Title of PAPER:
Secure X.500 Border Directory Proxy Server

Abstract:
Following our participation in a Multilevel Secure X.500 Directory Server pilot sponsored by the U.S. Air Force's Rome Laboratory in 1996-1997, Wang Government Services has undertaken further research and development into the creation of a Secure X.500 Border Directory Proxy Server that would enable the sharing of directory information among the U.S. and its allies; or between any other organizations that require limited, strictly controlled, secure sharing of directory information.

To achieve the secure "release" of internal directory information to the outside world, the Secure X.500 Border Directory Proxy Server would provide directory-specific firewall protection mechanisms that would filter and in some cases modify or delete ("sanitize") specific directory information before release to ensure conformance of the released information with the defined releasability policy.

In military/defense, intelligence, diplomatic, and similar communities, in addition to firewall protections, the Secure X.500 Border Directory Proxy Server could also act as a Trusted Guard, with the level of assurance required to maintain the mandatory separation between the "inside" and "outside" domains for everything except the controlled sharing of directory information according to a releasability policy strictly enforced by the Border Directory system.

To provide the highest possible level of assurance, the Secure X.500 Border Directory Proxy Server would run on the Wang Government Services XTS-300™ Trusted Computer System, which has been evaluated at the Class B3 level by the National Computer Security Center (NCSC). The Secure X.500 Border Directory Proxy Server would integrate the existing X.500 filtering capabilities and trusted X.500 routing of Wang's Defense Information Infrastructure (DII) Guard with additional filters that would collectively implement all potentially required directory firewall and Trusted Guard filtering mechanisms.

Authors: K.M. Goertzel
Organization: Wang Government Services, Inc.
Phone Number: 703-698-5605
FAX Number: 703-689-4765
E-mail: km.goertzel@wang.com
Point-of-Contact: K.M. Goertzel

1 Introduction

Electronic directory services are a key component of distributed computing. Every application and user operating in a distributed environment depends either directly or indirectly on some form of directory service. Directories assist in locating information and resources on both an enterprise-wide and a global basis. For this reason, directory technology is being implemented by more and more government and commercial organizations as the foundation for an increasing number of their strategic enterprise information management applications.

The original, and still most widespread, use of an electronic directory is to support the operation of the organization's electronic mail/message handling system and its supporting Public Key Infrastructure (PKI). In addition, Directories can contribute significantly to other business processes, such as user registration, personnel management, and physical security management. In all cases, a directory can provide a single, authoritative source for information such as user names, e-mail addresses, telephone numbers, and security credentials (often several of each per user). A full-service directory can act as a central repository that maintains and provide access to this information.

Directory technology and standards have become increasingly sophisticated over the past few years. Current directory technologies fall into four categories:

- Document-based directories, such as paper telephone books, printed mailing lists, online telephone and mailing lists stored as word processing documents, paper Rolodexes, and "hypercard" rolodexes
- Proprietary databases
- LDAP directories
- X.500 Directories

Two complementary international standards predominate in providing directory services: X.500, the International Telecommunications Union (ITU) standard for directory service; and the Lightweight Directory Access Protocol (LDAP—RFC1777), developed by the Internet Engineering Task Force (IETF), which has become the *de facto* standard protocol for directory client access to directory servers.

Directory products are being developed with increased functionality, such as directory firewalls and meta-directories. Directory firewalls perform application-layer security filtering on directory requests, responses, and errors. Meta-directories can synchronize the information from multiple directories to provide a common, central source of directory information. Both products allow existing investment in directories to be leveraged by enabling the construction of a distributed directory system.

In many cases, the mission and/or enhanced business performance of an organization will require it to share its directory information—or more accurately, a *subset* of that information—with external entities. The communication between an organization's internal directory and other, external directories is often implemented via the Border Directory concept, as defined in ACP 133 (the Allied military standard for directory service). A Border Directory is a directory situated on the boundary between the internal organization's network and directory infrastructure and the external network. The Border Directory is used to make accessible to the external network a subset of the total directory information found in the internal directory. It can provide this access in one of two ways:

1) By acting as a gateway to release internal directory information only when needed, i.e., on demand, in response to external directory requests (LDAP, DAP, and/or DSP).

2) By itself acting as a shared directory/repository (or moving internal directory information to another directory that acts as a shared repository) to which the internal directory information is shadowed (replicated). This shared directory is made accessible to both internal and external users.

Because the internal organization can only define

what information it will share, but cannot hope to dictate how external users handle the shared directory information once it is accessible to them, the shared repository approach increases the risk that a much larger amount of the organization's directory information could be abused or corrupted (either unintentionally or maliciously) by the external users than does the gateway approach. When the Border Directory acts only as a gateway allowing "on-demand" release of internal directory information, the directory information continues to reside and is maintained only in the internal directory, and because it will not be made externally accessible "in bulk", but will only be made accessible on a "per transaction" basis, i.e., a very small amount of information in response to a request, the risk that the organization will lose control of the integrity of the information is vastly reduced.

However, there may be operational situations where performance concerns outweigh fears about loss of information integrity. In such situations the strong protections of a trusted platform for hosting the shared Border Directory are desirable.

While a wide variety of directory products exist that can provide Border Directory functionality, these products do not also provide the necessary security protections to ensure protection from unauthorized access by external users of the organization's internal directory, once communication has been allowed between the internal directory and external Directories.

Following our participation in a Multilevel Secure X.500 Directory Server pilot sponsored by the U.S. Air Force's Rome Laboratory in 1996-1997, Wang Government Services has undertaken to further specify and design a Secure X.500 Border Directory Proxy Server that could be used to enable the sharing of directory information among the U.S. and members of the Combined Communications-Electronics Board (CCEB, which includes the U.S., UK, Canada, Australia, New Zealand), and the NATO alliance, and/or the members of other collaborative alliances or joint military task forces. Moreover, the Secure X.500 Border Directory Proxy Server would be appropriate for use by any organization that desires strict control and assured security when sharing its internal directory information with other entities.

This paper discusses the basic concepts that are germane to understanding the operation of X.500 directory systems, and their security protections in particular. The paper then goes on to describe a secure directory architecture that will allow the strictly controlled sharing of directory information across domain boundaries, and specifically the architecture and implementation of a Secure X.500 Border Directory Proxy Server on the XTS-300 that can function as (1) a secure gateway for controlling the release of internal directory information, or (2) a "shared repository" on which to store and make externally accessible a subset of internal directory information, or (3) a combination of the two.

2 Secure X.500 Border Directory Proxy Server Concept

As described in Section 1, a Border Directory is a directory owned by or operated by one domain (the "internal domain") but which can participate in the directory service of another, often broader, domain (the "external domain"). A Border Directory should be accessible from Directory User Agents (DUAs) and Directory Server Agents (DSAs) in both the internal and external domains. In this way, the Border Directory acts as the X.500 interface and interconnection point between the domains that allows the flow of directory information from an internal directory to users in the external domain.

A Border Directory can participate in a larger organizational (or inter-organizational) X.500 directory service while ensuring the protection of the internal domain. Architecturally, it should either operate on a different sub-domain/network separated from the internal domain by an appropriate security safeguard (e.g., a firewall or trusted guard), or it should implement such a safeguard itself at a sufficient level of assurance. The lower the integrity is of the external domain, the lower the integrity is of the sub-domain in which the Border Directory will operate, and the stronger the assurance (integrity and security) should be of the security safeguard protecting the external domain from the internal domain. This need for separation is even more critical when the internal and external domains operate at different classification levels or in different mandatory need-to-know categories/compartments.

As noted, the Border Directory provides a logical connection between the internal and external domains that allows a larger directory service to be built from what would otherwise be isolated "islands" of directories. In short, the Border Directory enables the creation of a single "virtual" global directory by logically integrating some or all of the information stored on physically separate directories while allowing them to maintain their physical separation and control which information they share with the "outside world".

Two types of connections between the Border Directory and the internal directory and external directories which it intermediates are possible:

- *Chaining*, in which requests for directory information are passed along from a DSA in one domain to a DSA in another domain;

- *Shadowing*, in which directory information is copied from the DSA in one domain to the DSA in another domain, so that requests can be satisfied locally by the target DSA that has been shadowed to, rather than having to chain a new directory request across domain boundaries each time that information is needed from the source (shadowing) directory.

There are pros and cons to each approach, but in general limiting connections to chaining ensures that stricter control of directory information can be maintained by its owners.

There is a common assumption that shadowing, and allowing directory requests to be satisfied locally, provides better performance. However, it has been observed that, because of the sometimes frequent need to re-synchronize directories that have shadowing relationships—re-synchronization that entails the repeated replication of entire, huge directory structures over the network—any performance gains in terms of bandwidth savings achieved by storing the replicated directory information locally are counteracted by the bandwidth required to repeatedly shadow the entire source directory over the network to achieve re-synchronization.

The Border Directory shall accept chaining of DSP requests from internal and external Directories on either side of it. The organization that "owns" the Border Directory may decide whether such requests are allowed to be chained beyond the Border Directory into the organization's internal Directory, in which case the Border Directory can act purely as a gateway between the external and internal domains, or whether the requests will be satisfied by the Border Directory itself, in which case the Border Directory will have to maintain its own directory Information Base (repository) containing the subset of internal directory information that is to be made accessible to the external domain.

The Secure X.500 Border Directory Proxy Server would need to establish agreements with directories in both the internal and external domains to enable it to access information across those boundaries. This would include determining, especially for shadowing, whether Simplified or Basic Access Control should be used for information shared across the boundary, whether cross-boundary chaining should be supported, and if so whether cross-boundary chaining should be limited to read-only accesses or whether the Border Directory should also provide cross-boundary modify-access (directory updates) from authorized Administrative DUAs.

2.1 Operational Environment

As illustrated in Figure 1, the Secure X.500 Border Directory Proxy Server would sit between an organization's internal directory and those external Directories to which it wishes to provide access to its internal directory information. The internal organization's directory information sharing policy—i.e., it's *release* policy—would specify a set of rules by which one could determine exactly which information from the internal directory may be shared with the external domain.

In most organizations, this "releasability" would be dictated solely by the internal organization's "need to know" based releasability policy.

In military/defense, intelligence, diplomatic, and other communities in which information is handled at multiple classification levels or is segregated according a non-hierarchical mandatory security policy (e.g., compartmentation or categorization), the policy defining which internal directory information may be shared via the Border Directory may be further complicated by the internal and external entities operating at different classification levels or *mandatory* needs-to-know. In these operational environments, in addition to providing directory firewall protections, the Border Directory would also act as a Trusted Guard at least the level of assurance required to maintain the mandatory separation of the "inside" and "outside" domains for everything except the strictly-controlled sharing of directory information according to the releasability policy enforced by the Border Directory. In these environments, it is anticipated that the Border directory system would be appropriately certified and accredited for operation.

The Secure X.500 Border Directory Proxy Server's trusted guard filters would not only validate

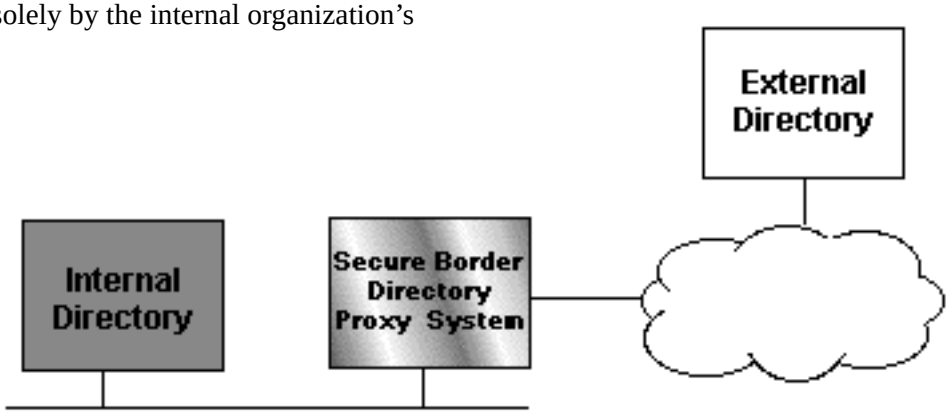

Figure 1: Border Directory Concept

sense of the external domain's "need to know". To achieve the secure, carefully controlled "release" of internal directory information to the outside world, the Secure X.500 Border Directory Proxy Server would provide directory-specific firewall protection mechanisms that filter and in some cases modify ("sanitize") directory information to ensure its conformance with the organization's the multilevel/inter-compartment/category releasability of directory information according to defined releasability policy rules, but it would further validate the correct performance of the Border Directory's firewall filters, to absolutely assure that no "unallowed" information has been inadvertently released by those filters in violation of the organization's mandatory security policy.

To provide the highest possible level of assurance and integrity protections, the Secure X.500 Border Directory Proxy Server would run on the Wang Government Services XTS-300™ Trusted Computer System, which has been evaluated at the Class B3 level by the National Computer Security Center (NCSC). In building the Secure X.500 Border Directory Proxy Server, Wang's engineers would integrate the existing, NSA-accredited and SABI-certified X.500 filtering and trusted X.500 routing capabilities of Wang's Defense Information Infrastructure (DII) Guard with additional new filters. Collectively, all these filters would implement the full range of potentially required directory firewall and trusted guard filtering. The filters could be configurable on a deployment-by-deployment basis, so that organizations that did not require strict mandatory security policy enforcement by the Border Directory could configure the only those filters required for the types of releasability control it desired, e.g., firewall filtering plus a meaningful subset of trusted guard validation filtering.

2.2 Secure Border Directory in PKI

In Public Key Infrastructures (PKIs), user identification and authorization information is usually distributed in two locations: in the user's encryption/certificate token (hardware or software-based) and in the PKI's X.500 directory. The user information stored in the token is usually that frequently-used user information that seldom or never changes—e.g., the user's ID and unchanging access authorization information. All other information is stored in the user's Attribute Certificate in the PKI's directory, for the very practical reason that it is much easier to manage frequently (or even occasionally) changing information centrally than to manage it by updating or reissuing the user token every time a piece of information changes. For this reason, the X.500 directory is a critical component of the PKI.

In environments where PKI operation may extend beyond the organization's internal boundaries, the Border Directory concept again applies. In this case, the Border Directory may store some or all of the user information that is to be used outside the organization's border, and the user may, in fact, have multiple attribute certificates stored in the Border Directory, with each certificate representing a different user "personality" or role, based on which external entity the user uses the certificate with. For example, in the environment depicted in Figure 2, a U.S. user may have a single attribute certificate stored on the U.S. Border Directory which gives him certain authorized permissions to all of the other Border Directories on the CCEB network; or he may also have one or more attribute certificates that give him different/additional authorized access permissions to individual Border Directories (e.g., according to bilateral agreements between the U.S. and the individual allies).

If the Trusted X.500 Border Directory is to be used as a PKI Certificate Repository, the schema should be designed in a way that easily supports PKI-specific requirements. Schema definition considerations would includes such things as which object classes and attributes should be mandatory, and whether the PKI uses any proprietary object classes or attribute types not included in the X.509:1997 standard, such as the object classes/attribute types defined in X.520 or PKCS standards. If any of the X.509 standard object classes are not supported by the PKI, or if any of the standard object classes used include mandatory attributes not supported by the PKI, the schema would have to reflect these exclusions and their operational implications on the Border Directory minimized if possible.

Similarly, if the PKI lacks support for any particular DAP or LDAP operations, the schema design must reflect this exclusion. The schema must also support LDAP handling of the Object Class Identification (OID) through use of mnemonics (e.g., cn, sn) instead of OIDs to identify attributes and object classes. The Border Directory schema must be synchronized with the DSAs and DUAs in the X.500 infrastructures on either side of it (internal and external) to ensure the correct mapping of the OID and LDAP mnemonic being used to represent it.

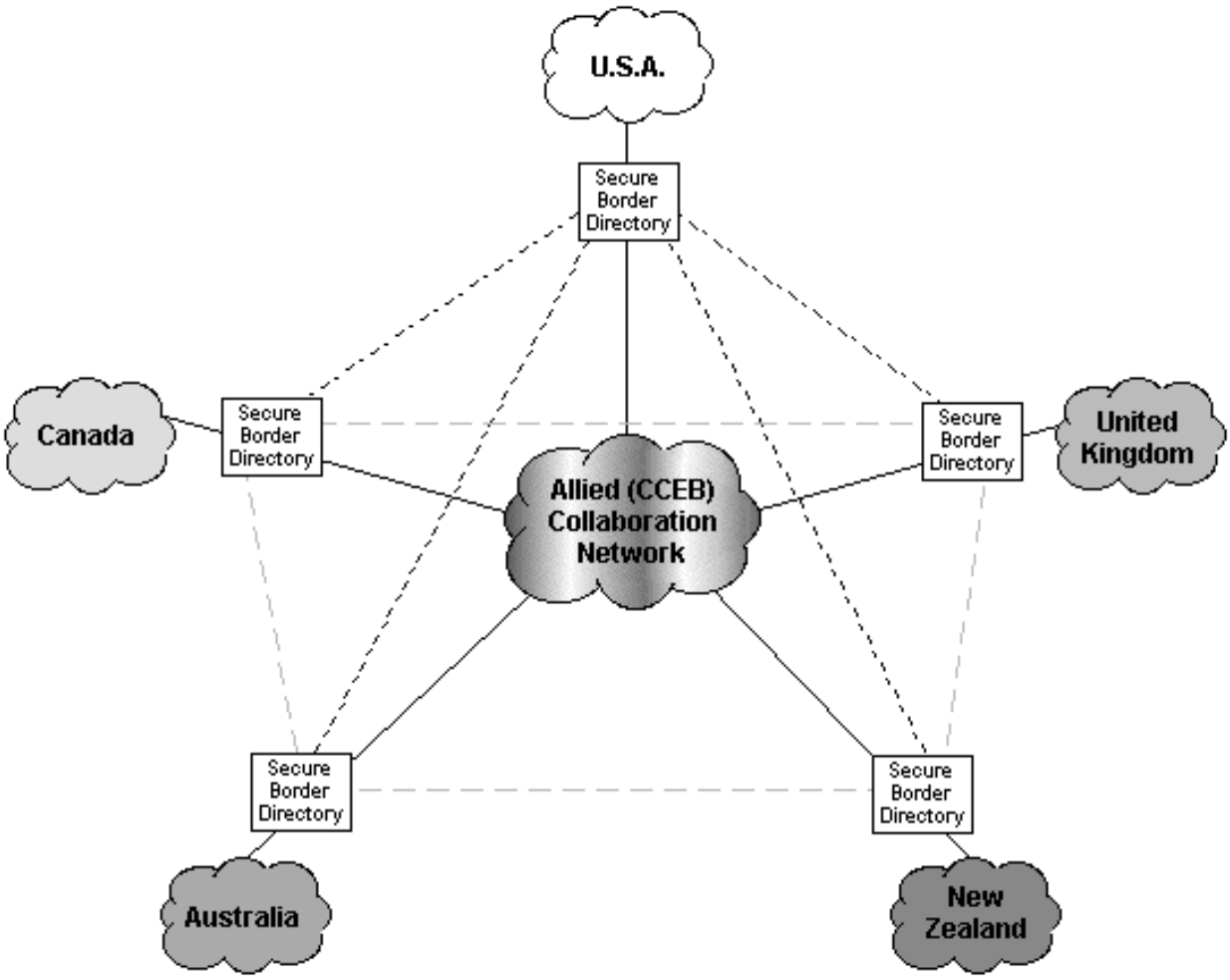

Figure 2: Secure Border Directories in a CCEB Distributed Global Directory System

3 Relevant X.500 Directory System Concepts

The following X.500 directory system concepts—mainly access-control concepts— and their relevance to the Secure X.500 Border Directory Proxy Server will be considered as we define the details of the Secure X.500 Border Directory Proxy Server implementation. If deemed relevant, we will then determine how they should be implemented. In some cases, the concepts described would be relevant only if the Secure X.500 Border Directory Proxy Server contained its own data repositories (Directory Information Bases [DIBs]), instead of acting only as a Trusted Gateway between internal and external directories. (These different Border Directory options will be discussed further in Section 4.)

An X.500 directory's security mechanisms must be able to refuse a user access to any information to which that user is not specifically authorised. Standard directory products, by and large, are hosted on low-assurance platforms and, thus, protect their information only through use of low-assurance discretionary access controls that are used to limit disclosure of certain directory information to only the owner of that information, to provide access to other information to only authorised users, and provide access to yet other information to "the world". The directory's discretionary access controls can hide single directory entries, specific attributes, even entire databases from unauthorised users.

When the directory in question is a Border Directory, however, and accessible to external as well as internal users, a mandatory need-to-know policy—enforced by mandatory access controls and other strong protection mechanisms—becomes highly desirable. When the "inside" is considered more sensitive (i.e, higher classification) than the "outside" (with sensitivity defined by classification level or mandatory category, caveat, or compartment), Border Directory enforcement of a mandatory access control policy at an appropriate level of assurance becomes imperative. In this case, the use of a trusted platform, such as the XTS-300, to run a policy-enforcing Secure X.500 Border Directory Proxy Server application makes sense.

3.1 LDAP-Based Directory vs. X.500

Recently, LDAP-based directory services, decoupled from X.500, have emerged, particularly in the Internet environment. These directory services are designed to leverage the use of LDAP while avoiding use of the more robust X.500 standards that provide for the implementation of a more sophisticated, intelligent directory service.

LDAP-only directory services either rely on one or more central directory repositories, often proprietary databases, with only point-to-point client-to-server access possible. Unlike X.500, this directory service scheme provides no server-to-server (directory-to-directory) access. So that with LDAP-only directory service, one central directory is expected to satisfy most if not all user LDAP requests. If that directory is unable to satisfy the request, a scheme of *referrals* is relied upon, wherein the primary directory returns the address of another directory which may be able to satisfy the request, then disconnects itself from the LDAP client. According to this scheme, the LDAP client must then connect to the *referred* directory and repeat its request. If the referred directory cannot fulfil the request, it may respond with yet another referral. Because the directories cannot communicate with each other on the LDAP client's behalf, the LDAP client must repeat its request to as many different directories as necessary until the desired information is found.

By contrast, in full X.500 directory service, the directories are able to *chain* (forward) requests "behind the scenes" (from the client's point of view) on behalf of clients, so that the process of finding the information requested by the client is the responsibility of the directory servers; the client need make its request only once, and to only one directory server. In the X.500 scheme, the client needs to be aware of and have access to only one, or possibly two directory servers—a primary and possibly an alternate. Similarly, each directory server needs to be aware of and have access to only the directories immediately next to it in the directory chain. From a security standpoint, this is scheme is much more desirable, as it minimizes the number of directories about which any client or server must have knowledge, and to which any client or server must have access.

3.2 Directory Interoperability through ACP 133

We expect to implement the Secure X.500 Border Directory Proxy Server in accordance with ACP 133, which is specification (developed by the CCEB and adopted by NATO) of an X.500-based allied directory. ACP 133 specifies that user (client) access from Directory User Agents (DUAs) to the allied directory will be achieved via Directory Access Protocol (DAP), with specifications for adopting Lightweight Directory Access Protocol (LDAP) under review. Directory-to-directory access is achieved via Directory System Protocol (DSP).

3.3 Access Control

ACP 133 Access Control features, including Access Control Statements, are defined both for Rules Based Access Control (RBAC) and Simplified (or Basic) access control. For RBAC, the Mandatory Access Privilege Set indicates the clearance and classification levels, including hierarchical, compartments, and categories/caveats of entities allowed to access the directory information. For Simplified/Basic access control, the Discretionary Access Privilege Set indicates the operations each user/client is authorized to perform on the directory information once mandatory access has been granted.

Directory access control statements may express:

- the unique identity of each protected item entered at the subtree level;
- the user class—including user name, user group name, as defined by type/role of users (e.g., using wildcard characters, or access

control lists of groups of user names);
- the access permissions to be granted (read, write, delete);
- the strength of authentication to be performed (strong or simple—the DUA (Directory User Agent) and DSA (Directory System Agent) must be able to mutually exchange strong bind tokens);

Indication of precedence may or may not be required.

3.3.1 Rules-Based Access Control

If used, Rules-Based Access Control (RBAC) should be able to enforce as many different RBAC policies as necessary, with an RBAC policy check performed determine the object ID of the RBAC policy to be enforced for a particular DAP or DSP operation. This policy check would be implemented using an ASN.1 decoder/encoder that could enable RBAC policy information (stored in files) to be implemented in ASCII form or in ASN.1 encoded form. On the XTS-300, the integrity of the RBAC policy files could be strongly protected by assigning those files in the STOP™ file system a higher integrity level than the that of the RBAC policy checking executable process that refers to those policy files.

Rules-based access control (RBAC) decision and enforcement functions operate at the subtree level (e.g., at level of "organizational directory"). An RBAC control policy stored in the Directory as an ASN.1 encoded file would be identified by a unique object ID. In this way, an individual external or internal Directory's RBAC policy could be invoked (retrieved and activated) by object ID. At a minimum, GENSER and COCOM (Intelligence) RBAC policies, and possibly also the NATO RBAC policy, should be enforced.

The Secure X.500 Border Directory Proxy Server's RBAC could also support application-level labels that indicate the hierarchical classification of data to be accessed. The intent would be to use X.411 labels with X.509v3 clearance extensions.

The concept of "configuration of components" as it pertains to the Secure X.500 Border Directory Proxy Server RBAC implementation would have to be defined to determine whether there is a requirement for same.

3.3.2 Labelling of Directory Entries

In military/defense, intelligence, and diplomatic operating environments it is anticipated that directory information with one or more of the following hierarchical labels would have to be understood and processed throughout any multilevel or inter-domain directory system:

1) Unclassified
2) Sensitive-but-Unclassified (no longer used?)
3) Restricted (NATO & CCEB)
4) Confidential
5) Secret
6) Secret Compartmented
7) Top Secret
8) Top Secret/SCI (1C and MC)

Most likely, only Unclassified, Restricted, Confidential/Secret, and Top Secret/SCI (including compartmented) directory information would be handled. It is also desirable for the Directory Servers to perform some degree of handling based on mandatory need-to-know labels (reflecting compartments, caveats, or categories).

In civilian and commercial environments other types of sensitivity labels (hierarchical and/or non-hierarchical) might be used, and would have to be understood and handled by the Directory Servers in those environments.

ACP 133 provides for directory entry labels with the following characteristics:

1) *Human-readable output* — Labels, when displayed on-screen or in printed output generated by the Directory, must be human readable. This may be achieved by using an ASN.1 decoder.

2) *Labelling of individual attributes in an entry* — Labels on ADUA-generated directory entries can be applied for the following attribute types (object classes): address list/ mail list; subtree. "Children" of these attribute types are not explicitly labelled; instead, they will inherit the attributes of their "parents'" labels.

3) *Labelling at the subtree*, or address list/ mailing list, level.

4) *X.411 label format including SDN.801-defined security categories* — Labels on directory information are applied by the ADUA using the X.411 label format (i.e., using address indicator groups), upon which the Directory Server may be able to take security decisions. ACP 117 may also be accommodated, although it is not strictly required.

5) *FIPS 188 Tag Types as defined in SDN.801*—FIPS 188 tag types are applied to directory information by the ADUA to indicate caveats/compartments. The Directory Server may be able to take security decisions based on these tag types.

6) *Labels bound to strength and/or type of algorithm* — It may be possible for the Directory Server to determine a directory entry's implied label based (in whole or in part) on the strength and/or type of encryption algorithm used.

7) *Extensibility* — RBAC policy enforcement should allow for configuration of additional security classification labels and/or tag types. These additional labels/tag types might be extensions/additions to X.411/SDN.801 labels and FIPS 188 tag types, or they might be organization-specific custom labels/tag types (e.g., NATO labels/tag types). Simple/Basic DAC policy enforcement should allow for configuration of new/additional handling caveats.

3.4 Authentication Service

The Industry-Government Open System Specification (IGOSS) Directory Profile defines eight authentication modes for a directory server. But few of these authentication modes are implemented in today's directory products, and it can be confusing to attempt to use eight different profiles that have only minor variations among them. The authentication profile(s) of organizations that share directory information must be known and supported by the Border Directory through which they share this information.

For Directory Access Protocol (DAP), IGOSS defines three Authentication Service options:

1) No authentication;
2) *IGOSS Authentication Mode 1:* Simple Authentication (user name + protected password);
3) *IGOSS Authentication Mode 2:* Strong authentication on bind (supported by PKI).

For Lightweight Directory Access Protocol Version 2 (LDAPv2), two options are defined:

1) Simple authentication (cleartext password);
2) Strong authentication (using Kerberos).

LDAPv2 simple authentication is inherently insecure to be sufficient for use by a Border Directory system. Meanwhile, Kerberos is not a universal standard, and thus is impractical for implementing strong authentication in many environments. However, the proposed LDAPv3 (RFC2251) standard provides for a *Simple Authentication and Security Layer* (SASL) that provides a choice of several simple authentication methods in addition to use of cleartext password. However, none of these modes are currently mandatory. In addition, standards for LDAP strong authentication mechanisms are evolving, and are a high priority within the IETF, which is expected to select and publish a method shortly. Thereafter, this method should quickly be implemented by X.500 vendors.

For Directory System Protocol (DSP), two options are defined:

1) *IGOSS Authentication Mode 7:* Simple authentication (name + protected password);
2) *IGOSS Authentication Mode 4:* Strong authentication (supported by PKI).

For Directory Information Shadowing Protocol (DISP), two options are defined:

1) *IGOSS Authentication Mode 7:* Simple authentication (name + protected password);
2) *IGOSS Authentication Mode 4:* Strong authentication (supported by PKI).

3.4.1 Strong Authentication Exchanges

Strong authentication entails the generation and exchange appropriate strong authentication information by a Directory Server with any other qualified DSA, as well as with a DUA or Admin-

istrative DUA, and the handling by the Directory Server of strong authentication information received from the other DSA or DUA/ADUA. At a minimum, this involves two-way mutually strong-authenticated Directory Access Protocol (DAP) binds between the Directory Server and the DUA/ADUA, while strongly-authenticated DSA-to-DUA binds may also be implemented via Secure Light Directory Access Protocol (SLDAP). Two-way mutually strong-authenticated DSP binds between the Directory Server and other DSAs are also desirable. In certain circumstances, one-way and three-way authentication exchanges may have to be supported.

For ADUA update operations, additional strong authentication may be desirable using SSL, SSH, static user ID+password, SecureID tokens, or another authentication method. Similarly, when the directory information to be accessed is considered sensitive, the strong two-way DUA-DSA binds for *read* operations may need to be augmented by additional strong authentication.

3.4.2 Strong Authentication Technologies

Strong Authentication (i.e., strong two-way binds) between X.500 directories are accomplished via generation and exchange by the directories of X.509 bind tokens. The options for implementing strong authentication include software encryption and hardware encryption tokens using FORTEZZA®, RSA, or a "brand-name" software- or token-based encryption. There is some doubt as to whether software-based encryption can be considered trustworthy enough to support strong authentication. For authentication to be truly *strong*, hardware tokens are preferable.

Wang Government Services has developed Access Control Decision Function (ACDF) software that makes an access control decision based on: X.411 security label (including MISSI SDN.801-defined security categories, which are closely aligned with the FIPS PUB 188 Standard Security Label); X.501 Clearance Attribute (including MISSI SDN.801-defined security categories); and the SDN.801-defined Security Policy Information File (SPIF). The SPIF is a signed configuration file that includes information about a specific security policy. The ACDF software that we have developed implements generic access control rules (permissive, restricted, enumerated) and uses the SPIF to obtain the specific information related to a security policy. This allows the owner of the directory to change the security policy details or to add a new security policy without needing to change the ACDF software.

If the security policy changes, then a new SPIF is distributed, but the ACDF software does not change. This should satisfy the extensibility requirement. Wang has tested the ACDF software for the GENSER and CAPCO (i.e. Intel Community PRBAC) security policies; NATO has not yet published an RBAC policy. Note that the DII Guard and all other ACP 120-capable DMS products use Wang's ACDF software to support the DMS ACP 120 automated access control requirements. In summary, the proposed Wang Secure Border Directory could use the ACDF software to fulfil all or some of the access control requirements.

3.5 Use of X.509 Certificates

Underpinning the strong authentication exchanges should be the use of X.509v3 certificates, which are used as "bind tokens" exchanged between the Secure X.500 Border Directory Proxy Server and the connecting Directories to implement strong authentication. The appropriate attributes/extensions of the X.509v3 certificate to be implemented for strong two-way authentication exchanges need to be determined and implemented—at a minimum, this would include the Clearance Attribute.

Extensions and handling of extensions when validating X.509 certificates should be implemented according to the SDN.706 certificate/CRL profile and RFC2459 Internet PKIX Certificate and CRL Profile guidelines, and not according to the Canadian SBKM guidelines (used in the Entrust PKI). However, the ability should exist to validate Entrust-generated X.509 certificate paths when cross-certificates are provided from the Entrust environment.

The ability should also exist to validate X.509 certification paths. Wang Government Services have implemented a freeware Certificate Management Library (CML) that performs X.509 certification path validation. The CML also provides optional local cache management functions and can optionally obtain data objects using LDAPv2 and ASN.1 encode/decode certificates and CRLs.

The CML is also able to process all X.509 certificate extensions through its ASN.1 decoding of all extensions, and use of those extensions as part of its certification path validation process. Additional mechanisms to support matching rules for some of the extensions may need to be developed.

Certificate handling could be implemented via a trusted process that uses Version 3 X.509 Certificates for strong authentication. This process could check a variable in the incoming X.509 Certificate indicating which extensions have been implemented for the particular certificate being authenticated. After checking this variable, the trusted process would reference the appropriate security policy (software) that will enable the trusted process to take security decisions based on the particular combination of extensions indicated for the particular certificate being processed—initially, only the Clearance Attribute extension would be configured. However, writing a configurable trusted process that checks for a variable in this way, rather than hard-coding the X.509 extension types, could enable the later addition of other extensions as they are adopted.

3.5.1 X.509 Attribute Certificates

X.509 Attribute Certificates are not yet defined in ACP 133. As their definitions emerge, their appropriateness to and method of implementation would have to be determined.

3.5.2 Version 2 CRLs and ICRLs

Version 2 Certificate Revocation Lists (CRLs) and Indirect Certificate Revocation Lists (ICRLs) are lists of X.509 certificates that are no longer valid and should be revoked rather than used by PKI applications. CRLs and ICRLs may be stored in a Directory according to the CRL's *reason code*. If this is done, a different handling policy for a CRL may be implemented, depending on the *reason code* associated with that CRL or ICRL. Other CRL processing functions may include checking at a defined time interval for CRLs with a particular reason code—for example, checking every four hours for CRLs with a "key compromise" reason code. The schema of the Directory must be implemented in such a way that CRLs can be included under certain directory entries.

3.5.3 OCSP and RCSP

Increasingly, use of periodically-issued CRLs to verify the revocation status of an X.509v3 certificate have begun to give way to realtime verification mechanisms, such as Online Certificate Status Protocol (OCSP; RFC2650). OCSP relies on the existence of realtime "Responders," which can be queried to check a certificate's validity. OCSP has also given rise to similar protocols, such as Realtime Certificate Status Protocol (RCSP), which attempt to improve on OCSP.

Both OCSP and RCSP are designed to be used in lieu of or as a supplement to checking against a periodic CRL to determine the revocation status of a particular certificate, particularly by applications where obtaining the revocation status of a certificate on demand is required. In short, OCSP and RCSP are designed to address well-known concerns about CRL size and timeliness.

In cross-boundary PKIs wherein the entity that is operating the OCSP or RCSP Responder did not also issue the certificates being validated, it is not clear how such services would obtain their revocation information, or prove that the information was timely, accurate, and authentic. There is no explicit definition in the OCSP and RCSP draft specifications of a "delegation certificate", an "attribute certificate", or another such construct to be generated by the CA which produces the status information. The OCSP and RCSP drafts only define the mechanisms by which the trust relationships may be established between Responders and Clients. Constraints upon the structure and enforcement of those trust relationships remain a local responsibility, which begs the question of how "local responsibility" can be defined and established in a cross-boundary PKI environment.

The implementation of OCSP in a cross-boundary PKI/Directory environment in which there is no single local authority responsible for acting as the OSCP responder, the relative ease (and security) of using OCSP vs. traditional CRLs needs to be carefully considered. Until such issues are resolved, use of public key technology across security boundaries may remain better served by use of CRLs.

3.6 Confidentiality of Information

The typical Directory Server is expected to maintain the confidentiality of transmitted and stored information, both through use of discretionary

access controls and by preserving the encryption of directory entry content when stored in the DIB or transmitted to another DSA or DUA. While the Directory Server must understand and process X.500 protocol commands (DSP, DAP, DISP), it requires no knowledge or awareness of the actual content of the directory entries being operated upon by those commands.

In the case of a Secure X.500 Border Directory Proxy Server, discretionary access controls may be augmented by mandatory access controls. Further, if filtering is to be performed on directory transactions (including content of directory entries) to determine its releasability, unlike other Directory Servers, the Secure X.500 Border Directory Proxy Server will have to be able to read and understand, to the extent required by filtering, the content of directory entries.

If the Directory Server itself performs encryption, that encryption should probably be implemented according to X.500-97 via templates that include a selective field indicating the encryption key[s] and/or algorithm defined for the intended recipient. In this case, key management issues must also be considered, including which key management protocol to use (KEA, Diffie-Hellman, etc.).

3.7 Integrity Protection

Integrity of X.500 operations (DSP, DISP) may be implemented by digital signature of those operations by their originator. Digital signature may be implemented using FORTEZZA® or another encryption technology.

If the Border Directory system is expected to perform digital signatures on its operations, the digital signature process should be configured with the encryption methods of all of the Directory Servers it will connect to. For every operation to be signed, the Border Directory's digital signature process must first determine which digital signature method to use based on the intended recipient of the operation.

3.8 Non-Repudiation and Auditing

Non-repudiation of events can be achieved through the use of event logging and security auditing on the Directory Server, at both the operating system and Directory Server application levels. At a minimum, digital signature operations and accesses and manipulations of the Directory Server by the administrator should be audited; other operations/events should be digitally signed for non-repudiation purposes. Additional or alternative non-repudiation methods may also be identified. Relevant audit events must be defined, and the combined operating system audit logs and Directory Server event logs should be correlated. Methods for reducing audit, and format and content for audit reports, should also be determined, including whether the combined audit trail can/should be re-formatted and exported to an intrusion detection or audit reduction tool.

3.9 ADUA Considerations

A minimum set of security features should probably be supported by any Administrative DUAs that are authorized to directly connect to the Border Directory to maintain the information therein (if the Border Directory is configured to store information locally). These security features should include:

- support for strong authentication exchanges,
- digital signature of operations,
- use of Version 3 X.509 certificates,
- data labelling,
- content encryption.

3.10 Locally-Controlled Releasability

Operation of the Border Directory could be simplified through the local implementation of "release authorization flags". These "flags" would actually be in the form of a newly-defined attribute within each user directory entry in the internal directory (as allowed for by the X.500 standard for schema extensibility). The Administrative DUAs that originally created that directory entry would simply include the "release authorization attribute" in the entry at time of creation, indicating whether the entry could be published outside the internal directory, i.e., provided to the Border Directory system. The Border Directory system could then verify the presence of the "release authorization attribute" as a prerequisite to making the directory entry accessible to external entities.

4 Secure Border Directory Proxy Server Implementation

4.1 Concept of Operation

The Secure X.500 Border Directory Proxy Server would be used to separate internal and external domains while enabling the strictly-controlled publication of directory information from the internal domain to directory users in external domain. The Secure X.500 Border Directory Proxy Server could be used, for example, between Directories from different countries (e.g., the U.S. and its allies). It could be used by the defense/ military, intelligence, and diplomatic communities to implement "Community of Interest" separation while enabling inter-community Directory information sharing.

Similarly, the Border Directory could be used by other government agencies to enable inter-agency directory sharing. It could also be used in commercial or public service applications—for example by banks to enable the sharing and exchange of PKI (certificate) information with other banks, by health care organizations that wish to share and exchange public information, or by any other commercial organizations that wish to share some directory information with other organizations or with the public at large. In all cases, the Border Directory would ensure that only the limited subset of internal directory entries and attributes designated by the internal directory owner(s) would be replicated to the outside world, and that no access would be allowed from the outside the boundary into the internal network.

4.1.1 Directory Information Filtering

In addition to functioning as a standard X.500 directory (with or without a DIB, depending on the implementation requirements), the Secure X.500 Border Directory Proxy Server would provide a full set of Directory Information filtering capabilities to ensure that the information it made available to external entities strictly conformed with the internal organization's policies governing Directory information releasability.

The filtering in the Border Directory would be intended to achieve two different functions:

1) *Firewall-type filtering function,* whereby certain types of directory information would be prevented from release, while other directory information would actually be modified, or more often "sanitized", to ensure its conformance with the release policy ("sanitization" means selective deletion of a designated part of the information, as determined by a sanitization rule, while maintaining the remainder of the information). These filters would be designed to reduce the source information in a directory entry or replication to render that information releasable. Simply, by and large, the firewall filters enforce a "change-and-release" policy.

2) *Trusted guard filtering function,* whereby the correctness of the firewall filtering would be validated, and other releasability criteria would also be validated to ensure strict conformance with the releasability policy. Unlike the firewall-type filters, these filters would be designed not to modify information but to prevent the release of unallowed information. Simply, the Guard filters enforce a "go/no-go" policy.

The Secure X.500 Border Directory Proxy Server would also ensure that no external directory can chain into the internal network to retrieve internal directory information. The Border Directory could also ensure the enforcement of different access control policies pertaining to the directory information depending on which side of the boundary the requester is on (the assumption being that internal users will have more access privileges than external users). Separate domain-based policies could also be defined for different external users, e.g., alliance member nations.

4.1.1.1 Firewall-Type Filters

The kinds of firewall filters that could be implemented in the Secure X.500 Border Directory Proxy Server include:

- *Attribute filter*—would enable the configuration of a set of attributes that may or may not be requested by inside users when querying outside directories. The attribute filter could be either be configured to reject the X.500

operation if the unallowed attributes were present or to "sanitize" (remove) the unallowed attributes from the X.500 operation then forward the sanitized X.500 request to the external directory

- *Knowledge Reference Filter*—would remove specified knowledge reference information from the Directory operation (e.g., from the DSP chaining argument) before releasing the sanitized operation to the external directory. The information to be removed could include:

  - subordinate knowledge references;
  - continuation reference & referral information;
  - trace information about system names, IP addresses, etc.;
  - cross references in chained results;
  - partial outcome qualifier.

- *Shadowing Subset Filter*—would ensure that only information (sub-trees, entries, and attributes) allowed outside the domain are ever shadowed outside the domain. For example, this filter could check and sanitize the data to ensure that only a defined subset of the user entry was released (e.g., name, X.400 address, telephone number, certificates, CRL, CKL). This filter must be able to actually parse the shadowed data to find the access control information in the DISP Protocol Data Unit (PDU). Restricted entries (access control="deny") would then be removed from the *updateShadow* PDU.

- *Shadowing Entry Based on Releasability Authorization Attribute Value (i.e., publish flag)*—This filter will check to ensure that only a defined subset of user entries is released via shadowing. This filter presumes the inclusion of the "release authorization attribute" described in section 3.1.10. If this attribute were included in each internal directory entry, the filter would check that attribute's value to determine whether the entry was designated "releasable" or not, i.e., whether or not it could be shadowed to the external directory. Use of the Releasability Authorization Attribute would eliminate the need to maintain a separate list of DNs on the Secure X.500 Border Directory Proxy Server to indicate which entries could or could not be replicated. This filter would override any conflicting decisions taken by the shadowing subset filter. For entries that did not include the Release Authorization Attribute, we would have to determine whether releasability would be determined on Access Control Information (stored in an Access Control List on the Border Directory).

4.1.1.2 Trusted Guard Filters

A number of X.500 filters are already implemented within the Wang DII Guard. These filters would be integrated into the Secure X.500 Border Directory Proxy Server, and include:

- *Directory Protocol Filter*—allows or denies access to the directory by DAP, DSP, and/or DISP on a per-flow basis. This filter can be configured to ensure that chaining only occurs in one direction, i.e., from the internal directory to the external directory, and not in the other direction.

- *Directory Operation Filter*—determines whether to allow or deny the performance of different directory operations (e.g., read, list, compare, search, abandon, add entry, delete entry, modify entry, modify Distinguished Name) on the protected directory information. In addition, each of these operations can be required to be digitally signed and/or implemented with strong authentication.

- *Distinguished Name (DN) Filter*—verifies the requester's DN by comparing it to an Access Control List (ACL) of allowable DNs stored in the Guard. This filter also verifies the responding DSA's DN to ensure that it appears on the Guard's ACL of DSAs from which the Guard is allowed to receive directory results. These filters can be configured to support user class filtering, whereby the administrator can create access control groups or permission categories based on user role (e.g., administrator, CA, user). The Guard will then grant or deny a specific request by mapping the operation/action requested to the permissions (for operations) the group is allowed to perform.

- *Directory Information Shadowing Protocol (DISP) Filter*—verifies that the information in

a specific directory shadowing agreement is correctly configured. The shadowing agreement information to be verified includes:

- Agreement ID
- Version number
- Information allowed to be replicated
- Master/shadow relationship
- Direction of replication (i.e., low to high)
- DSA network validation
- Existence of a shadowing agreement.

In addition to these existing Guard filters, the following new filters would be added to the Secure X.500 Border Directory Proxy Server:

- *Override Access Control Filter*—This filter would act as a domain-based security filter to provide a more restrictive access control policy implementation on any data leaving the domain than is applied to the same data when access from within the domain. For example, internal users may be able to update certain attributes within their own directory entries, but no updates would be allowed on those entries after they leave the domain.

- *Hide Internal User Information Filter*—This filter would ensure that all requests leaving the Secure X.500 Border Directory Proxy Server would appear to originate from the Directory itself, and not from the true internal originator. The purpose is to hide the identity of the internal users, and would involve replacing the *requestorDN* in the DAP *commonArgs* or the *originatorDN* in the *chainingArgs* with a *TrustedDirectoryServer* DN value.

- *LDAP Version 3 support and filters*—TBD.

4.2 Architecture

In its initial implementation, we envision the Secure X.500 Border Directory Proxy Server as a Trusted X.500 Gateway that will enable the strictly controlled release of directory information from the internal directory to external entities, and the strictly controlled release of directory requests from internal users to external directories. However, initially, the Secure X.500 Border Directory Proxy Server will not store any directory information locally, i.e., there will be no Directory Information Base (repository) on the Secure X.500 Border Directory Proxy Server.

Construction of the Secure X.500 Border Directory Proxy Server will involve integration of the existing Wang DII Guard X.500 DSAs, X.500 router, and X.500 security filters with additional X.500 security filters (the new firewall and Trusted Guard filters described in Section 4.1).

Hosted on the Wang XTS-300™, the Secure X.500 Border Directory Proxy Server will most often be used to interconnect only two networks. However, the XTS-300™ can support up to four Ethernet (10BaseT/100BaseT) connections, so the Server could be used as the mediation point between up to four networks. Figure 3 illustrates the internal architecture of the proposed Secure X.500 Border Directory Proxy Server configured with three network connections.

Because the XTS-300™ is a B3-evaluated system, the connected networks (and their associated internal Secure X.500 Border Directory Proxy Server components) can reside at multiple classification levels (or non-hierarchical mandatory compartments, caveats, or categories), with the XTS-300™ Trusted Computing Base mandatory security and integrity access controls enforcing the strict separation of those different-level components, except when they are allowed to share information only via the Trusted X.500 Router according to the filter-enforced information releasability policies.

Aside from the very small, simple Trusted X.500 Router process, none of the components within the Secure X.500 Border Directory Proxy Server need to reside within the XTS-300 Trusted Computing Base. Instead, each of these processes—including the network interfaces, the DSAs, and the security filters—will operate at the single mandatory security and integrity level corresponding to that of the network with which they are associated. By absolutely minimizing the amount of TCB-privileged software in the Secure X.500 Border Directory Proxy Server, we hope to minimize the effort required to certify and accredit it.

Later, we will determine whether it makes sense to actually store directory information on the Secure X.500 Border Directory Proxy Server,

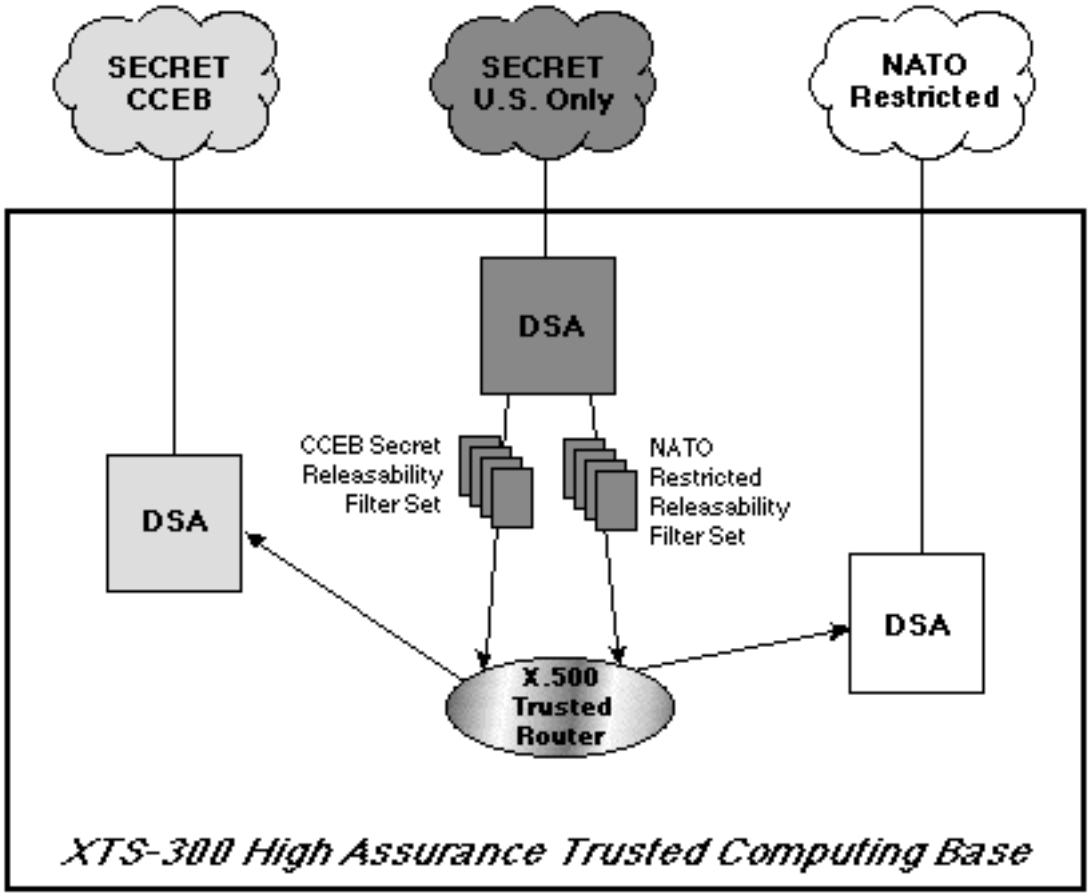

Figure 3. Initial Secure X.500 Border Directory Proxy Server Architecture (without DIBs)

rather than using it as a Trusted X.500 Gateway. This would entail porting some form of database (relational, flat file, hierarchical) to the XTS-300, with a separate database process providing the repository function to each DSA at each different security level supported on the system. Figure 4 illustrates the Secure X.500 Border Directory Proxy Server with the DIBs added.

Based on our experiences in implementing a Multilevel Secure X.500 Directory Server pilot for the U.S. Air Force's Rome Laboratory in 1996-1997, and our analysis of actual operational requirements, we have determined that an actual multilevel secure (MLS) Directory Server with an MLS DIB would not be practical to implement, nor is it desired by potential users of the Secure X.500 Border Directory Proxy Server. The only feasible way to implement an MLS DIB would be to use an existing MLS database product. Because MLS database management systems are virtual stand-alone products in terms of their handling of trusted labelling, this means that a MLS database like Trusted ORACLE incorporates its own Trusted Computing Base which does not derive its mandatory access labels from the underlying operating system. To the underlying operating system, the database appears, in fact, to be a single large file (or group of files) residing a single security level (system high) in the trusted file system. It makes little sense, when the underlying operating system TCB is evaluated at the B3 level, to overlay it with a database TCB that has been evaluated only at the B1 level, if it has been evaluated at all.

Our choice of Trusted RUBIX for the Rome Lab pilot was based, in part, on the fact that Infosystems Technology Inc., the owners of the product, were willing to re-engineer their system to minimize its TCB and derive, to some extent, the security labelling from the underlying XTS-300 TCB; further discussions led to a paper redesign of Trusted RUBIX to, in fact, eliminate

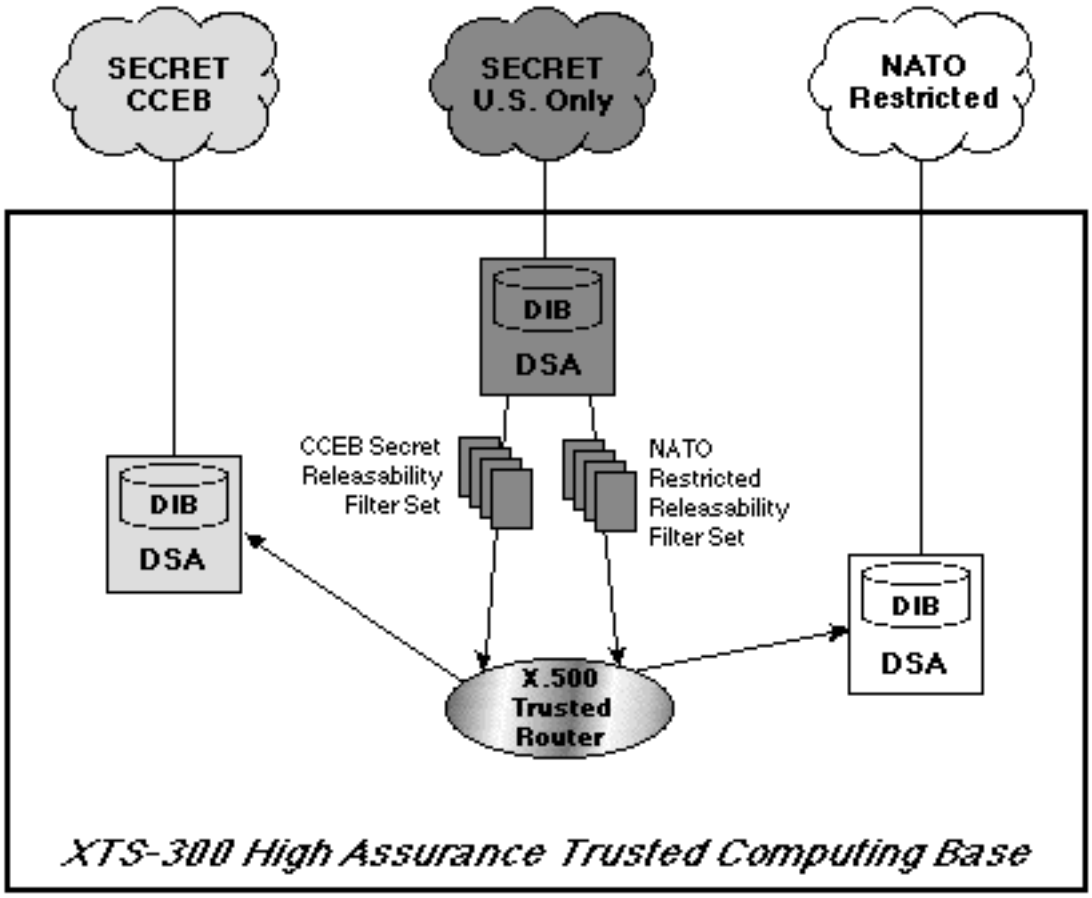

Figure 4. Second-Phase Secure X.500 Border Directory Proxy Server Architecture (with DIBs)

the TCB altogether, and instead use the STOP™ file system (and security and integrity labelling mechanisms) as the RUBIX TCB. However, it has not been proven that an MLS database is a necessary or even a desirable part of a Secure X.500 Border Directory. Moreover, during the Rome Lab pilot project, it was revealed that despite supporting ANSI SQL, the Trusted Rubix SQL interfaces were not always sufficient to satisfy the SQL interface requirements of the OpenDirectory X.500 DSA (which uses a relational database as its DIB), and a number of modifications had to be made both to the RUBIX SQL API and the OpenDirectory SQL API to get the two products to interoperate. Given this further complication, we feel it would make more sense to use a database that has already been fully integrated with whatever X.500 DSA product we ultimately choose for our Secure X.500 Border Directory Proxy Server.

—END—