UniCERT

Security Target

Serial Number	490
Release	3.10
Status	Release
Issue Date	10th November 2000

1 Table of Contents

1 Introduction

1.1 Purpose

This document is the Security Target for UniCERT, which is an enabling technology for implementing Public Key Infrastructures (PKI). It provides a complete and consistent statement of the Security Enforcing Functions (SEF) and mechanisms of UniCERT (the Target of Evaluation (TOE)).

The Security Target is the baseline document for a formal security evaluation under the Australian Information Security Evaluation Program (AISEP). It fulfils the requirements of the Information Technology Security Evaluation Criteria [ITSEC]. The evaluation is being sponsored by Baltimore Technologies Pty Limited. UniCERT was developed at Baltimore's Laboratories in Dublin, Ireland.

The Security Target details the TOE's security requirements and the countermeasures proposed to address the perceived threats to the assets protected by the TOE.

The Security Target is also intended to provide a more detailed technical description of the intended operational environment, security objectives and security functionality of the TOE for those responsible for managing, purchasing, installing, configuring, operating and using PKIs.

This Security Target has been produced in accordance with guidance provided in the Information Technology Security Evaluation Methodology [ITSEM] and assistance from Admiral Management Services.

The role of the Security Target within the development and evaluation process is further described in the Information Technology Security Evaluation Criteria [ITSEC].

It is assumed that the reader already has an understanding of what a PKI is and of the related concepts.

1.2 Scope

Section 2 is the Product Rationale, which contains a description of the product, the security objectives of the product, a list of assumed threats the product is designed to counter and the security related assumptions on the operating environment in which the product is designed to operate.

Section 3 contains a description of the Security Enforcing Functions (SEFs) of the product, as well as a description of how these functions meet the objectives listed in Section 2.

Section 4 defines the claimed minimum strength of the security mechanisms and the target evaluation level for the product.

1.3 Terminology

This section contains definitions of technical terms that are used within this document.

ARL	Authority Revocation List
Authorised	A person or entity authorised to carry out an action or be in possession of certain information according to a procedure defined by the customer. This can include the legitimate holder of a private key and corresponding certificate which has been issued by the CA.
Authorised certificate	A certificate which has been granted according to the rules defined by the CA.
Authority Revocation List	A signed list of certificates belonging to PKI components which have been revoked (as defined in [X.509v3])
CA	Certification Authority
CAO	Certification Authority Operator (application)
CAO User	Person who operates the CAO and CA
Certificate	See definition of public key certificate.
Certificate hierarchy	A system of certification authorities and registration authorities that verify and authenticate parties involved in a transaction.
Certificate Revocation List	A signed list of certificates that have been revoked (according to the standard for CRLv2 as defined in [X.509v3]).
Certification Authority	The component within the TOE which is responsible for the creation, distribution, or revocation of X.509 public key certificates
CRL	Certificate Revocation List
Customer	The organisation or individual having legal responsibility for running the TOE.
CRP	Customer Registration Policy. Customer Registration Policies are set up by CAO users. They define how the RAO user is to register the end user and they control the information which goes into a public key certificate for an end user. The CAO users control which RAO users can use which Customer Registration Policies. The Customer Registration Policy consists of a template that defines what registration information is to be collected or checked by the RAO user, how the users keys are to be generated and the key lengths and algorithms that are allowable, and the certificate extensions that are to be included in the certificate. CRPs are stored as blobs in the ORACLE database.

DAP	The Directory Access Protocol (specified in [DAP]) which allows information to be posted to an X.500 directory.
end user	An entity that uses public key certificates which have been generated by the TOE to secure files or messages (using a product such as MailSecure from Baltimore).
face-to-face registration	The process of entering end user details at the RAO directly, without a remote request coming in through the gateway.
Gateway	The component within the PKI which accepts remote e-mail and WWW originated certification requests from end users.
Interface to the TOE	The logical connection to the TOE which allows a remote component to communicate with the TOE.
IV	Initialisation Vector. See [DES] for further details.
LDAP	The Lightweight Directory Access Protocol (specified in [LDAP]) which allows information to be posted to an X.500 directory.
Operational Policy	The operational policy defines the operation of a PKI entity e.g. the CA operational policy defines certain parameters, such as frequency of CRL generation.
Public key certificate	A X.509v3 certificate which binds the user's public key to its identity (see [X.509v3] for details); also known as certificate.
PKI	Public Key Infrastructure.
PKI component	One of either CA, CAO, RA, RAO within the PKI structure.
.pse file	The file containing the component's private key(s) and certificate(s), as well as other information including the CA's certificate(s) where appropriate. Note this may be made up of one or more components, all of which are required to generate the complete .pse file.
RA	Registration Authority
RA Administrator	Person who operates the RA.
RAO	Registration Authority Operator (application)
RAO User	Person who operates the RAO
Registration Authority	The component within the TOE which is responsible for the registration of a defined group of end users.
Remote entity	An entity able to send information to, or receive information from, any of the interfaces to the TOE platform.

Revocation	The process of invalidating a public key certificate. There are a number of reasons for revocation, including: unspecified, keyCompromise, cACompromise, affiliationChanged, superseded and certificatehold. Certificatehold places a certificate on hold, referred to as suspension of a certificate in this document. With the exception of certificatehold, all other reasons for revocation are permanent, which means the certificate will no longer be valid.
Revocation passphrase	A secret value optionally defined during the registration process, which must be included in any remote revocation request for that certificate.
Revocation Request	Revocation requests include requests to revoke, suspend and unsuspend a certificate.
Revoke	To invalidate a certificate.
root CA	The Certification Authority at the top of the public key Hierarchy.
sub CA	A Certification Authority which is below the level of the root CA.
SEF	Security Enforcing Function
System Administrator	The person responsible for maintaining the systems necessary for the smooth running of UniCERT, including Windows NT, the Oracle database, communications lines etc.
Suspension	The temporary revocation of a certificate. Once a certificate has been suspended it can be handled in one of three ways: • It may remain on the CRL with no further action, causing users to reject transactions issued during the hold period; or, • It may be replaced by a (final) revocation for the same certificate, in which case the reason shall be one of the standard reasons for revocation, the revocation date shall be the date the certificate was suspended; or • It may be explicitly released and the entry removed from the CRL.
Trusted certificate	A certificate which has been created by an authorised CA.
TOE	Target of Evaluation
User ID	Unique identifier for a user.
Windows NT	Version 4.0 of the Microsoft NT operating system

Unsuspension	Removing the temporary hold (suspension) of a certificate and therefore removing it from the CRL.
Valid certificate	A certificate which is both trusted and authorised.
X.500 directory	A publicly available directory, as defined in [X.500], containing X.509 public key certificates.
WWW	World Wide Web

1.4 References

1.4.1 Standards and Papers

[DAP] ISO / IEC 9594-5: 1995 *Information Technology — Open Systems Interconnection — Part 5 The Directory: Directory Access Protocols*

[DES] FIPS PUB 47 *Data Encryption Standard*, November 23, 1976

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

[ITSEC] Information Technology Security Evaluation Criteria, Provisional Harmonised Criteria, Version 1.2, Commission of the European Communities, 28 June 1991

[LDAP] RFC1777 Lightweight Directory Access Protocol, March 1995

[MD5] R.L.Rivest *The MD5 Message Digest Algorithm, RFC 1321* April 1992,

[PKCS#10] RSA Laboratories, *PKCS#10 – Certification Request Syntax Standard*, Version 1.5, November 1993

[PKCS#12] RSA Laboratories, *PKCS#12 – Personal Information Exchange Syntax Standard*, Version 1.0 DRAFT, April 1997

[PKIX] S. Farrell and C. Adams, *Internet Public Key Infrastructure, Part III: Certificate Management Protocols*, Internet Draft, December 1996.

[RNG] CESG X11 Random Number Generator for DSA

[RSA] RSA Laboratories, *PKCS #1: RSA Encryption Standard*, Version 1.5 Revised November 1, 1993

[RSA_KeyGen] ANSI, *Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (X9.31)* - Publication Date: 1998

[X.500] ISO / IEC 9594: 1995 *Information Technology — Open Systems Interconnection —The Directory: Series of standards*

[X.509v3] Final Text of Draft Amendments DAM 1 to ITU Rec. X.509 (1993) | ISO/IEC 9594-8 : 1995 *Information Technology — Open Systems Interconnection —The Directory: Authentication Framework.*

1.4.2 Baltimore Documents

[UN-SA] Baltimore Technologies Pty Limited, *UniCERT: ITSEC Suitability Analysis,* Version 1.5, 18th April 2000.

[UN-DD] Baltimore Technologies Pty Limited, *UniCERT: ITSEC Detailed Design,* Version 1.2, 19th April 2000.

2 Product Rationale

2.1 Introduction

This section contains the product rationale. It contains a brief description of the TOE — what it is used for, what platform it runs on and what other products it is designed to interact with.

2.2 Product Description

UniCERT provides all the functionality needed to implement a PKI system: essentially a system that provides registration, PKI management and certification authority functions. This can then be used to manage all the keys necessary for a system requiring security for end users, such as a secure messaging system e.g. Baltimore's MailSecure, or security on Web browsers. UniCERT provides the ability to set up a centralised or a distributed PKI for organisations of any size. UniCERT is capable of operating with a single level or multilevel certification authority to suit the structure, function and geography of the organisation. UniCERT also provides centralised management of the PKI that was set up.

There are a number of different components within UniCERT. The main components are the Certification Authority (CA), the CA Operator (CAO), the Registration Authority (RA), the RA Operator (RAO), the Token Manager and the Gateway. These components are described in more detail in Section 2.2.1. UniCERT allows the customer to separate components onto different hardware systems in order to distribute the workload to different corporate departments, sections etc. thus reducing bottlenecks in processing.

UniCERT generates X.509 certificates for end users and for the different components within the PKI. It can be used to revoke certificates and to generate certificate revocation lists (CRLs) and authority revocation lists (ARLs). Both certificates, CRLs and ARLs can be published on an X.500 directory using the Lightweight Directory Access Protocol (LDAP). In the evaluated configuration the X.500 directory will be hosted on a separate machine. Although other configurations are possible these will not be evaluated.

Some examples of how UniCERT could be used to implement a PKI are shown in Figure 1.

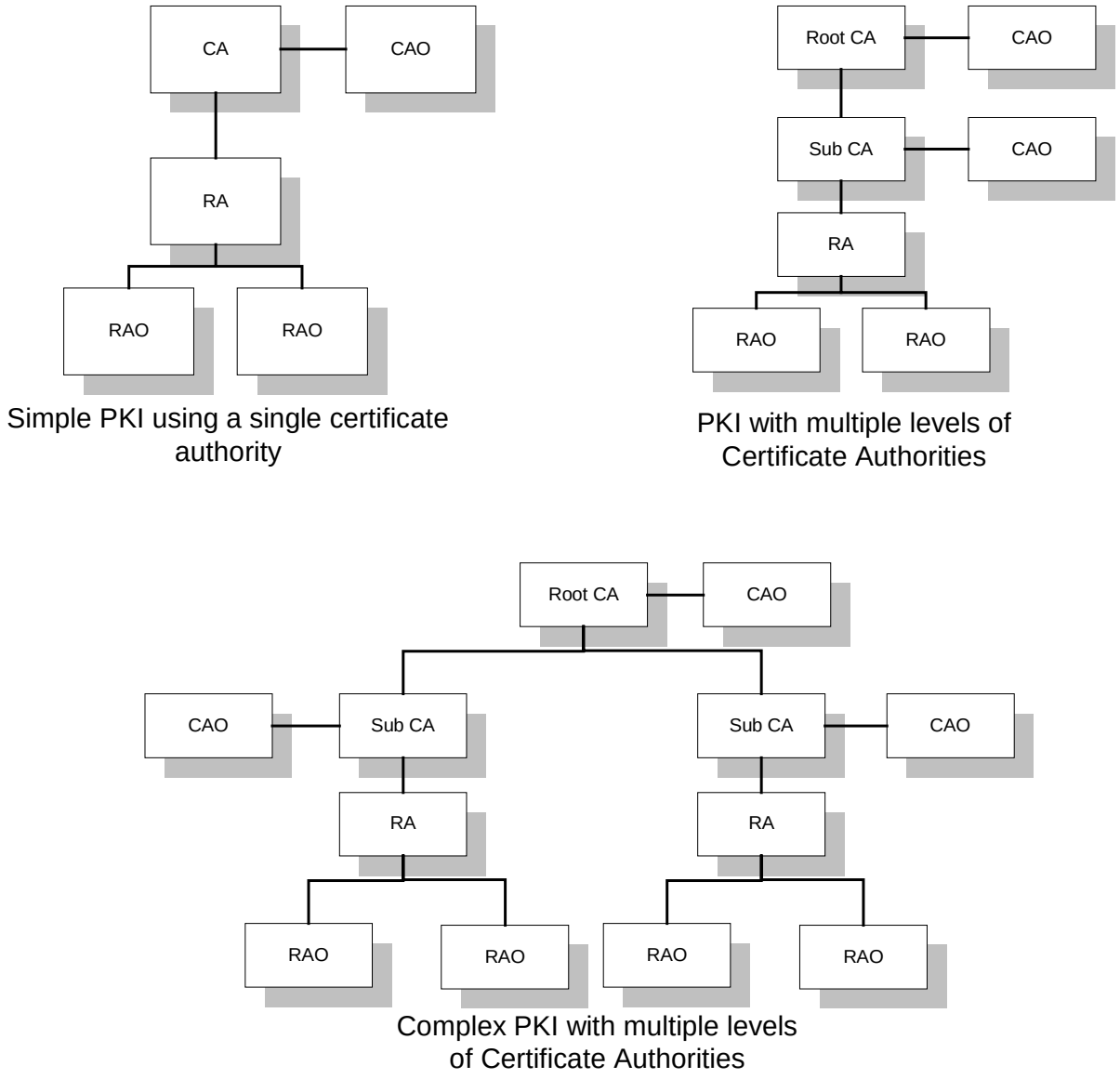

Figure 1 - Example Configurations

Figure 1 shows a number of possible architectural configurations for using UniCERT. UniCERT can be used to implement a variety of PKI configurations. The simplest PKI that can be implemented with UniCERT consists of a CA, CAO, RA and a number of RAOs. From this base configuration a number of more complex configurations can be developed including the ability to add multiple levels of CA under a single Root CA. The information flows between the components is shown more fully in Figure 2 - UniCERT Concept of operations.

2.2.1 Components

The TOE is made up of a number of distinct components that inter-operate in order to provide the functionality of a PKI. This section describes the major components UniCERT utilises in order to offer PKI services. It is noted that whilst each of the components is functionally separate, the components inter-operate in such a manner as to preclude them from being considered separate entities in their own right: i.e. they cannot operate independently.

2.2.1.1 UniCERT CA

The CA is the highest hierarchical component within a PKI system, UniCERT's CA is able to operate both in the role of a 'Root CA' where the CA is the topmost element of a certificate hierarchy, or as a subordinate CA within the hierarchy. The CA consists of two elements: the CA service and the CA Configuration tool. The CA Configuration tool is used initially to configure the CA and the CAO and to generate their keys. The CA service can then run as an NT service. The primary function of the CA is to sign and issue certificates. Each CA requires its own CAO and at least one RA.

The CA performs the following functions:

- generating its own public key pair(s),
- receiving certificate and revocation requests from RAs,
- receiving certificate requests from CAOs for certificates for PKI entities,
- signing certificates using the CA's certificate signing private key[1],
- revoking, suspending and unsuspending certificates,
- where appropriate, checking the revocation passphrase included within the revocation request,
- generating and signing CRLs and ARLs, using the CA's CRL signing private key,
- optionally posting certificates, CRLs and ARLs into an X.500 directory,
- archiving all certificates, CRLs and ARLs created by itself to an Oracle database and also optionally to disk,
- logging operational errors and events to an Oracle database (entries in the log are signed using the CA's non-repudiation private key)
- signing Customer registration Policies (CRPs) using the CA's private key
- pushing Customer Registration Policies to RAs.

2.2.1.2 UniCERT CAO

The CAO is the management interface to the UniCERT CA and the other components within the PKI. Through the CAO the various elements that make up the PKI can be defined. Additionally the CAO is used to enter certificate details for the components of the PKI as well as setting up and distributing Customer Registration Policies for the RAOs.

The CAO contributes to the PKI by:

---

[1] The CA may have one or more private keys which are used for a variety of different functions. For the ITSEC evaluated version of UniCERT the following functions are used: certificate signing, CRL signing, digital signature (for PKIX messages), non-repudiation (for the audit log and archiving). Up to four different keys could be used to perform these functions.

- Providing a tool to allow a CAO User to create and manage a PKI,
- generating key pairs and obtaining certificates for components within the PKI at or below the level of the CAO,
- allowing configuration of CAs, RAs, CAOs and RAOs,
- allowing definition and distribution of the Customer Registration Policies to be used by RAOs,
- revoking, suspending and unsuspending certificates,
- logging operational errors and events to an Oracle database (entries in the log are signed using the CAO's private key).

2.2.1.3 UniCERT RA

The UniCERT RA provides a facility for forwarding certification and revocation requests from the RAO(s) to the CA. The RA is also able to forward certification and revocation requests directly from a Gateway to a CA. The RA also provides a facility for forwarding replies from the CA to the Gateway which in turn sends the reply to an end user. The RA is also responsible for tracking all events in the RA database, which can be accessed by the RAO as required.

The RA performs the following functions:

- receiving requests from either a Gateway or an RAO,
- sending authenticated requests to the CA,
- returning the certificate to the requestor,
- logging operational errors and events to the Oracle database (entries in the log are signed using the RA's private key).

2.2.1.4 UniCERT RAO

The RAO provides the interface through which requests for certification and revocation are received and processed. Requests can be received via e-mail, WWW (remote requests through a Gateway) or in person (face to face). The RAO user must process these requests and perform whatever procedural steps are necessary to ensure that the end user's credentials are valid.

The following functions can be performed by an RAO user from the RAO:

- approving, modifying or rejecting remote certification requests which have been forwarded from a Gateway, based on the Customer Registration Policy,
- providing facilities for face-to-face registration based on the Customer Registration Policy,
- requesting revocation or suspension / unsuspension of a certificate,
- optionally (according to the CRP) generating end user private keys when performing face-to-face registration,

- optionally (according to the CRP) defining a revocation passphrase during the registration process. By defining such a passphrase, it prevents an attacker from revoking someone else's certificate without their permission.

- logging operational errors and events to an Oracle database (entries in the log are signed using the RAO user's private key)

2.2.1.5 UniCERT Gateway

UniCERT also allows users to receive certificates via a Gateway either by e-mail or web.

The Gateway will verify the format of the remote request, in addition checking the signature on it in the case of a signed PKCS#10 request. It then passes the request to the RA which will write it to the Oracle database for further processing by the RAO.

2.2.1.6 UniCERT Token Manager

The UniCERT Token Manager provides the means by which an operator can change the passphrase which protects the component's private key (or keys). The passphrase can be changed only upon correct entry of the current passphrase. The Token Manager may also be used to split .pse files.

2.3 The Security Objectives

The Security Objectives of UniCERT are described below. Each Security Objective is defined as "SOx" where "x" represents the number of the objective.

	Name	Description
SO 1	Provide trusted certificates	The TOE will provide trusted certificates to protect public keys of end users and UniCERT components. This provides a binding of the public key to the user ID and is a function of a Certificate Authority.
SO 2	Provide authorised certificates	The TOE will ensure that all certification and revocation requests are made through the authorised channels and according to an authorised Customer Registration Policy. This provides a binding of the actual user and holder of the private key to the User ID and is a function of a Registration Authority.
SO 3	Keep private keys confidential	The TOE will ensure the confidentiality of private keys for use within the TOE and, optionally, for end users.
SO 4	Create end user private keys	The TOE will provide the facility to generate private authentication and / or confidentiality keys for end users.
SO 5	Generate audit log events and archive	The TOE will generate an audit log of security related events and an archive of certificates, CRLs, ARLs and requests.

2.4 Intended Method of Use

The TOE is designed to fulfil the requirements of a PKI by providing tools to allow the technical aspects of a PKI to be performed. A high level overview of how the TOE operates is provided as Figure 2 - UniCERT Concept of operations.

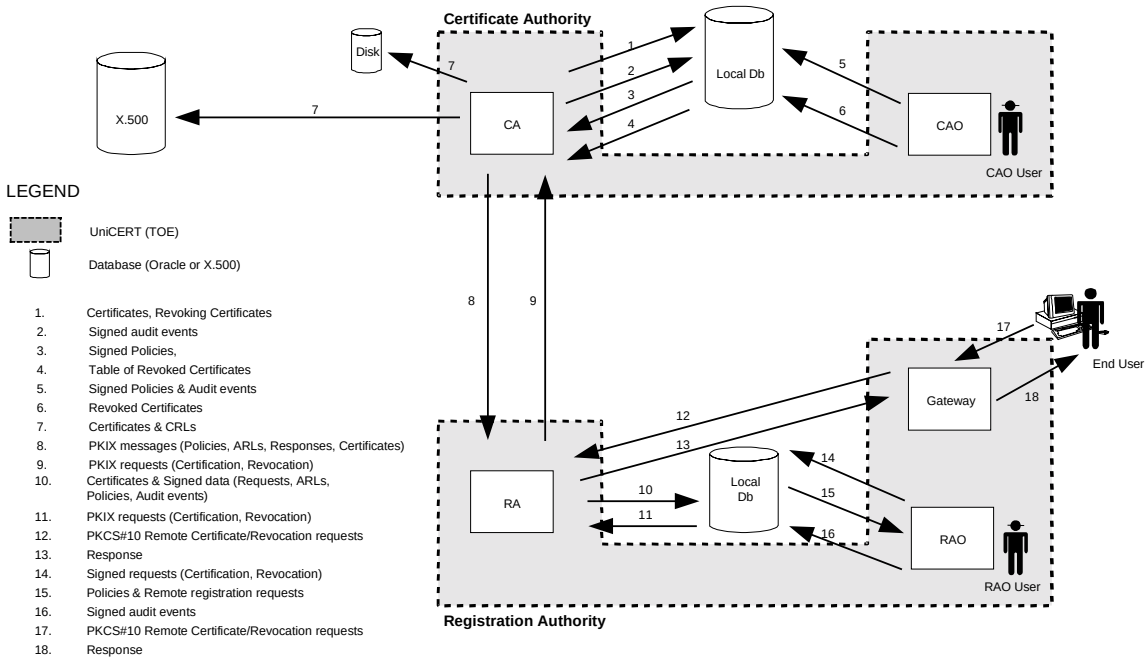

Figure 2 - UniCERT Concept of operations

It should be noted that the TOE components are not intended to inter-operate with other Certificate Authorities or Registration Authorities. For example, a UniCERT CAO could not be used to administer non-UniCERT RAs. UniCERT does support cross certification however, this is beyond the scope of the product for evaluation.

The TOE may have the following two methods of use:

- Certificate Authority (MOU1), and
- Registration Authority (MOU2).

2.4.1 Certificate Authority

One of the primary uses of the TOE is to provide services that allow an organisation to operate a Certificate Authority. In the context of a PKI, a Certificate Authority refers to a trusted organisation that issues *trusted certificates*, where *trusted certificates* can be considered as a binding between a public key and a user ID.

An Oracle Database is required to support the operation of the TOE as a Certificate Authority. This database will need to be accessed by all components which make up the Certificate Authority. The same database may also be used to support the operation of the TOE as a Registration Authority subject to the Assumptions detailed in Section 2.4.3.

The TOE provides features to:

- create and manage the PKI structure including both the Certification Authority and the Registration Authority functions,
- generate, sign and publish public key certificates,
- revoke, suspend and unsuspend certificates as required (verifying revocation passphrases where appropriate),
- maintain and promulgate a CRL and ARL of certificates revoked or suspended by the Certificate Authority,
- verify certificate requests generated by subordinate Certificate Authorities or Registration Authorities,
- identify and authenticate subordinate Certificate Authorities, Registration Authorities and Certificate Authority Operators,
- archive certificates, CRLs, and ARLs,
- log operational errors and events.

When the TOE operates as a Certificate Authority (MOU1), several environmental and procedural aspects also need to be considered. These assumptions are detailed in Section 2.4.3.

The components of the TOE required for this method of use are the CA and the CAO. Note that this may include more than one CAO.

2.4.1.1 Required Roles

A CAO user will run the Certificate Authority. The CAO user is responsible for running the CAO component. The ability to start up the CAO is controlled through access to all of the components which make up the .pse file. Each component requires the correct entry of a passphrase. The CAO user will also be responsible for starting up the CA. Again, more than one passphrase is required to gain access to the CA functions[2]. A CAO user can be configured to have access to a selection of different functions, including:

- creating the PKI with its associated keys and certificates,
- revoking certificates for components within the PKI,
- revoking end user certificates,
- creating, deleting and modifying policies,
- browsing internal audit logs and the NT event log for events generated by the CA and CAOs,
- allow distribution of Customer Registration Policies.

[2] One or more different people may be required to gain access to the CAO component. Depending on the environment, one person may have responsibility for all of the components of the .pse file, or a different person may be responsible for each component. The same is true for the CA component.

Any person who is authorised to know one or more of the passphrases required to access the CAO or CA components is assumed to be a CAO user and therefore considered to be trusted.

2.4.2 Registration Authority

Within a PKI, the Registration Authority is the entity to which an end user would go to in order to receive an authorised certificate. A Registration Authority does not have responsibility for developing the policy under which the certificate is issued, rather the Registration Authority is responsible for ensuring that the policies of its Certificate Authority are implemented and that any certificates issued are in accordance with that policy. The Registration Authority is responsible for proving that the individual granted the certificate is who he or she claims to be.

An Oracle Database is required to support the operation of the TOE as a Registration Authority. This database will need to be accessed by all components which make up the Registration Authority. The same database may also be used to support the operation of the TOE as a Certificate Authority subject to the Assumptions detailed in Section 2.4.3.

The TOE provides the following features for use in operating a Registration Authority:

- signing certificate requests to be sent to a Certificate Authority where the certificate request is either received by face-to-face contact or remotely through a Gateway,
- receiving certificate revocation requests, including requests for suspension and unsuspension, and then sending the signed requests to the Certificate Authority,
- logging user requests for certificate generation and revocation,
- receiving user certificates from a Certificate Authority,
- delivering users' certificates,
- ensuring (where possible) that the requestor of a certificate is in possession of the private key associated with the public key certificate,
- enforcing conformance with the certificate registration requirements of the Certificate Authority,
- optionally generating end user private keys and delivering the key to the end user via a diskette,
- logging operational errors and events,
- dynamically create and update tables in the oracle database upon receipt of a valid Customer Registration Policy push message from a CA.

When the TOE operates as a Registration Authority (MOU2), several environmental and procedural aspects also need to be considered. These assumptions are detailed in Section 2.4.3.

The components of the TOE required for this method of use are the RA, RAO and Gateway. Typically a single RA component will be connected to one or more RAO components and Gateways, and each RAO component will be connected to only one RA component.

2.4.2.1 Required Roles

The RAO user is responsible for processing end user certification requests through running the RAO component. The ability to start up the RAO is controlled through access to all of the components, which make up the .pse file. Each component requires the correct entry of a passphrase. [3]

The RAO user has many different functions, for example registering users and requesting certification. Also, multiple authorisation may be stipulated for certification and revocation requests. If this is the case, more than one RAO user will need to be established and sign the requests. The multiple authorisation functionality is excluded from the scope of this evaluation.

Other functions that may be assigned to RAO users include:

- generating revocation requests including those for suspension and unsuspension
- browsing internal audit logs and the NT event log for events generated by the RA and RAOs
- processing face-to-face registration requests from end users.

Any person who is authorised to know the RAO passphrase is assumed to be an RAO user and therefore considered to be trusted.

An RA administrator is also required for running up and managing the RA component. This user may or may not also be responsible for registering users, depending on the environment in which the TOE is being run. Again more than one passphrase may be required to gain access to the RA functions depending on the number of components which make up the .pse file.

2.4.3 Environmental and Method of Use Assumptions

This section details what assumptions are made about the operation of UniCERT. It is the responsibility of the customer to comply with these assumptions. The relevant information will be supplied to the customer before installation of the TOE. Note that the security of the product depends upon its proper use by trusted CAOs and RAOs. It is not designed to counter attacks made by these users.

Assumption 1	Trusted system and database administrators are available to install and maintain the TOE in accordance with the configuration described in Appendix A.
Assumption 2	The people responsible for running the CAO and RAO components can be trusted, within the bounds of the functionality provided by these components.
Assumption 3	Procedural measures are in place to verify the identity of an end-user.
Assumption 4	Physical access to the CA and CAO is restricted to CAO users and system / database administration staff.

[3] One or more different people may be required to gain access to the RAO component. Depending on the environment, one person may have responsibility for all of the components of the .pse file, or a different person may be responsible for each component. The same is true for the RA component.

Assumption 5 Physical access to the RA is restricted to RAO Users, RA administrator and system / database administration staff.

Assumption 6 Logical access to the systems on which TOE components and Database(s) are installed are restricted in accordance with a defined security policy which must define suitable countermeasures to mitigate any risk factors to an acceptable level.

Assumption 7 Network links between TOE Components and Database(s) are protected from eavesdropping in accordance with a defined security policy that defines suitable countermeasures to mitigate any risk factors to an acceptable level.

Assumption 8 Operation of the TOE is performed according to organisational polices that specify sound information management procedures including disaster recovery of TOE components and data.

Assumption 9 Procedural and physical measures will be established to ensure that the back-up media is held securely, is accessible only to authorised administrators, and when no longer required is deleted or destroyed in a secure manner.

Assumption 10 Operators will keep their passphrases secret and change them at regular intervals.

Assumption 11 Procedural measures are in place to ensure that CRLs and ARLs are made available in a timely manner, to end users and applications that are using certificates issued by CAs in the PKI hierarchy. Users are responsible for the correct use of such CRLs and ARLs, in accordance with policies and procedures issued by the operators of the TOE e.g. Certificate Practice Statement.

2.5 Threats

The assumed threats that the TOE is designed to counter are listed below. The correlation between threats and SEFs will be described in the Suitability Analysis associated with this product [UN-SA]. Each threat is defined as “Tx” where “x” represents the number of the threat.

	Name	Description
T 1	Creation of untrusted certificate	An attacker may attempt to create an untrusted certificate and masquerade as a holder of a trusted certificate.
T 2	Unauthorised request for certificate	An attacker may attempt to masquerade as a valid requestor of a certificate by impersonating an authorised RAO, RA or CAO.
T 3	Unauthorised modification of CRP	An attacker may attempt to amend the Customer Registration Policy which is enforced by an RAO for registering end users.
T 4	Unauthorised access to private keys	An attacker may attempt to obtain access to the private key of a component or an end user.

	Name	Description
T 5	Unauthorised modification of certificate	An attacker may attempt to modify a trusted certificate.
T 6	Unauthorised revocation	An attacker may attempt to revoke a valid certificate without authorisation.
T 7	Undetected modification / deletion of CRL or ARL	An attacker may attempt to modify or delete a CRL or ARL to prevent revocation of invalidated certificates.
T 8	Undetected modification of audit log entry	An attacker may attempt to modify entries within the audit log.
T 9	Carry out undetected attack on the system	An attacker may attempt to mount a direct attack on the system without detection.
T 10	Generate bad keys	The keys generated by the product may be invalid or may be mere repetitions of previous keys.

2.6 Evaluated Configuration

The version of the TOE to be evaluated is UniCERT version 3.1.2.A. The TOE is to be configured according to the guidelines contained in Appendix A – Evaluated Configuration(s).

In the course of its operation, the TOE relies upon aspects of the Windows NT Operating System (OS) and Oracle Database Management System (DBMS) to operate in a reliable fashion. In order to ensure that the TOE operates as it has during the evaluation, it is important that the same versions of these products are used in any evaluated configuration. This means Windows NT4, Service Pack 5 and Oracle 8.0.5 must be used.

The TOE relies upon Windows NT functionality in order to support the correct operation of the TOE and also to ensure that there is process separation between different processes running on the same machine. This feature protects one process's memory-resident sensitive information from other processes.

3 Security Enforcing Functions

This section describes the SEFs provided by the TOE in an informal style as required by [ITSEC] for a target evaluation level of E3. Each SEF is defined as "SEFx" where "x" represents the number of the SEF. The first paragraph in the description column is italicised and is a statement of the SEF, and the subsequent paragraphs are the description.

	Name	Description
SEF 1	Sign certificates	*The TOE shall create and sign X.509 certificates of end users and UniCERT components.* On receipt of a valid request for a certificate to be created (see SEF2), the CA shall generate a certificate containing the public key of the end user or UniCERT component. The CA shall then sign this certificate using the certificate signing private key of the CA.
SEF 2	Sign / verify certification requests	*The TOE shall provide the functionality to sign and verify certification requests.* All requests for a certificate shall be signed by an authorised component within the TOE and verified by the component which acts upon this request. Requests for certificates for components within the TOE are generated and signed by the CAO using its own private key, and verified by the CA before the certificate is generated.[4] Requests for certificates for end users are generated by the RAO, and sent to the CA via the RA. The RAO signature on the request, generated using the RAO's private key, is verified by the RA. The RA signature on the request, generated using the RA's private key is verified by the CA. The CA will use its list of valid RA certificates from the Oracle database to verify the RA's signature, checking that the certificate was generated by itself. It will use a similar process for verifying the CAO's signature. The RA will verify the RAO signature using a certificate from the list of RAO certificates in the Oracle database. In the process it will check that the certificate was generated by the CA whose certificate is contained in its .pse file and that the certificate does not appear in the

[4] This is true for all components other than the root CA and its CAO and, optionally sub CAs and corresponding CAOs. These certificates are generated by the CA Configuration tool on initial configuration of the PKI.

	Name	Description
		current ARL which it has stored in its database. Requests for certificates which are received remotely from end users by e-mail or web (PKCS#10) are signed using the private key of that end user. The Gateway will check that the signature can be verified using the public key within the certification request. If the end user supplies a PKCS#10 to the RAO user (face to face) then the RAO will check that the signature can be verified using the public key within the certification request.
SEF 3	Sign CRL and ARL	*The TOE shall create and sign a CRL and ARL using all the certificates listed in the certificate revocation table contained in the Oracle database.* An X.509 v2 CRL and ARL shall be generated by the CA in one of the following ways: i) on a periodic basis based on the specified, configurable time period in the CA Operational Policy ii) the CAO can manually generate a new CRL and ARL at a time other than the configured time iii) when a certificate is revoked a CRL may be generated in addition to the CRL and ARL generated by i), if the TOE has been configured to do so. The CRL shall contain a list of all end user certificates held in the certificate revocation table, the date and time that the CRL is generated, and the date and time of the next specified CRL generation. The ARL shall contain the same but for PKI component certificates. The CRL and ARL are signed using the CRL signing private key of the CA. The next generation date is included within the CRL and ARL to allow users to determine if it is the most recently issued one and to alert them to a potentially missing CRL or ARL. They also include a sequence number for the same reason. On creation, the CRL and ARL are posted to the X.500 directory, if required and written to the local disk. The ARL is also sent down to the RA within a PKIX message.
SEF 4	Sign / verify revocation request	*The TOE shall provide functionality to sign and verify certificate revocation or certificate suspension/unsuspension requests.* All requests for certificate revocation shall be signed by an authorised component within the TOE and verified by the component which

	Name	Description
		acts upon this request. Either the CA or the CAO is able to revoke a certificate by adding that certificate to the certificates revocation table within the Oracle database. Component certificates will be revoked directly by the CAO. End users certificates can be revoked by the CA either upon request from an RAO via the RA, or upon request from a web browser via the RA. Alternatively they can be revoked by the CAO directly. Note that certificates cannot be suspended or unsuspended remotely. A revocation request from an RAO will be signed by the RAO using its own private key, and verified by the RA. A revocation request from the RA will be signed by the RA using its own private key and verified by the CA. A revocation request from a web browser will not be signed, but must be protected using a revocation passphrase as described below. The CA will use its list of valid RA certificates from the Oracle database to verify the RA's signature, checking that the certificate was generated by itself. It will use a similar process for verifying the CAO's signature. The RA will verify the RAO signature using a certificate from the list of RAO certificates in the Oracle database. In the process it will check that the certificate was generated by the CA whose certificate is contained in its .pse file and that the certificate does not appear in the current ARL which it has stored in its database. There is an option to mandate the use of a revocation passphrase within the CRP. If it has been mandated, a hash of the passphrase will be sent to the CA for storage during the initial generation of the certificate. Then, during revocation, the full passphrase will be included in the revocation request to the RA. The RA will then hash the passphrase and include it in the revocation request to the CA, and this will be verified by the CA against the stored value. The CA will only revoke the certificate if the hashed passphrase matches up.
SEF 5	Sign / verify Customer Registration Policy	*The TOE shall provide the functionality to sign and verify CRPs.* Registration of users will be carried out by the RAO in accordance with a CRP defined by the CAO. The CRP is signed by the CA and transmitted to the RAO via the RA. The CAO defines the CRPs under which RAO users will be entitled to register users. The RA, on receipt of the signed CRP information, will verify the signature and store the policy and RAO details in its database. The RAO user will then only be able to access the appropriate policies for registering users that are available to it from

	Name	Description
		the database, and will verify the signature on each available CRP.
SEF 6	Access control	*The TOE shall provide the functionality to control access to the private keys within the TOE.* Access to any of the private keys shall require the entry of the correct corresponding passphrase. Access to the private key is required prior to accessing the functionality of each component within the PKI. Every private key within the system is stored within a file in software (known as the .pse file). The .pse file is split into one or more components, all of which are required to gain access to the full file. The .pse file (or .pse component) is encrypted under a key derived from a passphrase using the Security Mechanism SM8 defined below. The integrity of the .pse file is also protected using the security mechanism SM5 defined below, which is used to hash the .pse file contents and append the hash to the .pse file before it is encrypted. The passphrase is defined by the CAO on generation of the private key. The passphrase shall consist of a minimum of 8 characters and will contain at least one of each of the following: 1 lower case character, 1 upper case character, 1 numeric character, 1 other character (i.e. not a letter, not numeric). This is policed by the product. The passphrase is entered into the TOE via the user interface, and kept in memory only whilst it is being used, after which it is destroyed. The passphrase is not stored in the TOE. This passphrase can then be changed using the Token Manager. It requires entry of both the current passphrase and the new one before the passphrase will be changed. An end user private key, which is generated at the RAO, is also stored in a file (to be written to the hard disk, or a diskette) encrypted under a passphrase. This file will then be given to the user. The choice of passphrase shall be as described above. When the private key is in active use by the software, it will be held obfuscated within memory, therefore any RAM dump will not easily reveal the content of the private key. Also, memory containing the obfuscated private key will be actively overwritten when no longer required.
SEF 7	Generate key pairs	*The TOE shall provide the functionality to generate key pairs for components within the PKI and optionally for end users.* The TOE allows the creation of RSA or DSA key pairs with a

	Name	Description
		maximum length of 2048 and 1024 bits respectively. The CA Configuration tool shall initially be used to generate the key pairs for the root CA and the CAO. Once this has been done, the CAO will then be able to create new PKI components and will generate key pairs for them. (If a sub-CA is created, its key pairs can be created either by the CAO which created it, or by itself using its CA Configuration tool.) Optionally, the RAO will create key pairs for an end user as part of a face-to-face registration process.
SEF 8	Generate archive and audit log	*The TOE shall provide the functionality to log security related events and to archive certificates, CRLs, ARLs and certification/revocation requests.* The TOE generates an audit log listing security relevant events, which may then be used by the CA or RA Administrator to identify attempts to attack the system. The security relevant events that are logged are defined in [UN-DD]. The CA writes details of each certificate it creates to a table within the Oracle database. Included within the record is the date and time when the certificate was created, the serial number of the certificate and the X.509 certificate itself. The CA also writes details of each CRL and ARL it creates to a table within the database. Included within the record is the date and time it was created, the serial number of the CRL / ARL and the CRL / ARL data itself. The RA maintains details of each certification and revocation request that it processes in a table within the Oracle database. Included in the table is the date and time of the event, details of the entity which created the request, and the request itself. The TOE will also write certain events to the NT event log in such cases where access to the Oracle database has not been established, and security-relevant events need to be logged. Events which are written to the NT event log include errors in entering the passphrase for accessing the .pse file of components and errors in changing the passphrase.
SEF 9	Sign audit log entries	*The TOE shall provide the functionality to individually sign each audit event that is written to the UniCERT audit log..* Each event is signed by the entity which writes the event to the database. This may then be verified by the CAO or RAO (depending on whether it is written to the CA or the RA database).

3.1 Correlation of Security Objectives to Threats

A summary of the correlation of Security Objectives to threats is given in the table below.

		SO1	**SO2**	**SO3**	**SO4**	**SO5**
		Provide trusted certificates	Provide authorised certificates	Keep private keys confidential	Create end user private keys	Generate audit log events and archive
T1	Creation of unauthorised certificate					
T2	Unauthorised request for certificate					
T3	Unauthorised modification of CRP					
T4	Unauthorised access to private keys					
T5	Unauthorised modification of certificate					
T6	Unauthorised revocation					
T7	Undetected modification / deletion of CRL or ARL					
T8	Undetected modification of audit log entry					
T9	Undetected attack on the system					
T10	Generate bad keys					

3.2 Correlation of SEFs to Threats

		SEF1	**SEF2**	**SEF3**	**SEF4**	**SEF5**	**SEF6**	**SEF7**	**SEF8**	**SEF9**
		sign certs	sign / verify cert requests	sign CRL and ARL	sign / verify revocation request	sign / verify CRP	access control	generate key pairs	generate audit and archive log	sign audit log entries
T1	Creation of unauthorised certificate									
T2	Unauthorised request for certificate									
T3	Unauthorised modification of CRP									
T4	Unauthorised access to private keys									
T5	Unauthorised modification of certificate									
T6	Unauthorised revocation									
T7	Undetected modification / deletion of CRL or ARL									
T8	Undetected modification of audit log entry									
T9	Undetected attack on the system									
T10	Generate bad keys									

4 Security Mechanisms and Evaluation Level

4.1 Security Mechanisms

The following mechanisms are required to support the SEFs within the TOE. Each mechanism is defined as "SM x" where "x" represents the number of the mechanism. The SEF which is specifically supported by that mechanism is shown within the following table.

	Name	Description
SM 1	Generate certificate	Certificates will be generated using an X.509 certificate structure. See [X.509v3]. This mechanism supports SEF1.
SM 2	Protect messages	Messages between the different components are protected using the IPKI format defined in [PKIX]. This mechanism supports SEF2 and SEF4.
SM 3	Certificate Request	End user certificate requests which are sent remotely over e-mail or that are supplied to an RAO user when doing face to face registration, are protected with a signature using the PKCS#10 format defined in [PKCS#10]. This same mechanism is used to protect certification requests sent from a sub CA to a root CA. This mechanism supports SEF2.
SM 4	Signature algorithm	The signature algorithm can be either RSA or DSA, as defined in [RSA] and [DSA] respectively. RSA key lengths can be 768, 1024 or 2048 bits. DSA key lengths can be 768 or 1024 bits[5]. The choice is made for each individual component during the setting up of the PKI hierarchy. This mechanism supports SEF1, SEF2, SEF3, SEF4, SEF5 and SEF9.
SM 5	Hash Algorithm	The hash algorithm can be either SHA-1 or MD5 defined in [DSA] and [MD5] respectively. The choice is made for each individual component during the setting up of the PKI hierarchy. This mechanism supports SEF1, SEF2, SEF3, SEF4, SEF5, SEF6 and SEF9.

[5] In fact it is possible within UniCERT to create DSA keys of 2048 bits, although this contravenes the Digital Signature Standard [DSA].

	Name	Description
SM 6	Date and time of current and new release of CRL and ARL	The CRL and ARL will be generated using CRL v2 according to the standard defined in [X.509v3]. This mechanism supports SEF3.
SM 7	Derive key and IV from passphrase	The user's passphrase is used to derive a symmetric key and IV using the method described in [PKCS#12]. This mechanism supports SEF6.
SM 8	DES encryption	The private key file is encrypted using triple-DES (as defined in [DES]) using the key and IV derived from the passphrase, as described in the previous mechanism. This mechanism supports SEF6.
SM 9	Asymmetric key generation	RSA keys are generated to be compatible with [RSA_KeyGen]. DSA keys are generated in accordance with [DSA] but a different random number generator is used [RNG]. This mechanism supports SEF7.

4.2 Strength of Mechanisms

The minimum strength of the mechanisms used within the TOE will be medium.

4.3 Evaluation Level

The target evaluation level for the TOE is E3 [ITSEC].

Appendix A Evaluated Configuration

The evaluation of the TOE will be according to the following configuration:

1. General Requirements

The following specifications refer to each of the different configurations to be evaluated.

1.1 Version of UniCERT Software

The version of UniCERT to be evaluated will be 3.1.2.A.

There are two different releases of UniCERT. These are the US version and the Rest of the World versions. The difference is only with the underlying cryptographic module. Only the Rest of the World release is to be evaluated. UniCERT can be localised for specific language support, but only the English version is being evaluated.

An additional component normally available for UniCERT is the Archive Server. This will *not* form part of the ITSEC evaluation.

UniCERT can be run as a software only system or with hardware devices for performing the cryptographic functionality. All cryptographic functionality for the evaluation will be performed in software.

1.2 Operating System

The different components within UniCERT (Certification Authority (CA), Registration Authority (RA), Certification Authority Operator (CAO), Registration Authority Operator (RAO), Gateway and Token Manager) all run on Microsoft Windows NT4 (with Service Pack 5).

1.3 Hardware Requirements

The minimum requirements for each UniCERT component when running separately is a Pentium 266 MHz machine with at least 128 MB of RAM, with the initial paging file size set to be at least 125 MB in length. In addition, any system disk must have at least 50 MB of free space. These machines, and the database server machine(s), must be dedicated to running UniCERT components and Oracle respectively, and must not be used for running other applications.

1.4 Third Party Software Requirements

Oracle: The TOE will be running with version 8.0.5 of the Oracle Database.

LDAP Server: The X.500 server to which the CA posts information will not be part of the evaluation. However, in order that the process of posting the information should work, an LDAP server will be included within the configuration, on a separate machine, with only the LDAP interface connecting it to the CA.

1.5 Configuration of the CA Component

The following configuration options at the CA will be specified for the evaluation:

- Manual startup only
- Key Length: RSA 2048, 1024, DSA 1024
- HSM = none
- Not DAP enabled
- No OCSP support
- No unique DN or public key checking

1.6 Configuration of the CAO Component

The following configuration options at the CAO will be specified for the evaluation:

- HSM = none,
- Key Lengths: RSA 2048, 1024, DSA 1024

When the CAO is configuring other PKI entities then the following configuration options must be specified for the evaluation:

- For all PKI components – HSM = none,
- For all PKI components, Key lengths: RSA 2048, 1024, DSA 1024
- No Archive Server components within the PKI
- No Cross certification will be handled within the PKI
- No RAO components for bulk certification will be included within the PKI (ARM).
- No PKI-Plus components within the PKI
- When configuring RAs the following configuration options should be selected
  - "RA automatically accepts all certification requests" is disabled
  - "RA will accept ARLs" option selected
  - "RAOs can not use RC2" option selected
  - "RAOs can use unsigned policies" option disabled
  - The certificate rollover options are not enabled

The following restrictions for defining Customer Registration Policies will be made:

- Key sizes for RSA and DSA will not include 512 bits.
- ECDSA will be excluded.
- The "Archive Secret Key" option will not be included in any policy
- End user "Certificate Rollover" option will not be included in any policy
- Cisco SCEP option will not be included in any policy
- "Required Number of RAOs" option will not be included in any policy i.e. multiple authorisation is not supported

1.7 Configuration of the RA Component

The following configuration options for the RA will be specified for the evaluation:

- HSM = none
- Key Lengths: RSA 2048, 1024, DSA 1024

1.8 Configuration of the RAO Component

The following configuration options for the RA will be specified for the evaluation:

- HSM = none
- Key Lengths: RSA 2048, 1024, DSA 1024.

1.9 Configuration of the Gateway Component

The following configuration options for the Gateway will be specified for the evaluation:

- encrypted requests will be disabled.

2. Components for the Certificate Authority

2.1 Basic Configuration

The basic configuration for the Certificate Authority is as depicted in Figure 3. This comprises three UniCERT components: the CA, the CAO and the Token Manager. The CA and CAO can be installed on separate machines or on the same machine, but if they are on separate machines, a Token Manager needs to be installed for each one. An Oracle database will be connected to both CA and CAO. An X.500 directory (running on a separate machine) will be connected using LDAP, but this does not form part of the evaluation.

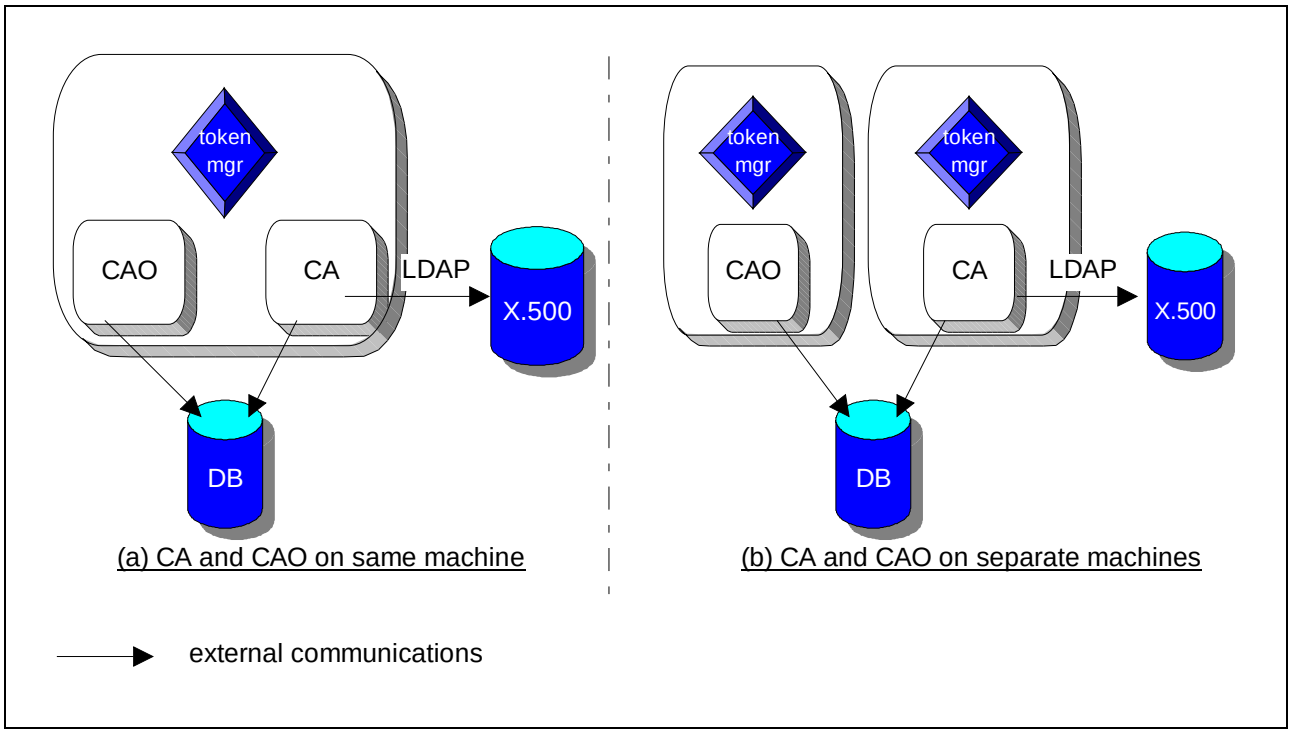

Figure 3: Basic CA Configuration

The reasons why the CA and CAO may be on separate systems are that the CA runs as a service, and is intended for unattended running, whereas the CAO is an operator based application.

In either case both the CA and the CAO have connections to the database which may be external to the system. When the CAO is on a separate system to the CA, there are no additional security risks (beyond those covered by the assumptions), as the messages passing between the CA and CAO are signed PKIX messages.

Note that for the evaluated configuration, no other software will be installed on the TOE platform other than that specified here.

2.2 Multiple CAOs

The Certificate Authority can be configured with a single CA component and multiple CAOs. Multiple CAOs may be required for operational purposes e.g. CAOs in different locations or working on different shifts. These CAOs interact only with the CA and not with each other, and do not provide any additional functionality over the interaction between a single CAO and CA.

2.3 Number of Certificate Authority Hierarchies

The certificate authority is designed to operate either on its own or as part of a hierarchy. It is impractical to evaluate all the different hierarchies which are possible, therefore we will define the interaction between the CAs in the hierarchy, evaluate this and evaluate a single Certificate Authority. This should provide sufficient confidence in the working of UniCERT with any number of levels in the hierarchy, and any number of Certificate Authorities in a single level.

The only interaction between CAs in a hierarchy, is for a subordinate CA to send a certification request for its own certificate signing key to its immediately superior CA, and for the superior CA to return the certificate. This process is only done when the subordinate CA is initially configured or when it renews its keys. Within UniCERT this is done in an off-line manner with files containing the certification request and response being passed between the two CAs. All CAs in the hierarchy can certify end users and certify subordinate CA requests.

If operating on its own then it will be designated the Root CA and will interact only with the Registration Authority (or Registration Authorities) underneath it (see Section 2.4.2). If there are other CAs in the hierarchy, they will interact with each other only for the purpose of obtaining certificates as described above.

The certification process of a single subordinate CA with its superior CA, and the interaction between a single CA and Registration Authority are equivalent to a system consisting of multiple Certificate Authorities in a hierarchy.

3. Components for the Registration Authority

3.1 Basic Configuration

The basic configuration for the Registration Authority is as depicted in Figure 4 This comprises four UniCERT components: the RA, the RAO, the Gateway and the Token Manager. The RA, RAO and Gateway can be installed on separate machines or on the same machine, or in other combinations. If the RA and RAO are on separate machines, a Token Manager needs to be installed for each one. An Oracle database will be connected to both RA and RAO. External browsers and e-mail clients can send messages to the Gateway, but they do not form part of the evaluation.

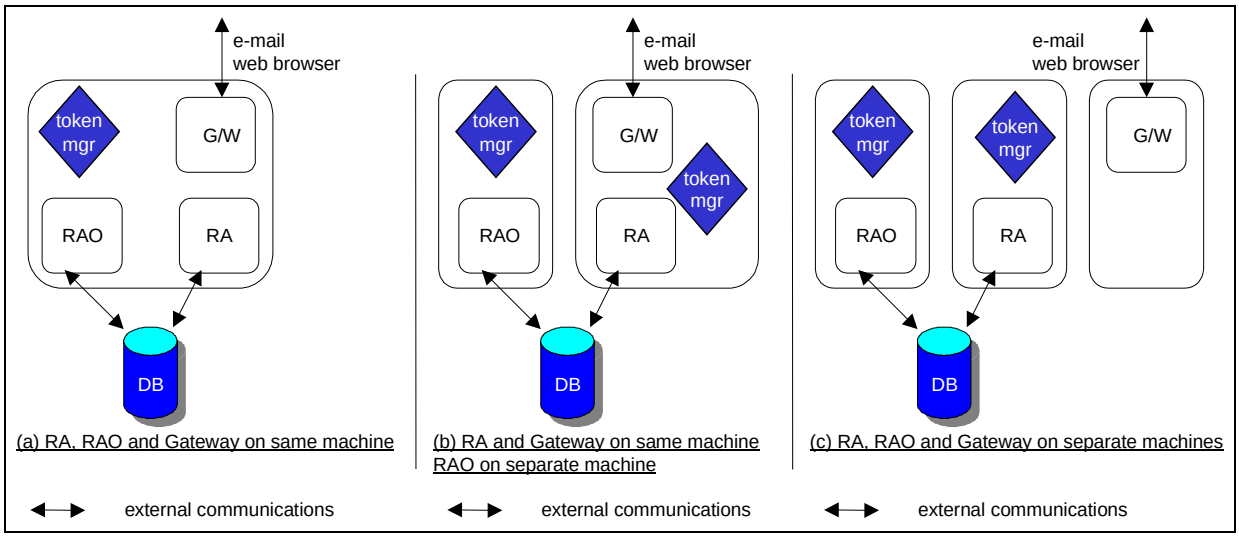

Figure 4: Basic RA Configuration

The reasons why the RA, Gateway and RAO may be on separate systems are that there may be operational requirements specific to an individual installation e.g. the Gateway and the RA do not require continuous operator access, whereas the RAO is an operator based application.

The RA and the RAO do not have any direct connection, all communication is via the database which may be external to the system(s), thus there are no additional security risks if the RA and RAO run on separate systems (beyond those covered by the assumptions). The Gateway only interoperates with the RA using a message based interface, so the only security implication of running on separate systems are covered by the assumptions.

Note that for the evaluated configuration, no other software will be installed on the TOE platform other than that specified here.

3.2 Multiple RAOs

The Registration Authority can be configured with a single RA component and multiple RAOs. Multiple RAOs may be required for operational purposes e.g. RAOs in different locations or working on different shifts. These RAOs interact only with the RA and not with each other, and do not provide any additional functionality over the interaction between a single RAO and RA.