SRE.1	Explicitly stated IT Security Requirements
		ASE_TSS.1	TOE Summary Specification
ACM	Configuration	ACM_CAP.3	Authorisation Controls

Assurance Class		Assurance Component	
	Management	ACM_SCP.1	TOE CM Coverage
ADO	Delivery and Operation	ADO_DEL.1	Delivery Procedures
		ADO_IGS.1	Installation, Generation, and Start-up Procedures
ADV	Development	ADV_FSP.1	Informal Functional Specification
		ADV_HLD.2	Security Enforcing High-Level Design
		ADV_RCR.1	Informal Correspondence Demonstration
AGD	Guidance Documents	AGD_ADM.1	Administrator Guidance
		AGD_USR.1	User Guidance
ALC	Life Cycle Support	ALC_DVS.1	Identification of Security Measures
ATE	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
AVA	Vulnerability Assessment	AVA_MSU.1	Examination of Guidance
		AVA_SOF.1	Strength of TOE Security Function Evaluation
		AVA_VLA.1	Developer Vulnerability Analysis

Table 5.3 – TOE Security Assurance Requirements

5.3.2 The remainder of this section contains details of the EAL 3 assurance components from Part 3 of the CC.

5.3.3 Configuration management (ACM)

5.3.3.1 Authorisation controls (ACM_CAP.3)

a) The developer shall provide a reference for the TOE.[ACM_CAP.3.1D]

b) The developer shall use a CM system.[ACM_CAP.3.2D]

c) The developer shall provide CM documentation.[ACM_CAP.3.3D]

d) The reference for the TOE shall be unique to each version of the TOE.[ACM_CAP.3.1C]

e) The TOE shall be labelled with its reference.[ACM_CAP.3.2C]

f) The CM documentation shall include a configuration list and a CM plan.[ACM_CAP.3.3C]

g) The configuration list shall describe the configuration items that comprise the TOE.[ACM_CAP.3.4C]

h) The CM documentation shall describe the method used to uniquely identify the configuration items.[ACM_CAP.3.5C]

i) The CM system shall uniquely identify all configuration items.[ACM_CAP.3.6C]

j) The CM plan shall describe how the CM system is used.[ACM_CAP.3.7C]

k) The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.[ACM_CAP.3.8C]

l) The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.[ACM_CAP.3.9C]

m) The CM system shall provide measures such that only authorised changes are made to the configuration items.[ACM_CAP.3.10C]

5.3.3.2 TOE CM coverage (ACM_SCP.1)

a) The developer shall provide CM documentation.[ACM_SCP.1.1D]

b) The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, and CM documentation. [ACM_SCP.1.1C]

c) The CM documentation shall describe how configuration items are tracked by the CM system.[ACM_SCP.1.2C]

5.3.4 Delivery and operation (ADO)

5.3.4.1 Delivery procedures (ADO_DEL.1)

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

b) The developer shall use the delivery procedures.[ADO_DEL.1.2D]

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

5.3.4.2 Installation, generation, and start-up procedures (ADO_IGS.1)

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

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

5.3.5 Development (ADV)

5.3.5.1 Informal functional specification (ADV_FSP.1)

a) The developer shall provide a functional specification.[ADV_FSP.1.1D]

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

c) The functional specification shall be internally consistent.[ADV_FSP.1.2C]

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

e) The functional specification shall completely represent the TSF.[ADV_FSP.1.4C]

5.3.5.2 Security enforcing high-level design (ADV_HLD.2)

a) The developer shall provide the high-level design of the TSF.[ADV_HLD.2.1D]

b) The presentation of the high-level design shall be informal.[ADV_HLD.2.1C]

c) The high-level design shall be internally consistent.[ADV_HLD.2.2C]

d) The high-level design shall describe the structure of the TSF in terms of subsystems.[ADV_HLD.2.3C]

e) The high-level design shall describe the security functionality provided by each subsystem of the TSF.[ADV_HLD.2.4C]

f) 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.2.5C]

g) The high-level design shall identify all interfaces to the subsystems of the TSF.[ADV_HLD.2.6C]

h) The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.[ADV_HLD.2.7C]

i) The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.[ADV_HLD.2.8C]

j) The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems. [ADV_HLD.2.9C]

5.3.5.3 Informal correspondence demonstration (ADV_RCR.1)

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

b) 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.[ADV_RCR.1.1C]

5.3.6 Guidance documents (AGD)

5.3.6.1 Administrator guidance (AGD_ADM.1)

a) The developer shall provide administrator guidance addressed to system administrative personnel.[AGD_ADM.1.1D]

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

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

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

e) The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.$^{AGD\_ADM.1.4C}$

f) The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.$^{AGD\_ADM.1.5C}$

g) 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.$^{AGD\_ADM.1.6C}$

h) The administrator guidance shall be consistent with all other documentation supplied for evaluation.$^{AGD\_ADM.1.7C}$

i) The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.$^{AGD\_ADM.1.8C}$

5.3.6.2 User guidance (AGD_USR.1)

a) The developer shall provide user guidance.$^{AGD\_USR.1.1D}$

b) The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE. $^{AGD\_USR.1.1C}$

c) The user guidance shall describe the use of user-accessible security functions provided by the TOE.$^{AGD\_USR.1.2C}$

d) The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.$^{AGD\_USR.1.3C}$

e) 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.$^{AGD\_USR.1.4C}$

f) The user guidance shall be consistent with all other documentation supplied for evaluation.$^{AGD\_USR.1.5C}$

g) The user guidance shall describe all security requirements for the IT environment that are relevant to the user.$^{AGD\_USR.1.6C}$

5.3.7 Life cycle support (ALC)

5.3.7.1 Identification of security measures (ALC_DVS.1)

a) The developer shall produce development security documentation.$^{ALC\_DVS.1.1D}$

b) The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.$^{ALC\_DVS.1.1C}$

c) The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.$^{ALC\_DVS.1.2C}$

5.3.8 Tests (ATE)

5.3.8.1 Analysis of coverage (ATE_COV.2)

a) The developer shall provide an analysis of the test coverage.[ATE_COV.2.1D]

b) The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.[ATE_COV.2.1C]

c) The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete. [ATE_COV.2.2C]

5.3.8.2 Testing: high-level design (ATE_DPT.1)

a) The developer shall provide the analysis of the depth of testing.[ATE_DPT.1.1D]

b) The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.[ATE_DPT.1.1C]

5.3.8.3 Functional testing (ATE_FUN.1)

a) The developer shall test the TSF and document the results.[ATE_FUN.1.1D]

b) The developer shall provide test documentation.[ATE_FUN.1.2D]

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

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

e) 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.1.3C]

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

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

5.3.8.4 Independent testing - sample (ATE_IND.2)

a) The developer shall provide the TOE for testing.[ATE_IND.2.1D]

b) The TOE shall be suitable for testing.[ATE_IND.2.1C]

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

5.3.9 Vulnerability assessment (AVA)

5.3.9.1 Examination of guidance (AVA_MSU.1)

a) The developer shall provide guidance documentation. [AVA_MSU.1.1D]

b) The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.AVA_MSU.1.1C

c) The guidance documentation shall be complete, clear, consistent and reasonable.AVA_MSU.1.2C

d) The guidance documentation shall list all assumptions about the intended environment.AVA_MSU.1.3C

e) The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).AVA_MSU.1.4C

5.3.9.2 Strength of TOE security function evaluation (AVA_SOF.1)

a) 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.AVA_SOF.1.1D

b) 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.1.1C

c) 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.AVA_SOF.1.2C

5.3.9.3 Developer vulnerability analysis (AVA_VLA.1)

a) The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.AVA_VLA.1.1D

b) The developer shall document the disposition of obvious vulnerabilities.AVA_VLA.1.2D

c) The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.AVA_VLA.1.1C

5.4 Security Requirements for the IT Environment

5.4.1 There are no security requirements for the IT environment of the TOE. Each of the environmental security objectives is either met by non IT means or is the responsibility of the National Authority. The National Authority will normally approve these measures on a case by case basis.

5.4.2 The following table summarises the way in which environmental security objectives are addressed:

Environmental Security Objective	IT Environmet Security Requirement	Comment
OE.Time_Source	Nil	addressed by Non-IT means

Environmental Security Objective	**IT Environmet Security Requirement**	**Comment**
OE.PKI	Nil	addressed by Non-IT means
OE.Cryptography	Nil	responsibility of the National Authority
OE.P11_Device	Nil	responsibility of the National Authority
OE.Key_Generation_Destruction	Nil	responsibility of the National Authority
OE.Passphrases_PINs	Nil	responsibility of the National Authority
OE.Key_Storage	Nil	addressed by Non-IT means
OE.Physical	Nil	addressed by Non-IT means
OE.Connectivity	Nil	addressed by Non-IT means
OE.System_Administrator	Nil	addressed by Non-IT means
OE.TOE_Administration	Nil	addressed by Non-IT means
OE.Audit_Log	Nil	addressed by Non-IT means

Table 5.4 – Method for addressing the environmental security objectives

5.5 Minimum Strength of Function Level

5.5.1 The TOE's security functions that are realised by probabilistic or permutational mechanisms are all cryptographic in nature and are therefore assessed by the National Authority. A statement of the minimum strength of function level is therefore not relevant for this TOE.

6. TOE Summary Specification

6.1 Introduction

This section defines the instantiation of the security requirements of the TOE. This specification describes the security functions and assurance measures of the TOE that meet the TOE security requirements.

6.2 TOE Security Functions

This section covers the IT security functions and specifies how these functions satisfy the TOE security functional requirements. It includes a bi-directional mapping between functions and requirements that shows which functions satisfy which requirements and that all requirements are met.

6.2.1 IT Security Functions

The security functions provided by the TOE are:

a) Identification and Authentication of TOE Administrators
   i) Register Administrators (IA_Register)
   ii) Identify Administrator (IA_Identify)
   iii) Authenticate Administrator (IA_Authenticate)

b) Digital Timestamping
   i) Manage TSAs (DT_Manage)
   ii) Generate Timestamp (DT_Generate)

c) Protected Logging of Audit Events
   i) Register Audit Key (AL_Register)
   ii) Events logging (AL_Logging)
   iii) Protect Audit Records (AL_Integrity)

d) Protected Configuration
   i) Protect Configuration File (PC_Integrity)
   ii) Protect Audit Key (PC_Audit_Key)
   iii) Protect TSA Key(s) (PC_TSA_Key)

Security Function	Description
Register Administrators (IA_Register)	The set of TOE Administrators that may administer the TOE can only be specified during the Bootstrap process. This list then remains fixed for the duration of the TOE's installation - no user interface exists to allow the modification of TOE Administrator details. The list is stored within the configuration parameters file, protected by the security function PC_Integrity. It may only be viewed from the Administration GUI. A TOE Administrator enters the file name and file location of one or more Administrator certificates during the bootstrap process.
Identify Administrator (IA_Identify)	The Administration GUI may only be accessed via a login prompt. TOE Administrators identify themselves by specifying details of their private key, which is stored either in a software file or on a hardware device. For software private key storage, an Administrator needs to specify: a) private key file name and path b) passphrase For hardware private key storage, an Administrator needs to specify: a) DLL b) PIN Keys stored in software are loaded into memory when the passphrase is entered and the keys are then available for use by the software cryptographic processor. Keys stored in hardware remain on the hardware device where the cryptographic operations are performed. The passphrase is used to "unlock" the hardware device so that it can be used for cryptographic processing.
Authenticate Administrator (IA_Authenticate)	All requests originating from the Administration GUI are digitally signed using the TOE Administrator's private key. The Timestamp Server verifies this signature and authenticates the TOE Administrator prior to carrying out the request.
Manage TSAs (DT_Manage)	Management of Timestamp Authorities is only possible when the TOE is in Configuration mode. TOE Administrators perform

Security Function	Description
	management functions via the Administration GUI. Functions are available to Create TSAs, Delete TSAs, and to Modify TSA parameters. Create TSA: Allows a TOE Administrator to specify the name of a new TSA. A request signed with the Administrator's key is sent from the Administration GUI to the Timestamp Server engine for processing. Modify TSA: Allows a TOE Administrator to select a TSA to be modified. The parameters for the TSA are retrieved from the Timestamp Server and displayed on the Administration GUI. Requests for changes to these parameters are signed with the Administrator's key and sent from the Administration GUI to the Timestamp Server engine for processing. Delete TSA: Allows a TOE Administrator to select a TSA to be deleted. A request signed with the Administrator's key is sent from the Administration GUI to the Timestamp Server engine for processing.
Generate Timestamp (DT_Generate)	The Timestamp Server only accepts timestamp requests when in the Timestamping mode of operation. The Timestamp Server generates a timestamp token according to the policy specified in the request. The resulting token contains: a) data that was timestamped b) time and date of timestamp c) policy OID. The TSA associated with the requested policy OID digitally signs the timestamp token.
Register Audit Key (AL_Register)	The Audit key and associated certificate is registered by a TOE Administrator as part of the Bootstrap process. The Audit key cannot be changed at a later time - only its location and passphrase may be modified via the Administration GUI. The Audit key is registered by entering the following details: a) key type (PKCS#12 or PKCS#11); b) passphrase (optional) c) key location details (depending on key

Security Function	Description
	storage type).
Events logging (AL_Logging)	The Timestamp Server logs all security related events to the Events log.
Protect Audit Records (AL_Integrity)	The Timestamp Server assures the integrity of each security related event record in the Events log, and the existence of the previous record, by signing each record with the Audit key and including in each security related audit record, the hash of the previous record (except the first). This is the hash of the previous security related record's data.
Protect Configuration File (PC_Integrity)	The Timestamp Server assures the integrity of the Configuration Parameters File by signing the file with the Audit key. The file is signed upon creation during the Bootstrap process and thereafter every time a change is made to the configuration of the Timestamp Server. The Timestamp Server verifies the Configuration Parameter File's signature every time the system starts. If the file was not signed by the Audit key, execution of the Timestamp Server ceases.
Protect Audit Key (PC_Audit_Key)	Replacement of the Audit key is prevented by the use of an Audit Key Vault Token (AKVT) - a salted hash of the Audit Key certificate created during the system bootstrap process. The Timestamp Server uses the AKVT to check the integrity of the Audit key every time the system starts. If the Audit key does not match the AKVT, execution of the Timestamp Server ceases.
Protect TSA Key(s) (PC_TSA_Key)	Replacement of the TSA key(s) is prevented by use of the Configuration File and the Audit Key that signs the Configuration file every time it is changed. The Configuration File contains the TSA key details from when the TSA key was configured (created or modified). On startup the TimeStamp Server extracts the certificate from the Key Tokens and checks that the certificate is the same as noted when the TSA key token that is in the Configuration File. If the TSA key does not match, execution of Timestamp Server ceases. The Timestamp Server assures the integrity of the Configuration via PC_Integrity.

Table 6.1 – IT Security Functions

6.2.2 Correspondence Between Security Functions and Requirements

The following table presents the correspondence between the functional specifications and the SFRs:

Security Functional Requirement	Security Function(s)
FAU_GEN.1	AL_Logging
FCO_NRO.2	IA_Authenticate IA_Register
FCS_CKM.3	IA_Authenticate DT_Manage DT_Generate AL_Integrity PC_Integrity PC_Audit_Key PC_TSA_Key
FCS_COP.1	IA_Authenticate DT_Manage DT_Generate AL_Integrity PC_Integrity PC_Audit_Key (Hash algorithm) PC_TSA_Key (Hash algorithm)
FIA_UAU.1	IA_Authenticate IA_Register
FIA_UID.1	IA_Register IA_Identify
FPT_ITT.1	IA_Authenticate IA_Register
FPT_ITT.3	IA_Authenticate IA_Register

Table 6.2 – SFR and Security Function correspondence

6.2.3 Strength of Functions

The TOE's security functions that are realised by probabilistic or permutational mechanisms are all cryptographic in nature and are therefore assessed by the National Authority. A statement of the minimum strength of function level is therefore not relevant for this TOE.

6.3 Assurance Measures

This section specifies the assurance measures of the TOE which are claimed to satisfy the stated assurance requirements. The assurance measures are traced to the assurance requirements so that it can be seen which measures contribute to the satisfaction of which requirements. This is done with reference to the appropriate documentation.

The assurance measures listed in Table 6.3 are required to be successfully evaluated in order for the TOE to be successfully certified.

Assurance Component	**Assurance Measure**
ASE_DES.1	Security Target document
ASE_ENV.1	Security Target document
ASE_INT.1	Security Target document
ASE_OBJ.1	Security Target document
ASE_PPC.1	Security Target document
ASE_REQ.1	Security Target document
ASE_SRE.1	Security Target document
ASE_TSS.1	Security Target document
ACM_CAP.3	Configuration Management Plan and configuration list
ACM_SCP.1	Configuration Management Plan
ADO_DEL.1	Delivery Procedures
ADO_IGS.1	Installation, Generation, and Start-up Procedures
ADV_FSP.1	Functional Specification
ADV_HLD.2	High-Level Design
ADV_RCR.1	Analysis of Correspondence
AGD_ADM.1	Administrator Guidance
AGD_USR.1	N/A as users interact with the TOE via client software that is not included in the scope of the evaluation
ALC_DVS.1	Development Security Documentation
ATE_COV.2	Analysis of Test Coverage (Test Documentation)
ATE_DPT.1	Analysis of Depth of Testing (Test Documentation)
ATE_FUN.1	Test documentation
ATE_IND.2	N/A as this is an evaluator action
AVA_MSU.1	Administrator Guidance

Assurance Component	Assurance Measure
AVA_SOF.1	Strength of Function Analysis
AVA_VLA.1	Vulnerability Analysis

Table 6.3 – Assurance Measures

7. Protection Profile Claims

This section contains the Protection Profile conformance claim statements.

7.1 Protection Profile Reference

No Protection Profile conformance claims are made.

7.2 Protection Profile Refinements

No Protection Profile conformance claims are made.

7.3 Protection Profile Additions

No Protection Profile conformance claims are made.

8. Rationale

8.1 Introduction

8.1.1 This section presents evidence that supports the claims that the Security Target is a complete and cohesive set of requirements, that a conformant TOE would provide an effective set of IT security countermeasures within the security environment, and that the TOE summary specification addresses the requirements. The rationale also demonstrates that any protection profile claims are valid.

8.2 Security Objectives Rationale

8.2.1 This section demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them.

8.2.2 Table 8.1 below maps the threats, assumptions and organisation security policies against the TOE security objectives which are intended to address them. Table 8.2 presents a similar mapping for the Environmental security objectives identified in Section 4.3. These tables show that each security objective covers at least one threat, assumption or policy and that each threat, assumption and policy (identified in Chapter 3) is covered by at least one security objective. All security objectives are thus shown to be necessary.

Security Objective	Threat/Assumption/Policy	
O.Ident_Authent	T.Hack_Imperson_Admin	(Threat)
O.Timestamp	T.Hack_Imperson_TOE	(Threat)
	T.Hack_Mod_Timestamp	(Threat)
	T.Client_Refute_Origin	(Threat) (Policy)
O.Audit	T.Config_Mod_Undetect	(Threat)
	T.Replace_Audit_Key	(Threat)
	T.Replace_TSA_Key	(Threat)
	T.Log_Mod_Undetect A.TOE_Administrator	(Threat) (Assumption)

Security Objective	Threat/Assumption/Policy	
O.Integrity_Config	T.Mod_Config_Data	(Threat)
	T.Replace_Audit_Key	(Threat)
	T.Replace_TSA_Key	(Threat)

Table 8.1 – TOE Security Objectives

Security Objective	Threat/Assumption/Policy	
OE.Time_Source	A.Time_Source	(Assumption)
	T.Replace_Audit_Key	(Threat)
	T.Config_Mod_Undetect	(Threat)
OE.PKI	A.PKI	(Assumption)
OE.Cryptography	P.Cryptography	(Policy)
	T.Client_Refute_Origin	(Threat)
	T.Hack_Imperson_TOE	(Threat)
	T.Hack_Mod_Timestamp	(Threat)
	T.Log_Mod_Undetect	(Threat)
	T.Mod_Config_Data	(Threat)
OE.Key_Generation_ Destruction	P.Key_Generation_Destruct ion	(Policy)
OE.Passphrases_PINs	P.Passphrases_PINs	(Policy)
	T.Hack_Imperson_Admin	(Threat)
OE.Key_Storage	A.Key_Storage	(Assumption)
	T.Hack_Imperson_Admin	(Threat)
OE.Physical	T.Hack_Imperson_Admin	(Threat)
	A.Location	(Assumption)
	T.Replace_Audit_Key	(Threat)
OE.Connectivity	A.Connectivity	(Assumption)
OE.System_Administrator	A.System_Administrator	(Assumption)
OE.TOE_Administration	A.TOE_Administrator	(Assumption)
	T.Log_Mod_Undetect	(Threat)
OE.TS_Requestor	A.TS_Requestor	(Assumption)
OE.Audit_Log	T.Config_Mod_Undetect	(Threat)
	T.Replace_Audit_Key	(Threat)
	T.Replace_TSA_Key	(Threat)

Security Objective	Threat/Assumption/Policy	
OE.P11_Device	A.P11_Device	(Assumption)
	P.Cryptography	(Policy)
	T.Client_Refute_Origin	(Threat)
	T.Hack_Imperson_TOE	(Threat)
	T.Hack_Mod_Timestamp	(Threat)
	T.Log_Mod_Undetect	(Threat)
	T.Mod_Config_Data	(Threat)

Table 8.2 – Environmental Security Objectives

8.2.3 The following sections demonstrate that the security objectives are sufficient to meet the security needs of the TOE. Each threat, assumption and policy is considered in turn.

Threats

T.Hack_Imperson_Admin

This threat is countered by O.Ident_Authent, OE.Passphrases_PINs, OE.Key_Storage and OE.Physical. O.Ident_Authent ensures that all TOE Administrators are identified and authenticated before modifications to the TOE's security data can occur. OE.Passphrases_PINs reduces the possibility of a person other than an authorised administrator from guessing or otherwise obtaining the passphrase or PIN of the private key belonging to an administrator. OE.Key_Storage contributes to ensuring that private keys are kept confidential. OE.Physical prevents unauthorised access to the TOE and its associated hardware and software.

T.Hack_Imperson_TOE

This threat is completely addressed by O.Timestamp (supported by OE.Cryptography and or OE.P11_Device) which provides the means for timestamp tokens to be produced that are associated with the TOE in a manner which cannot be easily duplicated, and the likelihood of such duplication is reduced to an acceptable level.

The PKI standards implemented ensure that such an attack is computationally complex.

T.Hack_Mod_Timestamp

This threat is completely addressed by O.Timestamp (supported by OE.Cryptography and or OE.P11_Device) which provides the means for protecting the integrity of timestamp tokens in a manner where the likelihood of undetected tampering is reduced to an acceptable level.

The PKI standards implemented ensure that such an attack is computationally complex.

T.Client_Refute_Origin

This threat is completely addressed by O.Timestamp (supported by OE.Cryptography and or OE.P11_Device) which ensures that the TOE's identity can be authenticated using evidence included in a timestamp token generated by the TOE. The likelihood of such evidence being altered is reduced to an acceptable level.

The PKI standards implemented ensure that such an attack is computationally complex.

T.Config_Mod_Undetect

This threat is addressed by O.Audit supported by OE.Audit_Log. O.Audit provides the means for all changes to the TOE's configuration to be recorded. OE.Time_Source provides a reliable timestamp to assist in the time-based analysis of the audit records. OE.Audit_Log ensures that the audit facilities are managed effectively.

T.Log_Mod_Undetect

This threat is countered by OE.TOE_Administrator and O. Audit supported by OE.Cryptography or OE.P11_Device.

OE.TOE_Administrator assumes the TOE Administrator regognises the importance of the audit log and reduces the possibility of the TOE Administrator inadvertently modifying the audit log.

O.Audit provides the means to verify the integrity of each security related event log record, and the existence of the previous security related record, using integrity checking mechanisms based on algorithms provided by OE.Cryptography for PKCS#12 files or OE.P11_Device when using PKCS#11 devices.

The PKI standards implemented ensure that the likelihood of the modification not being detected by the mechanism is reduced to an acceptable level.

T.Mod_Config_Data

This threat is countered by O.Integrity_Config supported by OE.Cryptography or OE.P11_Device. O.Integrity_Config provides the means for the configuration file to be protected from modification using integrity checking mechanisms based on algorithms provided by OE.Cryptography for PKCS#12 files or OE.P11_Device when using PKCS#11 devices.

The PKI standards implemented ensure that the likelihood of the modification not being detected by the mechanism is reduced to an acceptable level.

T.Replace_Audit_Key

This threat is partly countered by O.Audit supported by OE.Audit_Log. O.Audit provides the means for all discrepancies between the current audit key and the audit key details entered during the bootstrap process to be recorded. OE.Time_Source provides a reliable timestamp to assist in the time-based analysis of the audit records. OE.Audit_Log ensures that the audit facilities are managed effectively. OE.Physical prevents unauthorised access to the TOE and its associated hardware and software.

The threat is also countered by O.Integrity_Config which detects when the audit key is different from that entered during the bootstrap stage and if it is, stops execution of the Timestamp Server.

T.Replace_TSA_Key	The threat is countered by O.Integrity_Config. O.Integrity_Config detects when the TSA key is different from that entered during configuration stage. If there is a difference it sends a message to the TOE Administrator and enters the configuration mode and stops execution of the Timestamp Server. Additionally this threat is addressed by O.Audit supported by OE.Audit_Log. O.Audit provides the means for all changes to the TSA keys and tokens to be recorded. Changes to the TSA_Keys is an security related event recorded by O.Audit. OE.Audit_Log ensures that the audit facilities are managed effectively.

8.2.4 Assumptions

A.Time_Source

OE.Time_Source upholds this assumption by giving TOE owners the responsibility for ensuring that a time source acceptable to them is available for timestamping.

A.PKI

OE.PKI upholds this assumption by giving TOE owners the responsibility for ensuring that the TOE operates within a securely managed PKI.

A.Key_Storage

OE.Key_Storage upholds this assumption by giving TOE owners the responsibility to enforce procedures for the secure storage of private keys used in relation to the TOE. This includes protection of PKCS12 file and PKCS11 devices.

A.Location

OE.Physical upholds this assumption by specifying the protection of the physical aspects of the TOE. This includes protection of PKCS12 file and PKCS11 devices.

A.Connectivity

OE.Connectivity upholds this assumption by specifying the protection of the TOE (and all associated components of the Timestamp Server as set out in Section 2.4) from external connections originating on the untrusted external network. This includes protection of PKCS12 file and PKCS11 devices.

A.System_Administrator

OE.System_Administrator upholds this assumption by specifying the prerequisite qualities, knowledge and skills expected for System Administrators.

A.TOE_Administrator

OE.TOE_Administrator upholds this assumption by specifying the prerequisite qualities, knowledge and skills expected for TOE Administrators.

A.TS_Requestor

OE. TS_Requestor upholds this assumption by specifying the responsibilities of those requesting timestamps from the TOE.

A.P11_Device

OE.P11_Device upholds this assumption by giving the TOE owners the responsibility to select an appropriate hardware cryptographical device for use with the TOE.

8.2.5 Organisation Security Policies

P.Cryptography

OE.Cryptography provides coverage of this policy, when not using hardware based cryptographical modules, by specifying that the TOE owners are responsible for ensuring that the cryptographic algorithms implemented by the TOE have been approved by the National Authority.

	OE.P11_Device provides coverage of this policy when using hardware based cryptographical modules, by specifying that the TOE owners are responsible for ensuring that the cryptographic algorithms implemented by the device have been approved by the National Authority.
P.Key_Generation_Destruction	OE.Key_Generation_Destruction provides full coverage of this policy by specifying that TOE owners are responsible for ensuring that the method of key generation and key destruction for all cryptographic keys and certificates used by the TOE have been approved by the National Authority.
P.Passphrases_PINs	OE.Passphrases_PINs provides full coverage of this policy by specifying that TOE owners are responsible for ensuring that procedures exist for the management of passphrases and PINs protecting the private keys used with the TOE and that the requirements set by the National Authority are followed.

8.3 Security Requirements Rationale

8.3.1 This section demonstrates that the set of security requirements (TOE and environment) is suitable to meet and is traceable to the security objectives.

8.3.2 It demonstrates the following:

a) that the combination of the individual functional and assurance requirements components for the TOE and its IT environment together meet the stated security objectives

b) that the set of security requirements together form a mutually supportive and internally consistent whole

c) that the choice of security requirements is justified (including non-satisfaction of dependencies)

d) that the selected strength of function level for the Security Target, together with any explicit strength of function claim, is consistent with the security objectives for the TOE.

8.3.3 Security Functional Requirements Rationale

8.3.3.1 The following table maps each TOE security objective against the corresponding security functional components. It demonstrates that each security objective for the TOE is addressed by at least one SFR and that each SFR addresses at least one security objective.

Security Objective	**Security Functional Requirement**	
O.Ident_Authent	FCO_NRO.2	Enforced Proof of Origin

Security Objective	Security Functional Requirement	
	FCS_CKM.3	Cryptographic Key Access
	FCS_COP.1	Cryptographic Operation
	FIA_UAU.1	Timing of Authentication
	FIA_UID.1	Timing of Identification
	FPT_ITT.1	Basic Internal TSF Data Transfer
	FTP_ITT.3	TSF Data Integrity Monitoring
O.Timestamp	FCS_CKM.3	Cryptographic Key Access
	FCS_COP.1	Cryptographic Operation
O.Audit	FAU_GEN.1	Audit Data Generation
	FCS_CKM.3	Cryptographic Key Access
	FCS_COP.1	Cryptographic Operation
O.Integrity_Config	FCS_CKM.3	Cryptographic Key Access
	FCS_COP.1	Cryptographic Operation

Table 8.3 – Security Objectives mapped to SFRs

8.3.3.2 For each security objective, informal arguments are provided as to why the identified SFRs are sufficient to satisfy the objective.

O.Ident_Authent

FCO_NRO.2 is used to enforce the requirement that TOE Administrators identify themselves when requesting changes to the configuration of the TOE.

FCS_CKM.3 specifies the mechanism(s) used in loading private keys used in the administration of the TOE.

FCS_COP.1 contributes to meeting the objective by requiring that the TOE conforms to recognised digital signature standards.

FIA_UAU.1 specifies that a TOE Administrator may not modify any configuration data, except at bootstrap when Audit Key and List of administrators is added, the creation of the events log, and the starting of the TSS Server, until they have been successfully authenticated by the TOE.

FIA_UIU.1 specifies that a TOE Administrator must identify themselves to the TOE before any further interactions, other than at bootstrap when Audit Key and List of administrators is added, the creation of the events log, and the starting of the TSS Server, are allowed.

FPT_ITT.1 supports the objective by protecting administration requests from modification as they are transferred from one part of the TOE to another.

FTP_ITT.3 supports the objective by preserving the integrity of administration requests as they are transmitted between separate parts of the TOE and specifies that no action should be taken upon detection of a data integrity error.

O.Timestamp

FCS_CKM.3 specifies the mechanism(s) used in loading private keys used in the operation of the TOE.

FCS_COP.1 contributes to meeting the objective by requiring that the TOE conforms to recognised digital signature standards. This function also contributes to meeting the objectives by associating and binding the timestamp token with the TSA that produced it. The binding and association is performed by the contents of the timestamp token which is protected by the digital signature.

O.Audit	FAU_GEN.1 identifies the auditable events for which audit records should be generated and specifies the information to be provided in the audit records. FCS_CKM.3 specifies the mechanism(s) used in loading private keys used in the administration of the TOE. FCS_COP.1 contributes to meeting the objective by requiring that the TOE conforms to recognised digital signature and hashing standards.
O.Integrity_Config	FCS_CKM.3 specifies the mechanism(s) used in loading private keys used in the administration of the TOE. FCS_COP.1 contributes to meeting the objective by requiring that the TOE conforms to recognised digital signature and hashing standards.

8.3.4 Security Assurance Requirements Rationale

The target evaluation level of CC EAL 3 is sufficiently high given the identified threats and security objectives. In particular it considers the vulnerabilities that may be exploited by external threat agents in the vulnerability analyses that are not included in lower assurance levels. The TOE has been developed in a manner to ensure that CC EAL 3 is attainable.

8.3.5 Strength of Function Level Rationale

The TOE's security functions that are realised by probabilistic or permutational mechanisms are all cryptographic in nature and are therefore assessed by the National Authority. A statement of the minimum strength of function level is therefore not relevant for this TOE.

8.3.6 Dependency Rationale

8.3.6.1 The following table summarises how the dependencies among SFRs are satisfied. The first column is used to identify individual rows. The second column lists all SFRs that contribute to the TOE security objectives. The next column contains the dependencies on each SFR. The last column references the row that refers to the dependency or includes an explanation as to why the dependency does not need to be satisfied.

ID	SFR	Dependency	Satisfied by
1	FAU_GEN.1	FPT_STM.1	OE.Time_Source
2	FCO_NRO.2	FIA_UID.1	6
3	FCS_CKM.3	FCS_CKM.1	OE.Key_Generation_

ID	SFR	Dependency	Satisfied by
			Destruction
		FCS_CKM.4	OE.Key_Generation_ Destruction
		FMT_MSA.2	OE.Passphrases_PINs OE.Key_Storage
4	FCS_COP.1	FCS_CKM.1	OE.Key_Generation_ Destruction
		FCS_CKM.4	OE.Key_Generation_ Destruction
		FMT_MSA.2	OE.Passphrases_PINs OE.Key_Storage
5	FIA_UAU.1	FIA_UID.1	6
6	FIA_UID.1	None	N/A
7	FPT_ITT.1	None	N/A
8	FPT_ITT.3	FPT_ITT.1	7

Table 8.4 – SFR dependency analysis

8.3.6.2 Four dependencies are not directly satisfied: that of FAU_GEN.1 on FPT_STM.1, FCS_CKM.3 and FCS_COP.1 on FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2.

a) FAU_GEN.1 refers to the requirement of the TOE to have reliable timestamps for its own use, in order to put timing information in its audit logs. This dependency is not required as the environmental security objective OE.Time_Source provides a reliable time source for timestamps.

b) FCS_CKM.3 refers to the requirement of the TOE to access cryptographic keys in order to perform digital signing operations via FCS_COP.1. Both of these components have a dependency on FCS_CKM.1 (Cryptographic Key Generation) and FCS_CKM.4 (Cryptographic Key Destruction). These dependencies are not required as the activities are performed externally to the TOE. This is addressed by the environmental security objective OE.Key_Generation_Destruction.

c) FCS_CKM.3 and FCS_COP.1 also have a dependency on FMT_MSA.2 (Secure security attributes), in relation to cryptographic key access. This dependency is not required for the TOE as the management of security attributes is fully addressed by environmental security objectives OE.Passphrases_PINs and OE.Key_Storage.

8.3.7 Mutually Supportive Security Requirements Rationale

8.3.7.1 The security requirements are mutually supporting as all requirements are based purely on the CC part 2 and all dependencies have been addressed. The set of SFRs are internally consistent and include SFRs that defend other SFRs against attacks such as bypassing or tampering.

8.3.7.2 The internal consistency of the security requirements is demonstrated by considering how they fall under the following categories:

a) **Audit** - All of the audit SFRs relate to the same set of data, namely the auditable events. These events are recorded in a events log, in which the integrity of each security related record and the existence of the previous security related record are assured using the SFRs from Cryptographic Support. Analysis and review of the audit data is beyond the scope of the TOE. Only Authenticated Administrators are permitted to configure and modify the configuration of the TOE so there are no conflicting requirements.

b) **Communication** - The communication SFR relates to the non-repudiation of origin and is used to support the authentication of TOE Administrators. There are no instances where this SFR applies to other SFRs in a way where potential conflicts may arise.

c) **Cryptographic Support** - The cryptographic support SFRs specify the requirements for access to cryptographic keys and the algorithms used for cryptographic operations related to the integrity and authentication of data. These SFRs support the protection of the events log, enforced proof of origin (FCO_NRO.2) and internal transfer of TSF data (FPT_ITT.1 and FPT_ITT.3). There are no potential conflicts with the remaining TOE SFRs.

d) **Identification and Authentication** - The identification and authentication SFRs describe a number of rules for the identification and authentication of users by the TOE. These rules are specified in FIA_UAU.1 and FIA_UID.1. There are no instances where one of these identification and authentication SFRs applies to other SFRs in a way where potential conflicts may arise. These rules require that only TOE Administrators are required to be identified and authenticated before being granted access to the TOE. This does not conflict with the audit requirement as, with the exception of startup and shutdown, only TOE Administrators can execute auditable events. Startup and shutdown are events that have been designed to executed by non administrators and association of these events with an individual is not required. It should be noted that the person starting the TOE is required to know audit and TSA key passphrases in order to make the system operational. This knowledge acts as a substitute for Identification and Authentication. Unauthorised shutdown is not regarded as a threat and prevention by environmental measures is regarded as satisfactory.

e) **Protection of the TSF** - The protection of the TSF SFRs describe how the integrity of data transferred between separate parts of the TOE is preserved, and specifies the requirement for a reliable timestamp capability. The former SFRs support the authentication of TOE Administrators while the latter supports the audit function. There are no potential conflicts with any other TOE SFR.

8.3.7.3 Mutual support by SFRs that prevent bypassing of other SFRs is implemented by FIA_UID.1 and FIA_UAU.1 which identify and authenticate TOE Administrators and work to prevent the impersonation of a TOE Administrator. They require users to be identified and authenticated before allowing them to perform actions on the TOE. The remaining SFRs are always invoked when necessary and hence cannot be bypassed if the SFR is satisfied by the TSF.

8.3.7.4 Mutual support by SFRs that prevent anyone tampering with other SFRs is implemented by audit generation requirements FAU_GEN.1 which specifies the events recorded by the TOE in the events log. The TOE cannot protect itself from direct physical attacks and its physical protection is therefore specified in terms of a environmental security objectives OE.Physical and OE.Key_Storage.

8.3.7.5 Mutual support by SFRs that prevent deactivation of other SFRs is not relevant to the TOE as no SFR can be deactivated. The effect on the audit function resulting from the events log filling, and or the audit records being deleted, are addressed as an environmental security objective OE.Audit_Log.

8.3.7.6 Mutual support by SFRs that enable the detection of the misconfiguration of another SFR or of attack aimed at defeating another SFR is not relevant as the SFRs represent inherent functionality of the TOE that is invoked when necessary and cannot be configured or deactivated in a way that can affect how the security objectives are met.

8.4 TOE Summary Specification Rationale

8.4.1 This section demonstrates that the TOE security functions and assurance measures are suitable to meet the TOE security requirements.

8.4.2 It demonstrates the following:

a) that the combination of specified TOE IT security functions work together so as to satisfy the TOE security functional requirements;

b) that the strength of TOE function claims made are valid, or that assertions that such claims are unnecessary are valid

c) that the claim is justified that the stated assurance measures are compliant with the assurance requirements.

8.4.3 IT Security Functions Rationale

A mapping of IT Security Functions onto SFRs is provided in Table 6.2. It demonstrates that each SFR is mapped onto at least one IT Security Function and that each IT Security Function is mapped onto at least one SFR.

FAU_GEN.1

This component is satisfied by AL_Logging. AL_Logging logs all security related events to the Events log.

FCO_NRO.2

This component is satisfied by IA_Authenticate which depends upon the successful execution of IA_Register. IA_Authenticate causes all requests originating from the Administration GUI to be digitally signed using the TOE Administrator's private key. It also causes the Timestamp Server to verify the signature and authenticate the administrator prior to carrying out the request. Administrators must have first been registered via IA_Register.

FCS_CKM.3

This component is satisfied by aspects of IA_Authenticate, DT_Manage, DT_Generate, AL_Integrity, PC_Integrity, PC_Audit_Key and PC_TSA_Key which all utilise cryptographic services.

FCS_COP.1

This component is satisfied by aspects of IA_Authenticate, DT_Manage, DT_Generate, AL_Integrity, PC_Integrity, PC_Audit_Key and PC_TSA_Key which all perform cryptographic operations in accordance with the PKCS standards. These PKCS standards have been selected as representing industry best practice and for maximum interoperability.

FIA_UAU.1

This component is satisfied by IA_Authenticate which depends upon the successful execution of IA_Register. IA_Authenticate relates to the TOE's enforcement of user authentication before allowing any other TSF-mediated actions on behalf of that user. TOE Administrators must have first been registered via IA_Register.

FIA_UID.1

This component is satisfied by IA_Identify which depends upon the successful execution of IA_Register. IA_Identify relates to the TOE's enforcement of user identification before allowing any other TSF-mediated actions on behalf of that user. TOE Administrators must have first been registered via IA_Register.

FPT_ITT.1

This component is satisfied by IA_Authenticate and IA_Register. IA_Authenticate causes all requests originating from the Administration GUI to be digitally signed using the TOE Administrator's private key. The Timestamp Server then verifies this signature and authenticates the administrator prior to carrying out the request. The set of Administrators that may administer the TOE are defined by IA_Register.

FPT_ITT.3

This component is satisfied by IA_Authenticate and IA_Register. IA_Authenticate causes all requests originating from the Administration GUI to be digitally signed using the TOE Administrator's private key. The Timestamp Server then verifies this signature and authenticates the administrator prior to carrying out the request. The set of administrators that may administer the TOE are defined by IA_Register.

8.4.4 Strength of Function Claim Rationale

The TOE's security functions that are realised by probabilistic or permutational mechanisms are all cryptographic in nature and are therefore assessed by the National Authority. A statement of the minimum strength of function level is therefore not relevant for this TOE.

8.4.5 Mutually Supportive IT Security Functions Rationale

The TOE Summary Specification does not introduce any changes to the dependency and mutual support argument presented for SFRs.

8.4.6 Security Assurance Measures Rationale

The security assurance requirements of EAL 3 is achievable for the following reasons:

a) all documentation and other resources required by this assurance level as shown in Table 6.3 will be made available

b) the documents have been produced to fulfil the criteria of this assurance level

c) the TOE has been developed to achieve a high degree of security

d) the TOE was developed in a secure manner.

8.5 PP Claims Rationale

No Protection Profile conformance claims have been made.

Version History

Version No.	Details	Date of change	Author
1.0	Release	September 2000	GMG/Admiral
1.1	Draft	31 October 2000	H. Mullenger
1.2	Released	14 November 2000	G Sarandrea
2.0.1a	Draft – Modified to combine Version 1.0 paragraph numbering and make the version numbering and formats in line with Product documents. This is the revision after 1.2, and also includes EOR draft fixes.	13 December 2000	G Sarandrea
2.0.1b	Release – Modifications incl removal of FAU_GEN.2 and weak policy undetect. All current EORs addressed this issue.	10 January 2001	G Sarandrea
2.0.1c	Release – Modifications made by admiral re consistent wording.	16 January 2001	G Sarandrea
2.0.1d	Release – Modified to make consitent between FS and HLD.	24 January 2001	G Sarandrea
2.0.1e	Release – Modifications made by Admiral	29 January 2001	GMG/Admiral
2.0.2a	Release – Modifications due to EORs and release of TSS 2.0.2	9th April 2001	G Sarandrea
2.0.2b	Release – Modifications to fix EOR 45 and 53.	23th April 2001	G Sarandrea
2.0.2c	Release – Modifications to Fix EOR 50, 54, 55	3rd May 2001	G Sarandrea
2.0.2d	Release – Modifications to Fix EOR 62	10th July 2001	G Sarandrea
2.0.2e	Release – Added updates due to patch release	18th Sept 2001	G Sarandrea
2.0.2f	Release – Limit to Keyper 2.1	22nd Oct 2001	G Sarandrea
2.0.2g	Release – to add TSS patch and 2.0.2 to the TOE definition.	6th Nov 2001	G Sarandrea
2.0.2h	Final Release, to clear EORs/RFC	3rd December 2001	G Sarandrea
2.0.2i	Release for consistency with other documents	10th Dec 2001	G Sarandrea
2.0.2j	Released comments raised by DSD	15 May 2002	G Sarandrea
2.0.2k	Released. Corrections to the way the audit records are created	12th Dec 2002	G Sarandrea
2.0.2l	Release - change of copyright	03rd Oct 2003	U Brell