IBM Tivoli Storage Manager 6.2.1
Security Target

Version: 2.8
Status: Release
Last Update: August 10, 2011

IBM, IBM logo, GSKit, iKeyman, and ICC are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft Windows 95, Microsoft Windows 98, Microsoft Windows ME, Microsoft Windows NT, Microsoft Windows 2000, Windows 2000 Professional and Advanced Server , Microsoft Windows Server 2003, Microsoft Windows Server 2008 and Microsoft Windows XP are trademarks of Microsoft in the United States, other countries, or both.

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Copyright (c) 2011 IBM Corporation or its wholly owned subsidiaries.

Document History

Version	Date	Summary	Author
2.0	2009-07-08	Initial Draft for TSM 6.2.1 – Based upon evaluated ST from TSM 5.5.1 evaluation.	SAIC
2.1	2010-06-04	Updated to change the set of supported host platforms in the IT environment.	SAIC
2.2	2010-06-28	Updated for compliance with CC version 3.1	SAIC
2.3	2010-10-11	Updated to respond to evaluation issues	SAIC
2.4	2011-04-28	Corrected minor inconsistencies	SAIC
2.5	2011-05-04	Removed misstated SFR	SAIC
2.6	2011-05-19	Resolve minor issues from evaluation	SAIC
2.7	2011-07-29	Resolve questions from DSD	SAIC
2.8	2011-08-10	Fix typo	SAIC

Copyright © IBM Corp. 2010-11

Table of Contents

Copyright © IBM Corp. 2010-11

List of Tables

List of Figures

Terminology

This document defines the following terms in order to improve understanding.

Product	All components that comprise the Tivoli Storage Manager 6.2.1 (i.e., the set of TOE and non-TOE components).
TOE	The set of components of the product that comprise the evaluated configuration.
Non-TOE Components	The components of the product outside the scope of the evaluated configuration.

Copyright © IBM Corp. 2010-11

1 Introduction

This document defines the Security Target (ST) for the Common Criteria (CC) evaluation of Tivoli Storage Manager 6.2.1 (TSM).

1.1 Security Target Identification

The following table contains the Security Target identification information.

Table 1-1 - ST Identification Information

Attribute	Description
ST Title	Tivoli Storage Manager 6.2.1 Security Target, Version 2.8, August 10, 2011.
TOE Identification	Tivoli Storage Manager 6.2.1 Extended Edition
Developer	IBM
Distributor	IBM
Sponsor	IBM
Keywords	Tivoli Storage Manager, TSM, Backup, Archive, Password Policy, TLS

1.2 Security Target Overview

The target of evaluation (TOE) is Tivoli Storage Manager 6.2.1. This ST describes the TOE, its boundaries, environment, security requirements, and security functions.

TSM is a software application that employees a client-server architecture to perform enterprise-wide data backup, archival, and restoration supporting multiple operating systems and multiple media storage types. This security target includes both client and server components.

1.3 Common Criteria, and Protection Profile Conformance Claims

The following table contains the Common Criteria and Protection Profile (PP) conformance claims of this ST.

Table 1-2 – CC and PP Conformance Claims

Claim	Description
Evaluation Assurance Level	**EAL3** augmented by **ALC_FLR.1**
Protection Profile Conformance	None
CC Version	[CC], [CEM]
Part 2 Conformance	Extended
Part 3 Conformance	Conformant

1.4 Terminology

The following table contains terminology used in this ST.

Copyright © IBM Corp. 2010-11

Table 1-3 - Terminology

Term	Description
Account	Represents an Administrator, Client Node, or Server in the Server's database, including authentication credentials.
Administrative Client	An independent administrative command line interface, separate from the Console, used to administer the Server. The Administrative Client executes independently of the Server, but provides similar functionality.
Authentication Protocol	The protocol used for authentication between Client Nodes and Servers, Administrative Clients and Servers, and between Servers.
Backup/Archive Client	The TOE software on the Client Node.
Client Node	The component of TSM that performs the backup, archiving, and restoring of data on a client system.
Console	The administrative command line interface provided by the Server.
Database	The data repository used to store account information, backup information, etc., similar to a relational database.
Data Storage	The data repository used to store backup and archive data, commonly comprised of tape drives, tape library, optical media drives, and hard drives.
Server	The server component of the TSM product.
Storage Server	The system that contains the Administrative Client, Database, and Server.
User	A user can be a client node account user or an administrator account user.

Copyright © IBM Corp. 2010-11

2 TOE Description

2.1 Introduction

The TOE is a data backup, archive, and restoration software solution. It provides the ability to backup data from one or more computers, known as Client Nodes, to another computer known as the Server Node (see Figure 1) or to backup data from a network attached storage (NAS) file server (see Figure 4). The TOE supports backup of a NAS file server directly to a TSM Server or to tape devices directly connected to the NAS. The TOE also provides for the retrieval and restoration of the data from the Server Node to the Client Node or NAS file server file server as well as providing functions for the management of the TOE and its data. Use of the TOE with a NAS file server is optional. The NAS file server is part of the environment.

The Client Node consists of the Backup/Archive Client (a.k.a. B/A Client). The Server Node is comprised of an Administrative Client CLI, Database, Data Storage, and Server (the Server program being the hub of all activity). The B/A Client software communicates directly with the Server to backup and archive data to the Storage Server, restore data from the Storage Server, and manage data while it resides on the Storage Server. The B/A Client also provides the command line interface for administrators of the Client Node to interact with the Server. B/A Clients communicate with the Server over an TLS connection with a protocol that employs mutual authentication. Additionally, there's a unidirectional, unauthenticated communication path from the Server to a B/A Client that allows a Server to request a predefined backup of the Client Node. This communications path is used for scheduled backups initiated by the Server.

The TOE can also be used in a configuration that provides the ability to backup and restore a network-attached storage (NAS) file server (See Figure 4). A NAS file server is a dedicated storage device with an operating system that is optimized for file-serving functions. The NAS file server must be Network Data Management Protocol (NDMP) compliant. The TOE controls the backup and recovery of an NDMP-compliant NAS file server, without requiring the installation of additional software on that file server. The TOE uses NDMP to instruct the NAS file server to backup its data to a tape device directly connected to the NAS file server or to a tape device controlled by the TSM server. The TOE supports only the ability to backup the entire volume (not individual files) from a NAS file server when acting as an NDMP tape server. Operations on a NAS file server are initiated through the command line interface of the B/A Client and not through an administrative/user interface offered by the NAS file server. Data that may be transferred between a NAS file server and the TOE is not protected by the TOE using cryptography.

The Administrative Client CLI is a command line administrative interface that's used to manage the TOE and TOE data stored on the Server Node. It communicates to the Server over an TLS connection with a protocol that employs mutual authentication.

The Server uses an internal version of the IBM DB2 database (hereafter referred to as the "Database") to store information about each Client Node. The Database is also used to store information about each Client Node's backed up/archived data as well as to store other data needed to manage the TOE. The actual backups and archives are stored in a data repository known as Data Storage which may consist of disk drives, tape drives, tape library, optical media drives, etc.

Password-based accounts are created and maintained on the Server. These accounts are used by the Server to control access to the data on the Server and to maintain the Server and Server data. Client Nodes and the Administrative Client use accounts to login to the Server. Since the accounts are maintained by the Storage Server, the Server performs the authentication. Additionally, the Server has its own command line administrative interface called the System Console which doesn't enforce authentication, thus, requiring this command line to be protected by the environment.

Copyright © IBM Corp. 2010-11

The B/A Client can also encrypt backup and archive objects locally prior to sending these objects to the Server using a separate encryption password known only to the B/A Client. It can also decrypt these objects when it retrieves them from the Server. These passwords are known as the "data encryption passwords".

The TOE provides password quality enforcement for authentication passwords including password expirations. It also supports password generation for both authentication and data encryption passwords.

The Server provides for controlled data sharing between Client Node accounts. Data saved by a Client Node account on the Server has a list of rules that a Client Node account can modify to grant other Client Nodes access to its data on the Server. It can also remove rules from the list thereby denying access.

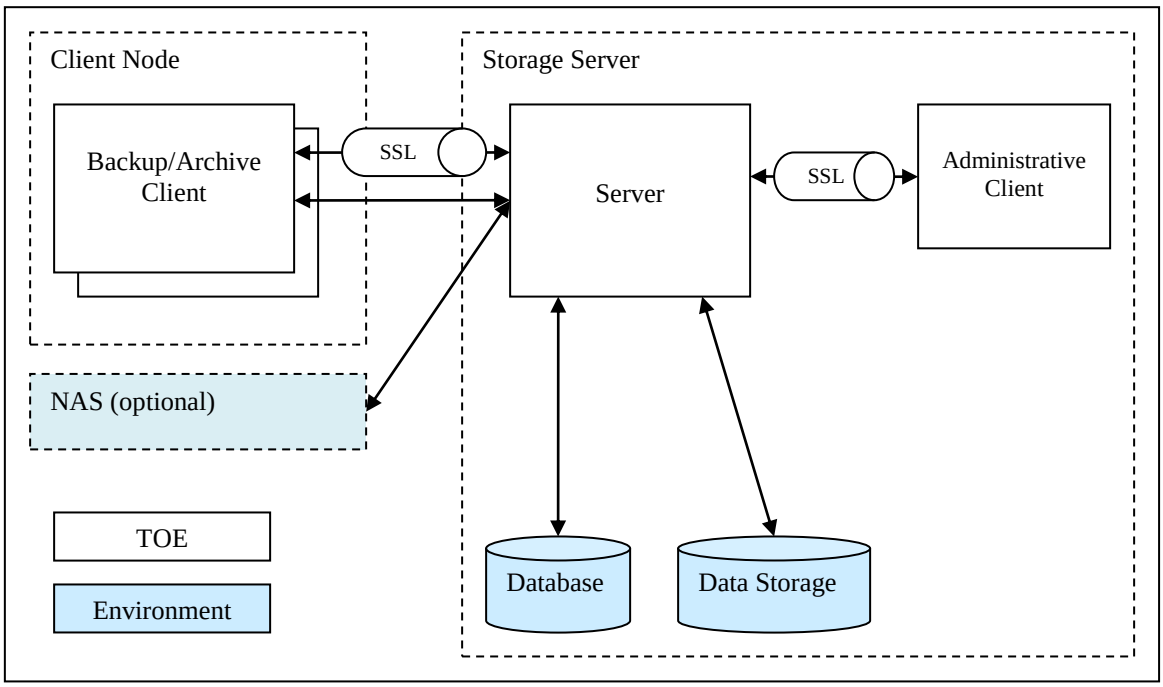

Figure 1 - TOE Logical Overview

The following figures show a handful of common ways that the TOE can be installed. Figure 2 shows a standalone installation of the TOE where the system being backed up is also the system containing the backups. In a standalone environment, a single system contains the entire TOE.

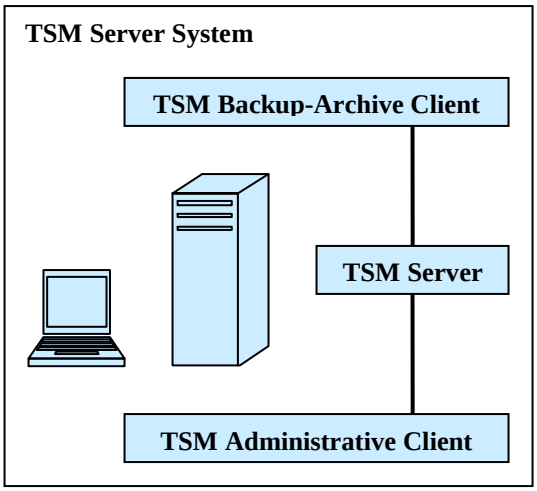

Figure 2 - Standalone Environment

Figure 3 shows a single server network environment where an enterprise contains one Server and multiple Client Nodes (where the system that contains the TSM server is also a client node).

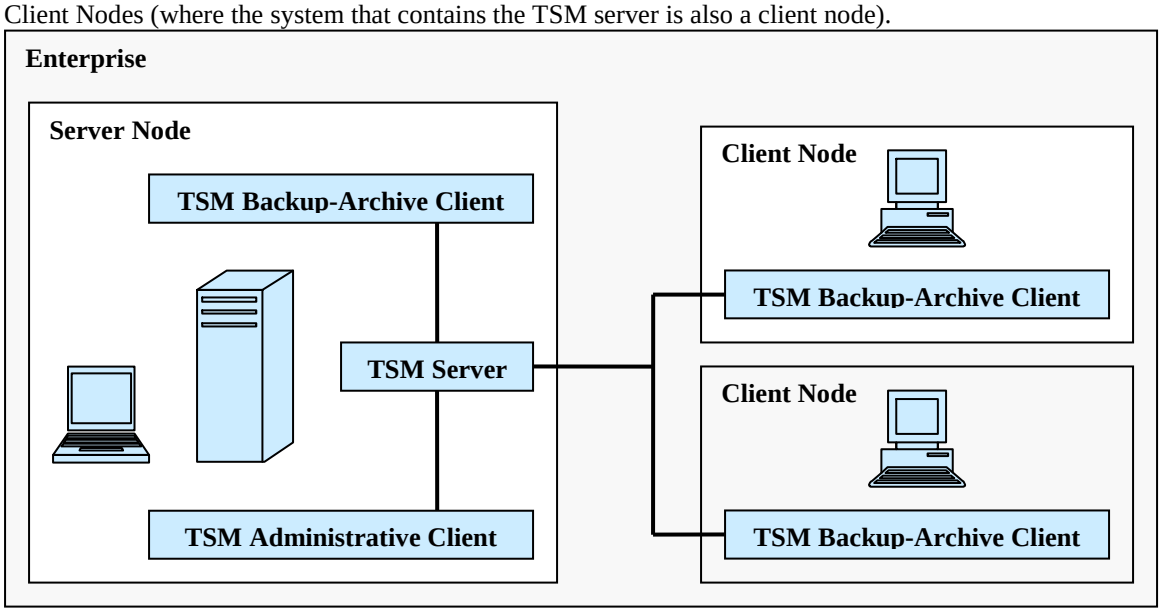

Figure 3 – Single Server Network Environment

Server to Server communications is not supported in the evaluated configuration.

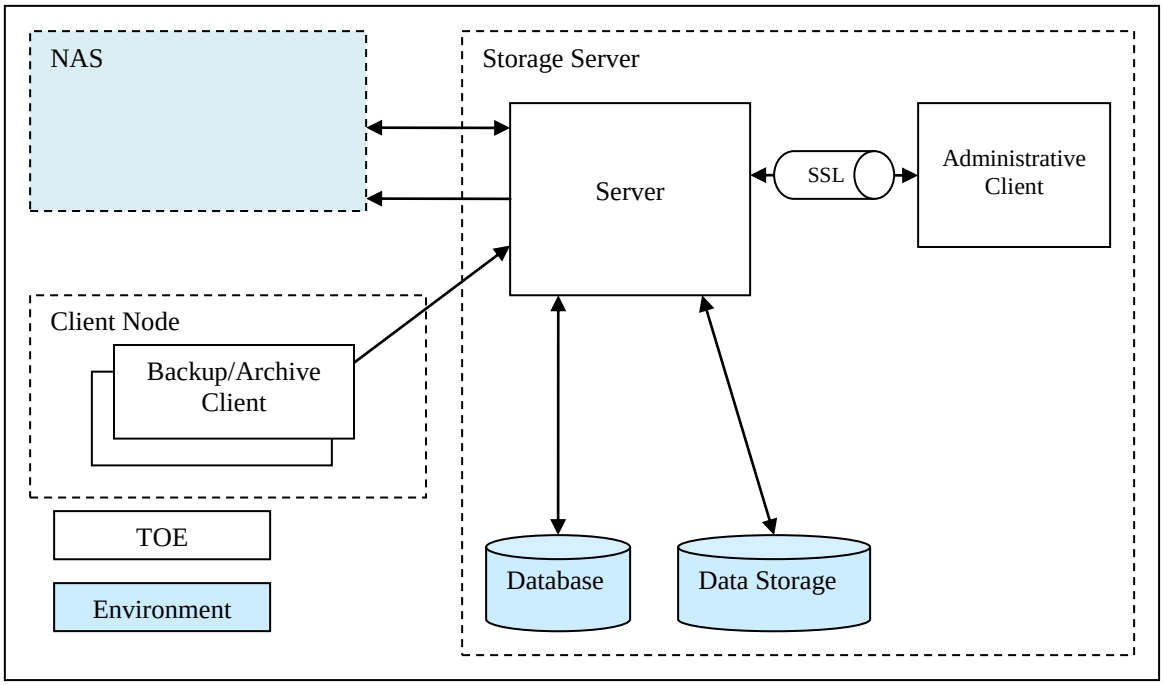

Figure 4 - Network Environment with NAS Device

2.2 TOE Overview and Boundaries

Figure 5 depicts a more detailed view of the TOE including a rough breakdown into functional components, the environment, and the users.

In the Client Node portion of the figure (upper half), the Backup/Archive Client has been subdivided into multiple components. The B/A Client CLI is the interactive human interface for the B/A Client and is the component that communicates with the Server. The B/A Client CLI has a configuration file for configuration data. The B/A Client CLI can store both an account password and a data encryption password locally so that automatic backups can be performed without human intervention. On Windows systems, the password is stored in an access protected entry in the registry. On non-Windows systems, the password is stored in an access protected file. Both the B/A Client CLI and B/A Client Scheduler Agent have an error log files where they write diagnostic error messages. The B/A Client also has a keystore file for storing the Server's public key. A brief description of the other components can be found in section 2.2.2. Additionally, the points of human interaction are shown in the figure.

Copyright © IBM Corp. 2010-11

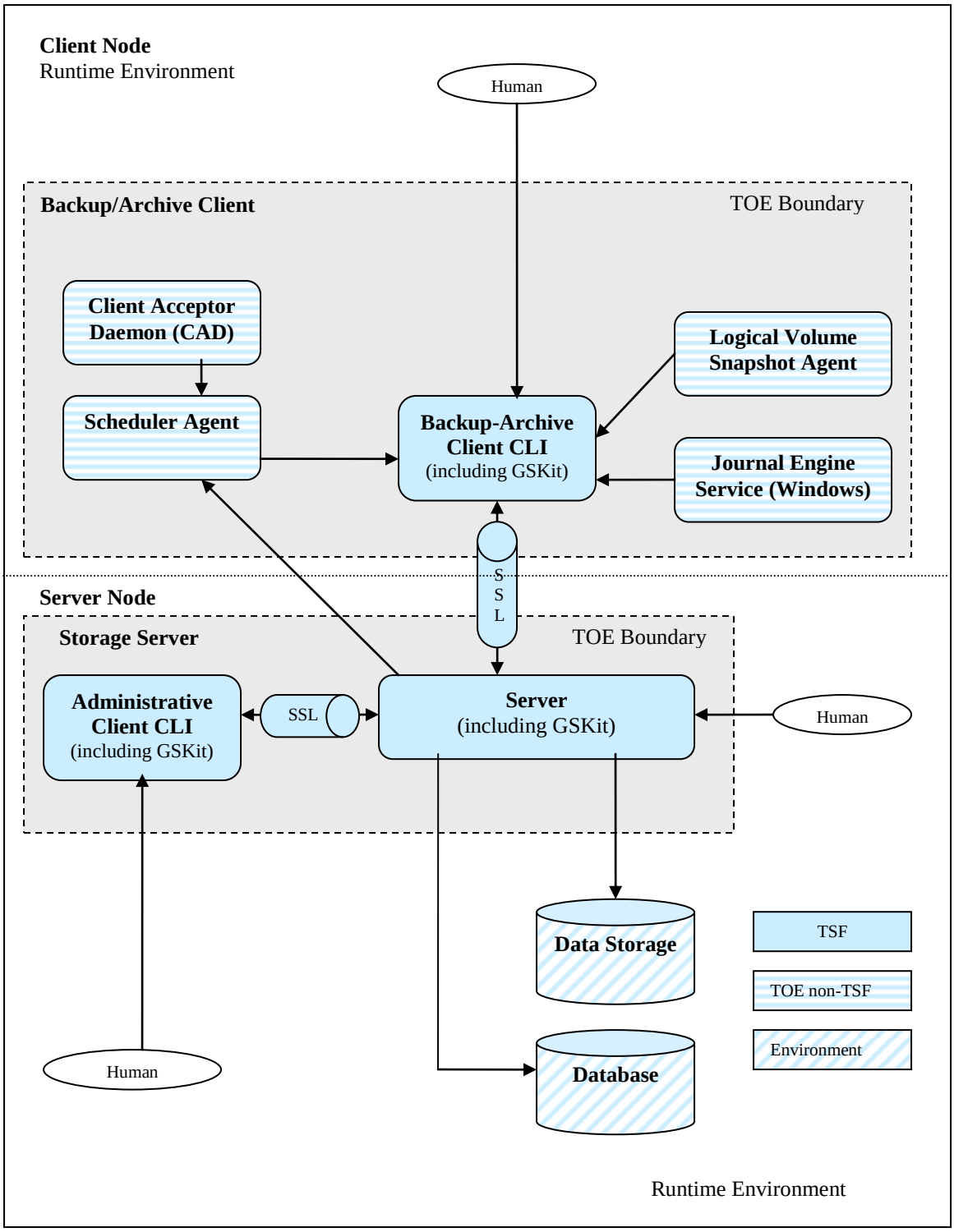

Figure 5 - Schematic TOE Component View and Physical Boundaries.

2.2.1 Physical Boundaries

The TOE is a distributed software product consisting of two main software components (B/A Client and Storage Server) running on the operating systems (which are in the Environment) listed in section 2.6. The software components include applications, daemons/services, data files, and libraries.

The TOE is delivered on CD-ROM and installs on the hard drive of the supported operating systems. The CD-ROM contains the items listed in section 2.6.1 including the guidance documentation which is also part of the TOE. The GSKit library and DB2 database are also included and packaged with the appropriate components. The CD-ROM includes the DB2 Database component which is installed along with TSM automatically. The customer has no separate interaction with the install of DB2. DB2 is part of the Environment.

The TOE also supports a wide variety of physically attached storage devices (disk drives, optical media drives, tape drives, tape library, etc.) collectively called Data Storage which is part of the Environment.

2.2.2 Logical Boundaries

The TOE is a distributed product consisting of two main parts (B/A Client and Storage Server) running on one or more operating systems allowing for multiple Client Nodes to communicate with a single Server. The B/A Client software can reside on the same system as the Server (i.e., a Server Node can also be a Client Node) and/or on other systems on the network.

The TOE components that are security enforcing or relevant are:

- The B/A Client Command Line Interface (CLI) which provides the command line interface of the B/A Client.
- The B/A Client configuration file (*dsm.opt* on Windows, *dsm.sys* on AIX).
- The B/A Client's password data. On AIX, passwords are maintained in the file*TSM.PWD*. On MS Windows, passwords are maintained in a Windows registry key.
- The B/A Client's error log file (*dsmerror.log*) where it writes error messages.
- The B/A Client Scheduler Agent's error log file (*dsmsched.log*) where it writes error messages.
- The keystore files for the B/A Client and Storage Server.
- The Server program which includes the Server Console CLI.
- The Administrative Client CLI which provides a command line interface to manage the Server.
- The IBM Global Security Kit (GSKit 7d for server and GSKit 8.0.13 for client) library package providing TLSv1 support for the Client Node, Administrative Client, and Server and includes the IBM Crypto for C (ICC version 1.4.5) – a FIPS approved cryptographic library containing the following algorithms:
    - AES 128 bit and 256 bit (FIPS approved)
    - TDES 168 bit (FIPS approved)
    - SHA-1 digest algorithm (FIPS approved)
    - MD5
    - random number generator (FIPS approved)
- The Server configuration files (*dsmserv.opt* and *dsmserv.dsk*).

The TOE components that are not security relevant are:

- The B/A Client's Client Acceptor Daemon used to ensure that the Scheduler Agent is always running.
- The B/A Client's Scheduler Agent used to periodically execute the B/A Client CLI for scheduled backups.
- The Logical Volume Snapshot Agent (LVSA) used to create an image backup of online file system.
- The Journal Engine Service (JES) (Windows only) used to detect file modifications in real-time and journal the file names of the modified files for backup at a later time.

Copyright © IBM Corp. 2010-11

The Environment components consist of:

- The operating systems supported by the evaluated configuration.
- An IBM DB2 Database (Version 9.7) used to contain TSF and non-TSF data including user account login data.
- The Data Storage which comprises the devices used to maintain backed up and archived data such as disk drives, tape drives, tape library, optical media drives, etc.

2.3 Summary of TOE Security Functions

This section provides a summary of the security functions provided by the TOE. This functionality is further detailed in the security policies represented by the security functional requirements in chapter 5 and in the description of the TOE Security Functions (TSF) in chapter 6.

2.3.1 Identification & Authentication

The Server maintains its own user account database separate from the operating system's user account database. The TOE defines two account types: client node and administrators. The Server requires all accounts, except for the predefined SERVER_CONSOLE administrator account, to identify and to password authenticate themselves before providing access to the Server. The SERVER_CONSOLE administrator account is only available to administrators who have direct access to the Server program's command line interface (called the TSM Server Console or simply the Server Console) on the Storage Server and requires no identification or authentication. The TOE always treats the identity of the administrator using the TSM Server Console as SERVER_CONSOLE. The quality of passwords used can be enforced through configuration options controlled by the TOE. Password generation is supported on B/A Clients. Accounts must be created on the Server first before they can be used.

The TOE uses TLSv1 communication between the B/A Client and the Server and between the Administration Client and Server requiring the Server to authenticate to these clients using certificate based authentication. Mutual authentication is achieved by combining the TLS certificate authentication with the password-based identification and authentication required by the Server.

2.3.2 Access Control

The backups and archives saved by a Client Node account to a Server are protected by an editable list of rules. By default only the Client Node account can access the data, but a Client Node account can modify its list of rules to grant other Client Node accounts access to one or more backup and archive objects owned by the Client Node account. It can also remove rules from the list thereby denying previously granted access. Each rule includes the name of the client node allowed access, the name of the client node account allowed access, and the name of the object that they are allowed to access, and whether the object name refers to a backup or archive object.

2.3.3 Secure Communications

The TOE uses TLSv1 communications between distributed components of the TOE to protect the data, including TSF data, transferred between these components. This functionality is provided through the use of IBM GSKit. GSKit uses ICC which contains FIPS 140-2 approved cryptographic functionality (the specifics of which are defined in the appropriate security functional requirements in chapter 5). The GSKit also ensures that only secure values are accepted for the attributes governing cryptographic key management and cryptographic operations. The supported cryptographic algorithms are specified in section 7.1.1.5.

2.3.4 Security Management

The management of security critical parameters of the TOE is performed by the administrators of the TOE. The TOE must be used and administered by operating system administrators only. Command line interfaces are provided to manage many of the security features and parameters of the TOE. The TOE also contains configuration files whose access is limited to operating system administrators.

Copyright © IBM Corp. 2010-11

The TOE supports privilege classes and client access authorities for administrative accounts allowing the administrative power of an administrator to be restricted to specific roles. These concepts are defined in section 7.1.5.1.1.

2.4 TOE Security Architecture

The TOE runs on a general purpose multi-user operating systems. In the evaluated configuration, the only individuals allowed to use the TOE are the operating system administrators. All executable files and data files are protected from non-OS administrative access by the use of access control mechanisms of the operating systems in the Environment.

The client/server design provides a natural boundary for scoping administrative tasks. Administrators logged into the B/A Client can only manage the B/A Client and the backup/archive data on the Server associated with that Client Node. They cannot perform Server specific management functions. Administrators logged in via the Administrative Client or Server Console can only perform Server related administrative tasks such as user account management. The Administrative Client CLI and Server Console CLI provide nearly identical command line interfaces. The only difference is that the Server Console does not support the database SELECT command used to perform SQL-like database queries in the Administrative Client CLI. They cannot directly manage a B/A Client.

In order to support a distributed TOE, the architecture provides secure communications using TLSv1 for inter-TOE communication. This not only provides for mutual authentication between the client and server portions of the product, but it provides for data integrity and confidentiality. The Server defines the set of acceptable cipher suites to be used during secure communications, not the Client Nodes. The B/A Client and Administrative Client have keystores which contain the Server's public key (certificate). The Server's keystore contains both its private and public key. The certificates are established when the parts of the TOE are installed. All keystores are protected by the operating system's access control mechanism(s) from unauthorized access.

The Storage Server also controls the creation and maintenance of accounts and the password policies used by all accounts.

The components that require cryptographic support use the same library package, GSKit, to provide these services. The ICC portion of GSKit has been FIPS 140-2 approved, thus, providing assurance of quality cryptographic functionality.

2.4.1 Circumvention Argument

All TOE administrators are considered trusted. Only operating system administrators are allowed to use the TOE (hence, administrators of the TOE must also be administrators of the Environment).

The operating system protects the executables and data components of the TOE from unauthorized access via its access control mechanism. The systems must be in a physically secure environment.

The TOE executes in its own domain within the operating systems specified for this evaluation.

Communication between the clients and Server (including TSF data and user data) is protected from disclosure through the use of TLS. Communication between the TOE and a NAS file server do not utilize TLS.

Users of the TOE must authenticate[1] with the TOE before the TOE will perform any action on their behalf. Each account is uniquely identified by the TOE. Multiple failed login attempts trigger account locking inhibiting the ability of an attacker to break an account's password. Only TOE administrators using the

[1] The SERVER_CONSOLE account does not require authentication but is controlled via physical means according to administrative guidance.

Administrative Client CLI are allowed to create, modify, and delete TOE accounts. Operations on a NAS file server (e.g., backup, restore) are available to authenticated TOE users through the Backup/Archive client.

The TOE protects backups and archives from being accessed by other TOE user accounts, therefore, protecting the user data within the TOE from other accounts. The TOE protects TSF data by allowing only TOE administrators from accessing and modifying TSF data.

2.5 Subjects, Objects, Security Attributes, TSF Data, and User Data

2.5.1 Subjects and Users

The TOE defines two types of user accounts in an evaluated configuration, and consequently, subjects associated with those users:

- Administrator – One who can administer the Server through the Administrative Client and Server Console and one who can administer the B/A Client using the B/A Client.
- Client Node – One who can log into the Server from the B/A Client and perform backup and archive related tasks.

A third account type, server accounts, is supported by TSM. TSM restricts the creation and management of server accounts to administrators. Administrators are instructed, Server accounts are **not** allowed in the evaluated configuration and administrative guidance instructs administrators **not** to create server accounts.

A TOE administrator can have restricted administrative abilities on an administrative account (called privilege classes by the product).

2.5.2 Objects

The TOE has the following objects that users can operate on:

- Backup and archived data objects.

2.5.3 Security Attributes

The follow data items are security attributes:

- user security attributes
  - see FIA_ATD.1 in section 6.1.20
- object security attributes
  - List of access rules used to protect user data. Each rule includes the following data:
    - client node names
    - client node account names
    - object names including pathname
    - object type (backup or archive)
  - Password policy
- Cryptographic attributes
  - Cryptographic algorithm identifiers
  - Cryptographic key sizes
  - Cryptographic key generation methods
  - Cryptographic key distribution methods
  - Cryptographic key destruction methods

2.5.4 TSF and User Data

The follow data items are **TSF data** items:

- configuration parameters located in the configuration files

Copyright © IBM Corp. 2010-11

- certificates stored in keystore files
- account information stored in the Database
- data encryption passwords used by the B/A Client
- the security attributes listed in section 2.5.3

The following items are **user data** items:

- Backup and archived data objects.

2.6 Components and Pre-Requisites – Versions and Packaging

2.6.1 TOE

The evaluated configuration of Tivoli Storage Manager 6.2.1 is delivered on CD-ROM and consists of:

1. IBM TSM Client:
    a. TSM Backup-archive Client – 32-bit
2. TSM Server
    a. TSM Server (including TSM Administrative Client)
3. TSM Publications

The TSM Publications include the TSM guidance documentation for each supported operating system. This includes the following documents:

- TSM for AIX, Administrator's Guide, Version 6.2
- TSM for Windows, Administrator's Guide, Version 6.2
- TSM Common Criteria Guide, Version 6.2
- TSM for AIX, Administrator's Reference, Version 6.2
- TSM for Windows, Administrator's Reference, Version 6.2
- TSM for UNIX and Linux, Backup-Archive Clients Installation and User's Guide, Version 6.2
- TSM for Windows, Backup-Archive Clients Installation and User's Guide, Version 6.2
- TSM Using the Application Programming Interface, Version 6.2
- TSM for AIX, Installation Guide, Version 6.2
- TSM for Windows, Installation Guide, Version 6.2

The packaging includes GSKit 7d for server and GSKit 8.0.13 for client. Only the TOE components for the operating systems mentioned in section 2.6.3 are part of the evaluated configuration.

2.6.2 Other Evaluation Evidence

The following are documents provided as evidence of the assurance mechanisms listed in the preceding table.

- Security Target (this document), Version 2.7, July 29, 2011.
- TSM for Windows and AIX, Configuration Management Plan, Version 6.2.1, Edition 5, December 1, 2010
- IBM Tivoli Storage Manager 6.2.1Life Cycle Support, Version 1.2, March 31, 2010
- IBM Tivoli Storage Manager Delivery Procedures, Revision 9, December 1, 2010.
- High-level Design & Functional Specification IBM Tivoli Storage Manager 6.2.1, Revision 2.0, May 17, 2011
- RFC 2246, The TLS Protocol, Version 1, January 1999

Copyright © IBM Corp. 2010-11

- Common Criteria Test Plan IBM Tivoli Storage Manager 6.2.1, Version 0.9, April 27, 2011
- The following test scripts:
    - ccapi.txt,
    - ccapiaix.tar,
    - ccapiwin.zip,
    - ccsec009.txt,
    - ccsec010.txt
    - Fill.txt,

2.6.3 Software in the Environment

The evaluated configuration (both B/A Client and Storage Server) is supported on the following operating systems:

- Microsoft Windows Server 2003 (32-bit or 64-bit)
- Microsoft Windows Server 2008 (32-bit[2] or 64-bit)
- IBM AIX 6.1 (64-bit)

The evaluated configuration works in a homogeneous OS environment, meaning that the B/A Clients on all supported operating systems work with the Storage Servers on all support operating systems.

The IBM DB2 Database (Version 9.7) used by TSM is part of the environment and is supplied as part of the TSM Server (see section 2.6.1).

The TSM Data Storage is part of the environment and consists of the list of supported media devices supplied with the TOE.

Any NDMP-compliant network attached storage (NAS) file server is an optional part of the environment.

2.6.4 Assurance Level

The evaluation assurance level (EAL) 3 was chosen as a medium level of assurance reflecting the expected assurance requirements of commercial customers using the target of evaluation (TOE) for the backup, archive, and restoration of an organization's data. The TOE is intended to provide a reasonable level of protection for this data comparable to the protection provided by most commercial-off-the-shelf operating system products.

The assurance level EAL3 is applicable in those circumstances where developers or users require a moderate level of independently assured security, and require a thorough investigation of the TOE and its development without substantial re-engineering.

The assurance level EAL3 was augmented with ALC_FLR.1 to address the flaw remediation process used for the product. Since the evaluation methodology for ALC_FLR.1 has been harmonized and is also covered by the Mutual Recognition Arrangement, this was considered to be a useful augmentation for the assurance level chosen.

[2] Only the TSM client runs on 32-bit Windows Server 2008.

3 TOE Security Environment

3.1 Introduction

The statement of TOE security environment describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be deployed.

To this end, the statement of TOE security environment identifies the list of assumptions made on the operational environment (including physical and procedural measures) and the intended method of use of the product, defines the threats that the product is designed to counter, and gives the organizational security policies with which the product is designed to comply.

3.2 Threats

The threats are categorized as those addressed by the TOE and those addressed by the environment.

The **assets** to be protected comprise the information stored, processed, or transmitted by the TOE. The term "information" is used here to refer to all data held or processed within the TOE, including data in transit between TOE components. It also includes TSF and user data stored in the Environment and used by the TOE. It is assumed that an attacker is either an unauthorized user of the TOE, or an authorized user of the TOE who has been granted rights to access the information or resources held by the TOE.

The TOE counters the general threat of unauthorized access to information, where "access" includes disclosure, modification and destruction.

The threat agents are assumed to originate from a well-managed user community in a non-hostile working environment, and hence the product protects against threats of security vulnerabilities that might be exploited in the intended environment for the TOE. The TOE protects against casual breach of TOE security.

3.2.1 Threats Addressed By the TOE

The TOE addresses the threats discussed below.

Table 3-1 - Threats addressed by the TOE

Threat	Description
T.ACCESS	An authorized user of the TOE could gain unauthorized access to resources or information protected by the TOE, or perform operations for which no access rights have been granted, via user error, system error, or other actions.
T.BYPASS	An unauthorized user may bypass TOE security functions to gain access to resources or information protected by the TOE.

3.3 Organizational Security Policies

The TOE complies with the following organizational security policies.

Table 3-2 - Organizational Security Policies

Policy	Description
P.PRIVS	Administrators shall be able to limit their scope of control over administrative tasks.

Copyright © IBM Corp. 2010-11

P.PASSWORD	A policy stating the minimum security requirements for data encryption passwords exists and is enforced outside of the TOE.

3.4 Assumptions

The following conditions are assumed to exist in the TOE operational environment. These assumptions include essential environmental constraints on the secure use of the TOE.

Table 3-3 - Secure Usage Assumptions

Assumption	Description
A.PHYSICAL	The Storage Server portion of the TOE is operated in a physically secure environment. The storage pool devices and media (both online and offline media) are maintained in a secure environment. The TSM Server Console, which provides an administrative CLI to the Server and does not require authentication, shall be operated in a protected environment accessible only to a TSM administrator. The Client Node is protected against unauthorized physical access and modification.
A.ADMIN	The TOE administrators are trustworthy to perform discretionary actions in accordance with security policies and not to interfere with the abstract machine, making sure that the TOE is competently installed and administered. TOE administrators must also be operating system administrators for the client nodes and servers that they administer. Similarly, client node users of the TOE are trustworthy and must also be operating system administrators of the client node.
A.SERVER_RT	The machine providing the runtime environment for all parts of the Storage Server TOE is assumed to be used solely for this purpose and is not used to run other application software except as required for the support of the TOE and for the management and maintenance of the underlying system and hardware. Especially, it's assumed that the underlying systems are configured in a way that prevents unauthorized access to functions provided by or protected by the runtime environment (including network services) either locally or via any network based connections.
A.CLIENT_RT	OS administrators of the client node will only run trusted software on the client node. They will protect their authentication credentials against unauthorized disclosure.
A.OS_SUPPORT	The operating system services that are used by the TOE use reliable time stamps when they determine the current time during the enforcement of inactivity timeouts, enforcement of password management constraints and during backup/restore operations.

Copyright © IBM Corp. 2010-11

4 Security Objectives

4.1 Security Objectives for the TOE

The following list contains the security objectives that the TOE meets.

Table 4-1 - TOE Security Objectives

Objective	Description
O.AUTHENTICATE	The TOE must ensure that all users authenticating to the Server and not using the TSM Server Console are identified and authenticated before being granted access to the TOE mediated resources. Also, the B/A Client and Administrative Client must authenticate the Server before providing identification and authentication information to the Server. The TOE must also terminate inactive sessions to force a re-authentication by the authorized user. Note: The TSM Server Console requires protection by the Environment since the Server Console provides an administrative command-line interface without authentication.
O.PRIVS	The TOE must provide a mechanism to limit the scope of control of a given administrator, so that TOE management responsibilities can be separated and/or limited.
O.NOBYPASS	The TOE security policy enforcement functions must be invoked and succeed before allowing access to TOE protected objects and services.
O.USERDATA	The TOE must protect backup, archive, and restore data from disclosure.
O.ADMDATA	The TOE must protect TSF data from disclosure.
O.SECURE_DEFAULTS	The TOE must ensure that administrators can provide only secure values for cryptographic parameters during their management of the security of the TOE.

4.2 Security Objectives for the Environment

The following list contains the security objectives for the environment.

Table 4-2 - Environment Security Objectives

Objective	Description
OE.ADMIN	Those responsible for the administration of the TOE must be trained such that they are capable of installing and managing the TOE and the security of the information it contains. Administrators are trustworthy. Client node users of the TOE are trustworthy. They will follow the user guidance provided as part of the TOE.
OE.PHYSICAL	Those responsible for the Storage Server portion of the TOE must ensure that the Storage Server is operated in a physically secure environment. The storage pool devices and media (both online and offline media) are maintained in a secure environment. The TSM Server Console, which provides an administrative CLI to the Server and does not require

Copyright © IBM Corp. 2010-11

Objective	Description
	authentication, shall be operated in a protected environment accessible only to a TSM administrator. Those responsible for the Client Node portion of the TOE must ensure that the Client Node is protected against unauthorized access and modification.
OE.SERVER_RT	Those responsible for the operation of the TOE must ensure that the systems hosting the Storage Server part of the TOE are used solely for this purpose and configured in a way that prevents unauthorized access. This includes preventive measures to ensure that all systems hosting parts of the TOE are protected against unauthorized physical access and network-based attacks by blocking network traffic that is not required for system operation.
OE.CLIENT_RT	OS administrators of a client node will only run trusted software on these systems. They will protect their OS authentication credentials against unauthorized disclosure.
OE.PASSWORD	Those responsiblefor the operation of the TOE must ensure that the minimum security requirements for data encryption passwords exist and are enforced procedurally.
OE.OS_SUPPORT	The operating system services that are used by the TOE use reliable time stamps when they determine the current time during the enforcement of inactivity timeouts, enforcement of password management constraints and during backup/restore operations.

Copyright © IBM Corp. 2010-11

5 Extended security requirements

5.1 Identification of Extended security requirements

This section is used for identifying the extended security requirements.

Extended requirements use a "_EXT" suffix attached to the acronym for the CC requirement upon which the explicit requirement is patterned.

The following Security Functional Requirements are explicitly stated for the TOE:

1) FDP_RIP_EXT.1 Subset residual information protection,
2) FDP_RIP_EXT.2 Full residual information protection,
3) FDP_ACF_ EXT.1 Security attribute based access control, and

The following Security Assurance Requirements are explicitly stated for the TOE: none.

5.1.1 FDP_RIP_EXT.1 and FDP_RIP_EXT.2

The FDP Class is extended with requirements which allow the author of a requirement to enumerate the sensitive data being affected by the residual protection.

FDP_RIP_EXT.1 Subset residual information protection

Hierarchical to:	No other components.
FDP_RIP_EXT.1.1	The TSF shall ensure that [assignment: explicitly stated sensitive data] of [assignment: explicitly stated resource] is made unavailable upon the [selection: allocation, deallocation] of [assignment: an explicitly stated resource] to/from the following objects: [assignment: list of objects].
Dependencies:	No dependencies

FDP_RIP_EXT.2 Full residual information protection

Hierarchical to:	FDP_RIP_EXT.1
FDP_RIP_EXT.2.1	The TSF shall ensure that [assignment: explicitly stated sensitive data] of [assignment: explicitly stated resource] is made unavailable upon the [selection: allocation, deallocation] of [assignment: explicitly stated resource] to/from all objects.
Dependencies:	No dependencies

5.1.2 FDP_ACF_EXT.1

The FDP Class is extended with a requirement which allows the ST author to model access to data contained within an object rather than access to the object itself.

FDP_ACF_EXT.1 Security attribute based access control

Hierarchical to:	No other components.
FDP_ACF_EXT.1.1	The TSF shall enforce the [assignment: access control SFP] to object data based on the following:[assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes].
FDP_ACF_EXT.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: rules

Copyright © IBM Corp. 2010-11

	governing access among controlled subjects and controlled objects using controlled operations on controlled objects].
FDP_ACF_EXT.1.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects].
FDP_ACF_EXT.1.4	The TSF shall explicitly deny access of subjects to objects based on the [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation

5.2 Justification of the Extended Security Requirements

As there are no extended Security Assurance Requirements for the TOE, the subsequent analysis only focuses on the Security Functional Requirements for the TOE.

FDP_RIP_EXT.2 is modeled after the extended FDP_RIP.2. The extended FDP_RIP.2 was not used because the intent was to focus the requirement upon specific attributes and resources, which would have been an invalid refinement of these requirements.

FDP_RIP_EXT.2 is quantifiable and compliance can be determined by examining each relevant operation on the stated resources to ensure that the necessary controls are implemented in each allocation and/or deallocation of the resource.

FDP_ACF_EXT.1 models the control of access to data within the specified objects that have used encryption as the mechanism to control user access. The base SFR FDP_ACF.1 models access to objects rather than data contained within the objects. Therefore, an explicit SFR has been created.
While modification of SFR FDP_ACF.1 to create this explicit SFR is minor it cannot be considered a refinement.

The explicit SFR is measurable and compliance or noncompliance can be readily determined. Additionally, as the requirement does not differ significantly from the base SFR the statement of requirement can be considered clear and unambiguous. The dependencies to FDP_ACC.1 and FMT_MSA.3 have also been retained.

Copyright © IBM Corp. 2010-11

6 Security Requirements

When iterating security functional requirements (SFRs), the convention of a hyphen followed by 3 lower case letters is used to show linkage between coupled SFRs. All other operations have been marked with bold face font. Explicit requirements use a "_EXT" suffix attached to the acronym for the CC requirement upon which the explicit requirement is patterned.

6.1 TOE Security Functional Requirements

6.1.1 FCS_CKM.1-sym Cryptographic key generation (TLS: Symmetric Algorithms)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **as defined in the TLS v1.0 standard** and specified cryptographic key sizes **256 bit (AES)** that meet the following:

- **TLS 1.0 standard.**

Application Note: Generation of symmetric keys is defined in section 8 in the TLS v1.0 standard. The library used by the TOE also supports SSLv2 and SSLv3, but these are not part of the evaluated configuration.

6.1.2 FCS_CKM.1-rsa Cryptographic key generation (TLS: RSA)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **RSA** and specified cryptographic key sizes **1024 bit** that meet the following: **conformant to RFC 2313 [RFC2313] and FIPS 140-2 [FIPS140-2] approved.**

Application Note: GSKit is used to generate the RSA keys. GSKit contains and uses the ICC library to generate the actual keys. ICC has been FIPS approved for generating RSA keys.

6.1.3 FCS_CKM.2-sym Cryptographic Key Distribution (TLS: Symmetric Keys)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **Transport Layer Security handshake using RSA encrypted exchange of session keys** that meets the following:

- **TLS Version 1.0 [RFC2246]**

Application Note: This requirement addresses the exchange of TLS session keys as part of the TLS handshake protocols.

6.1.4 FCS_CKM.2-rsa Cryptographic Key Distribution (TLS: RSA Public Keys)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **digital certificates for public RSA keys** that meets the following:

- **certificate format as defined in the standard X.509 Version 3.**

Application Note: This requirement addresses the exchange of public RSA keys as part of the TLS client and server authentication.

Copyright © IBM Corp. 2010-11

6.1.5 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [as specified by FDP_RIP_EXT.2-enc] that meets the following: [none].

6.1.6 FCS_COP.1-sym Cryptographic operation (TLS: Symmetric Operations)

FCS_COP.1.1 The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **AES** and cryptographic key sizes **256 bit (AES)** that meet the following:

- **TLS Version 1.0 and the following cipher suites as defined in [RFC3268]:**
  - **TLS_RSA_WITH_AES_256_CBC_SHA**

Application Note: This requirement applies to IBM Global Security Kit (GSKit).

6.1.7 FCS_COP.1-rsa Cryptographic operation (TLS: RSA)

FCS_COP.1.1 The TSF shall perform **digital signature generation and digital signature verification** in accordance with a specified cryptographic algorithm **RSA** and cryptographic key sizes **1024 bit** that meet the following:

- **TLS Version 1.0 [RFC2246]**

Application Note: This requirement addresses the RSA digital signature generation and verification operations using the RSA algorithm as required by the TLS session establishment protocols (provided a cipher suite including RSA is used). Note that the details of the signature format such as the use of the PKCS#1 block type 1 and block type 2 are defined in the TLS Version 1.0 standard.

6.1.8 FCS_COP.1-enc Cryptographic operation

FCS_COP.1.1 The TSF shall perform **symmetric encryption and symmetric decryption** in accordance with a specified cryptographic algorithm **AES (CBC mode)** and cryptographic key sizes **128-bits** that meet the following:

- **conformant to FIPS 197 [FIPS197] (AES, CBC mode) and FIPS 140-2 [FIPS140-2] approved.**

Application Note: This SFR applies to the data encryption feature of the TOE that allows the B/A Client to encrypt the backups and archives locally prior to sending them to the Server and then decrypt them when retrieved from the Server. The AES-128 bit functions are from the IBM Crypto for C (ICC) library in the FIPS Approved mode. The keys used for this encryption are the data encryption passwords found on the B/A Client.

6.1.9 FDP_ACC.1-obj Subset access control (stored objects)

FDP_ACC.1.1 The TSF shall enforce the **Stored Object Access Control SFP** on **users as subjects, backups and archives as objects, and all operations among subjects and objects covered by the Stored Object Access Control SFP.**

Copyright © IBM Corp. 2010-11

6.1.10 FDP_ACC.1-prv Subset access control (privilege class)

FDP_ACC.1.1 The TSF shall enforce the **Privilege Class Management SFP** on **administrators as subjects, administrator accounts as objects, and all operations on privilege classes and client access authorities of administrator accounts covered by the Privilege Class Management SFP.**

6.1.11 FDP_ACC.1-enc Subset access control (data encryption)

FDP_ACC.1.1-enc The TSF shall enforce the **[User Data Encryption SFP]** on **[**

a) **users (subject),**
b) **backup and archive files (objects), and**
c) **successful decryption and access to data contained within backup and archive files (operation)].**

6.1.12 FDP_ACF_EXT.1-enc Security attribute based access control

FDP_ACF_EXT.1.1-enc The TSF shall enforce the [User Data Encryption SFP] to object data based on the following: [

a) **users:**
    i. **user encryption password**
b) **backup and archive files:**
    i. **data encryption key].**

FDP_ACF_EXT.1.2-enc The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled object data is allowed: [

a) **the user must have the user encryption password that corresponds to the data encryption key to successfully decrypt and access data contained within backup and archive files].**

FDP_ACF_EXT.1.3-enc The TSF shall explicitly authorise access of subjects to object data based on the following additional rules: [none].

FDP_ACF_EXT.1.4-enc The TSF shall explicitly deny access of subjects to object data based on the following additional rules: **[none].**

6.1.13 FDP_ACF.1-obj Security attribute based access control

FDP_ACF.1.1-obj The TSF shall enforce the **Stored Object Access Control SFP** to objects based on the following:

- **the users as subjects,**
- **the backup and archive objects as objects,**
- **the user data encryption password (only applicable when requesting access to encrypted backup and archive objects) as a security attribute, and**
- **the client node name, the client node account name, the object name, and the object type (i.e., backup or archive) as access control attributes associated with each access control rule in the list of access control rules associated with the object.**

FDP_ACF.1.2-obj The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- **user *A* can query and restore a backup object from user *B*'s backup if user *A*'s name, user *A*'s associated client node name, and user *B*'s backup object's name are specified in the list of access rules associated with user B's backup objects; otherwise, the query and restore are denied.**
- **user *A* can query and retrieve an archive object from user *B*'s archive if user *A*'s name, user *A*'s associated client node name, and user *B*'s archive object's name are specified in the list**

of access rules associated with user B's archive objects; otherwise, the query and retrieve are denied.

FDP_ACF.1.3-obj The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

- **a client node user can perform all operations on all objects owned by the user on the originating client node.**
- **an administrator with system privilege class can perform all operations on the backup and archive objects associated with all users to any client node associated with the Server.**
- **an administrator with policy privilege class can perform all operations on the backup and archive objects associated with all users assigned to the policy domains associated with this administrator on any client node within these policy domains.**
- **an administrator with node privilege class and client owner authority can perform all operations on the backup and archive objects associated with the client nodes specified with the privilege defined for that administrator on any of the specified client node.**
- **an administrator with node privilege class and client access authority can perform all operations on the backup and archive objects associated with the client nodes specified with the privilege for that administrator only on the originating client node.**
- **an administrator with node privilege class and client owner authority can perform all operations on the backup and archive objects associated with the client nodes of the domain policies specified with the privilege defined for that administrator on any of the client nodes within the policy domain.**
- **an administrator with node privilege class and client access authority can perform all operations on the backup and archive objects associated with the client nodes of the domain policies specified with the privilege for that administrator only on the originating client node.**

FDP_ACF.1.4-obj The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

- **none.**

Application Note: These operations only exist on the B/A Client. In a list of rules, wildcard characters can be used in client node names and object names to allow matching of multiple client nodes and object names, respectively, by a single rule.

6.1.14 FDP_ACF.1-prv Security attribute based access control

FDP_ACF.1.1-prv The TSF shall enforce the **Privilege Class Management SFP** to objects based on the following:

- **administrators as subjects, administrator accounts as objects, and the following attributes of an administrator account:**
  - **privilege classes:**
    - **System**
    - **Policy**
    - **Storage**
    - **Operator**
    - **Node**
    - **Analyst**
    - **(none – the privilege classes value can be empty)**
  - **client access authorities:**
    - **Client owner**
    - **Client access**

FDP_ACF.1.2-prv The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- **an administrator must have the system privilege class to grant and revoke these attributes; otherwise, grant and revoke are denied.**

FDP_ACF.1.3-prv The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

- **none.**

FDP_ACF.1.4-prv The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

- **none.**

6.1.15 FDP_ITT.1 Basic internal transfer protection

FDP_ITT.1.1 The TSF shall enforce the **Stored Object Access Control SFP** to prevent the **disclosure** of user data when it is transmitted between physically-separated parts of the TOE.

Application Note: This requirement applies to the TLS connections.

6.1.16 FDP_RIP_EXT.2-rnd Full residual information protection

FDP_RIP_EXT.2.1-rnd The TSF shall ensure that ***the state information*** of ***the random number generator*** is made unavailable upon the **deallocation of the *random number generator* from** all objects.

Application note: The refinement states that the resource this SFR applies to is the random number generator. Since the random number generator's state is considered security critical, it will be cleared when it is removed from memory.

Application Note 2: This requirement applies to IBM Global Security Kit (GSKit).

6.1.17 FDP_RIP_EXT.2-enc Full residual information protection

FDP_RIP_EXT.2.1-enc The TSF shall ensure that any previous ***sensitive*** information content of ***the buffer space memory*** is made unavailable upon the **deallocation of the *buffer space memory* from** all objects.

Application Note: The refinement states that the resources this SFR applies to are sensitive information in the buffer space memory. The TOE considers all cleartext critical security parameters, random numbers and cryptographic keys as such sensitive information.

Application Note 2: This requirement applies to IBM Global Security Kit (GSKit).

6.1.18 FIA_AFL.1-nda Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when **3** unsuccessful authentication attempts occur related to **consecutive authentication attempts of the same client node account**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **met or surpassed**, the TSF shall **prohibit further login of that client node account until one of the following TSM administrator types unlocks the user's account:**

- **administrator with unrestricted policy privilege class**
- **administrator with restricted policy privilege class for that client node**
- **administrator with system privilege class**

Copyright © IBM Corp. 2010-11

6.1.19 FIA_AFL.1-ada Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when **3** unsuccessful authentication attempts occur related to **consecutive authentication attempts of the same administrative account**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **met or surpassed**, the TSF shall **prohibit further login of that administrative account until a TSM administrator with system privilege unlocks the account**.

6.1.20 FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

- **Account name – Specifies the user's account name.**
- **Account password – Specifies the user's account password.**
- **Password changed time – Specifies the time that the user's account password was last changed.**
- **Password expiration period – Specifies the password expiration period for the account.**
- **Account locked – Specifies whether the user's account is locked or unlocked.**
- **Consecutive failed logins – Specifies the current number of failed logins associated with the user's account.**
- **Account type – Specifies if the account is a client node account or administrative account.**
- **Privilege classes – Specifies the list of privilege classes of an administrator account, if any, and the restrictions applicable to the privileges, if any. (administrator accounts only)**
- **Client access authority – Specifies the client access authority of certain privilege classes. (administrator accounts only)**

Application Note: A user is defined as a TSM account.

6.1.21 FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets **for administrator and client node account authentication information** meet **the following metric:**

- **The characters are chosen from a set of 41 characters of the 7-bit ASCII character set consisting of 26 alpha characters (A-Z case insensitive), 10 numeric characters (0-9), underscore (_), period (.), hyphen (-), plus (+), and ampersand (&)**
- **Minimum length of 8 characters**
- **Maximum age of 90 days**

Application Note: TSM contains a server-wide default password expiration period and a per-account password expiration period. The per-account values have precedence over the default value. An administrator with system privilege can modify the default password expiration period and can individually add, modify, and remove per-account password expiration periods. The SERVER_CONSOLE administrator account ID never has a password.

6.1.22 FIA_SOS.2 Generation of secrets

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet **the following metric:**

- **The characters are chosen from a set of 41 characters of the 7-bit ASCII character set consisting of 26 alpha characters (A-Z case insensitive), 10 numeric characters (0-9), underscore (_), period (.), hyphen (-), plus (+), and ampersand (&)**

Copyright © IBM Corp. 2010-11

- **The password length is 24 characters.**

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for **none of the TSF functions**.

Application Note: The TOE does not enforce the use of TOE generated passwords. The generation function is provided to the client node user as an option and is configurable in the client node configuration file.

6.1.23 FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow [any action at a TSM server console] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note: The server console interface does not enforce authentication. It requires restricted access. The TSM account ID associated with the TSM server console is always SERVER_CONSOLE. The SERVER_CONSOLE account ID never has a password. Administrators are instructed by guidance documents to utilize physical means to control access to the server console in an evaluated configuration.

6.1.24 FIA_UID.1 Timing of identification

FIA_UID.1.1 The TSF shall allow [any action at a TSM server console] on behalf of that user performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note: A user is defined as a TSM account. The TSM server console does not require user identification. The user at the TSM server console is always considered to be the SERVER_CONSOLE user.

6.1.25 FIA_USB.1 User-subject binding

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on behalf of that user:

- **Account name**
- **Account type**
- **Privilege classes (administrator account types only)**
- **Client access authority (administrator account types only)**

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

- **For user's not using the Server Console, upon successful identification and authentication, the account name, account type, privilege classes, and client access authority shall be those specified in the account entry for that user.**
- **For user's using the Server Console, the account name will be SERVER_CONSOLE, the account type will be Administrator, and the privilege authorizations and client access authority shall be those specified in the account entry for the SERVER_CONSOLE account.**

FIA_USB.1.3 The TSF enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

- **none**

6.1.26 FMT_MSA.1-obj Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **Stored Object Access Control SFP** to restrict the ability to **modify** the security attributes:

- **associated with the backup and archive objects owned by a client node user**

to **the client node user and authorized administrators with client node access to the client node that owns the backup and archive objects.**

6.1.27 FMT_MSA.1-enc Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **User Data Encryption SFP** to restrict the ability to **modify** the security attributes:

- **associated with the encryption of backup and archive objects owned by a client node user**

to **the client node user and authorized administrators with client node access to the client node that owns the backup and archive objects.**

6.1.28 FMT_MSA.2 Secure Security Attributes (Crypto Attributes)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for [**the following cryptographic security attributes: key sizes, algorithms, key generation methods, key destruction methods and key distribution methods**].

6.1.29 FMT_MSA.3-obj Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **Stored Object Access Control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **client node user** to specify alternative initial values to override the default values when an object or information is created.

6.1.30 FMT_MSA.3-prv Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **Privilege Class Management SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **an administrator with system privilege class** to specify alternative initial values to override the default values when an object or information is created.

6.1.31 FMT_MSA.3-enc Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **User Data Encryption SFP**to provide **permissive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **an administrator with system privilege class** to specify alternative initial values to override the default values when an object or information is created.

Copyright © IBM Corp. 2010-11

6.1.32 FMT_MTD.1-acm Management of TSF data (Account management)

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize and modify** the **account attributes** to **administrators with system privilege class**.

Application Note: Administrators with system privilege class can manage all account types.

6.1.33 FMT_MTD.1-ccm Management of TSF data (Client configuration management)

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize and modify** the **client configuration** to **OS administrators of the client node.**

6.1.34 FMT_MTD.1-ipm Management of TSF data (Initial password management)

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize** the **account passwords** to **administrators with system privilege class.**

6.1.35 FMT_MTD.1-cpm Management of TSF data (Continuous password management)

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **account passwords** to

- **client node accounts for their own passwords**
- **administrator accounts for their own passwords**
- **administrators with system privilege class for modifying any account's password**
- **administrators with policy privilege class for client node accounts that belong to the policy domains associated with the administrator.**

6.1.36 FMT_MTD.1-ppm Management of TSF data (Password Policy management)

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **password policy parameters** to **administrators with system privilege class**.

6.1.37 FMT_MTD.1-scm Management of TSF data (Server configuration management)

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize and modify** the **server configuration** to **OS administrators of the server**.

6.1.38 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions:

- **Account management**
- **Client configuration management**
- **Password management (Administrator and Client Node)**
- **Password policy management**
- **Server configuration management**
- **Stored object access control management**

Copyright © IBM Corp. 2010-11

- **Privilege class management**

6.1.39 FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles **administrator and client node**.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

6.1.40 FPT_ITT.1 Basic internal TSF data transfer protection

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

Application Note: The TOE uses TLSv1 to protect data transmitted between separate parts of the TOE

6.1.41 FTA_SSL.3 TSF-initiated termination

FTA_SSL.3.1 The TSF shall terminate an interactive session after a **15 minute interval of user inactivity**.

Application Note: This applies to the client node to server communications and the administrative client to server communications only.

Copyright © IBM Corp. 2010-11

7 TOE Summary Specification

The TOE summary specification shall define the instantiation of the security requirements for the TOE. This specification shall provide a description of the security functions and assurance measures of the TOE that meet the TOE security requirements. This is a very important chapter as it describes the specific security functions and assurance measures of the TOE.

7.1 TOE Security Functions

7.1.1 Identification & Authentication (IA)

7.1.1.1 Authentication (IA.1)

The Server requires all users to identify and authenticate themselves before providing access (with the exception of the SERVER_CONSOLE account described below). The TOE uses its authentication mechanism as one of the ways to control access to TSF data and user data. Users use textual names to associate themselves with account names. The account names must be known to the Server in order for the Server to provide access. Each account type (administrator and client node) has its own separate name space for identifiers. Therefore, it is possible to have an administrator account with the same textual name as a client node account.

Passwords are used for authentication. The password policy is described later in this chapter.

The SERVER_CONSOLE administrator account is never authenticated and never has a password. The SERVER_CONSOLE administrator account is only available through the TSM Server Console interface. It is also the only account available for use on the Server Console.

7.1.1.2 Account Management (IA.2)

The Server supports both open and closed registration. The administrator must register client nodes when registration is set to closed. Open registration allows the client nodes to register their node names, passwords, and compression options. For the evaluated configuration, the administrator must have complete control over the creation of accounts; therefore, only closed registration is permitted.

Administrators can perform the following functions:

- create accounts for all account types
- remove accounts for all account types,
- lock out accounts for administrator and client node account types, and
- unlock accounts for administrator and client node account types.

An account that is “locked out” is prevented from logging into the system.

An administrator must have system privilege to perform any of these functions on an administrator account. To perform these functions on a client node account, an administrator must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node belongs.

The SERVER_CONSOLE administrator account can never be removed from the system or locked.

Copyright © IBM Corp. 2010-11

7.1.1.3 Consecutive Authentication/Password Failures (IA.3)

Once an administrator account or client node account reaches 3 consecutive failed logon attempts, the account is locked and can no longer be logged into until it's unlocked by an administrator. Only an administrator with system privilege can unlock an administrator account (implemented by IA.2). To unlock a client node account, an administrator must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node belongs.

The SERVER_CONSOLE account never has a password and can never be locked out.

7.1.1.4 Inactivity Timeout (IA.4)

The Server supports a per-session inactivity timeout for network sessions established with the Server for client node and administrative client sessions. The inactivity timeout is called an idle session timeout. If a session has been idle for 15 minutes, the network connection is closed by the Server and the session terminated.

7.1.1.5 Network Authentication (IA.5)

The B/A Client CLI and Administrative Client CLI connect to the Server using TLSv1. All three entities have their own keystores which are protected by access control mechanisms of the environment. The two CLIs have the public key (certificate) of the Server in their keystores. The Server's keystore contains both the Server's public and private keys. The Server's public key is stored in a file called cert.kdb. The TOE does not provide a mechanism to delete the Server's keys, merely to create a new key pair which overwrites the previous key pair with the new key data

The CLIs use the TLSv1 protocol, which in turn uses the Server's public key located in each CLI's keystore, to verify that they are communicating to the proper Server. Once the TLS session is established, the Server requires the CLIs to login using the mechanism described in **IA.1**. The entire session is protected using TLS until the connection is terminated. (Detailed descriptions of how the protocols work can be found in [RFC2246 and RFC3268] for TLSv1.) Cryptographic session keys for the TLS session are protected by the TOE against unauthorized access and are destroyed by the object re-use functions of the TOE. Long living private keys of a public/private key pair will also be destroyed by the object re-use function of the TOE when they are kept in memory.

The TOE uses the GSKit library to implement TLSv1. ICC, contained in GSKit, is FIPS 140-2 approved. The GSKit library protects sensitive data (e.g., keys) between different sessions. Unprotected (i.e. cleartext) critical security parameters are considered such sensitive data and are cleared before a buffer is released. The mechanism for this is that the TSF marks all critical data objects. Before releasing the buffers, the TSF clears all the objects that are marked as critical. The GSKit library is also responsible for ensuring that secure default values are used when new keys are created or destroyed. The B/A Client clears (i.e., writes zeros to) any memory buffer that is used by the B/A client to store cryptographic keys at the completion of each backup or restore operation.

The following cryptographic algorithm configurations are support by the TOE as defined by TLSv1:

- TLS_RSA_WITH_AES_256_CBC_SHA

7.1.1.6 User Attributes (IA.6)

The TSF data associated with each account is as follows:

- Account name – Specifies an account name.

Copyright © IBM Corp. 2010-11

- Account password – Specifies an account password. (Not applicable to the SERVER_CONSOLE account.)
- Password changed time – Specifies the time that the password was last changed on an account. (Applies to client node and administrator accounts only. Not applicable to the SERVER_CONSOLE account.)
- Password expiration period – Specifies the password expiration period for the account. (Not applicable to the SERVER_CONSOLE account.)
- Account locked – Specifies whether the account is locked or unlocked. (Applies to client node and administrator accounts only. Not applicable to the SERVER_CONSOLE account.)
- Consecutive failed logins – Specifies the current number of consecutive failed logins associated with the account. (Applies to client node and administrator accounts only. Not applicable to the SERVER_CONSOLE account.)
- Account type – Specifies if the account is a client node or administrative account.
- Privilege classes – Specifies the privilege classes of an administrator account, if any, and the restrictions applicable to "restricted" privileges.
- Client access authority – Specifies the client access authority of certain administrator types.

7.1.2 Access Control (ACCESS)

7.1.2.1 Object Access (ACCESS.1)

When the client node backs up or archives objects from a file system, it copies the objects and the objects' OS security attributes (such as the object's owner, owner group, and access permissions) from the file system and sends them to the Server. The operating system limits the access of a B/A Client to those operating system objects that are allowed by the process credentials associated with the B/A Client CLI process.

Once stored on the Server, a client node account can control which other client node accounts can access the backups/archives owned by the account through the use of the B/A Client CLI's **set access** and **delete access** commands. The commands add or remove, respectively, rules from the list of rules maintained for the objects stored by the account. When a client node account is created, the list of rules is empty.

Rules are always permissive, never restrictive. The owning account can always access the objects owned by the account. By default, all other client node accounts are denied access unless a rule permits access. The list of rules is processed until a matching permissive rule is found for the requesting account and requested object(s). If no matching rule is found, access is denied. Each rule consists of the object's collection type (i.e., backup or archive), the client node name, the client node account name, and the object's name including the pathname. If a client node account name is not specified, all accounts associated with the specified client node are allowed access to the object(s). The types of access permitted by a rule are retrieval (of archives), restoration (of backups), and query of archives/backups only. Thus, other client node accounts cannot add or delete objects owned by another client node nor can they modify the list of rules of another client node account.

The client node name and pathname can contain wildcard characters which are saved as part of the rule. Supported wildcard characters are:

- * - Asterisk. Matches zero or more characters.
- ? - Question mark. Matches any single character at the present position.

Backups for NAS file servers are initiated using the Backup/Archive client from a user that has at least client owner authority over the NAS file servers. Administrators register a NAS file server as a node and can schedule an NDMP operation. Under the direction of the TOE, the NAS file server performs backup and restore of its volumes to a tape library that is either connected directly to the NAS file server or is provided by the TOE. The TOE coordinates the NAS file server backup or restore operation.

Copyright © IBM Corp. 2010-11

7.1.3 Data Protection (DATA_PROT)

7.1.3.1 Backup Encryption (DATA_PROT.1)

The client node can encrypt the backup and archive objects prior to sending them to the Server using AES-128 bit encryption. The password used to encrypt the data is known to the B/A Client, but not known to the Server. The password is used as the key when encrypting and decrypting the objects.

When creating a backup or archive object, the B/A client configuration information or user commands indicate whether backups are encrypted prior to their being sent to the Server. The B/A Client CLI accepts a password from the user for encrypting the objects. The password can be entered by the user through the B/A Client CLI or, in the case of scheduled backups/archives, the password is obtained from the access protected TSM.PWD password file on AIX or from the access protected registry entry on Windows.

7.1.4 Password Management (PM)

7.1.4.1 Password Security Policy (PM.1)

An administrator with system privilege can specify on the Server the values for the password security policy. The TOE supports setting the values both globally as well as per-account. The per-account values take precedence over the global values. For the evaluated configuration, the password security policy settings are:

- Minimum length of a password: 8 characters.
- Maximum lifetime of a password: 90 days.
- Maximum number of consecutive failed logon attempts: 3 attempts.

The character set available for passwords consists of the following 41 characters from the 7-bit ASCII character set:

- 26 alpha characters A-Z ignoring case sensitivity
- 10 numeric characters 0-9
- 5 special characters: underscore (_), period (.), hyphen (-), plus (+), ampersand (&)

Note that the SERVER_CONSOLE account never has a password and, therefore, is not subject to the password security policy.

7.1.4.2 Password Creation/Modification/Generation (PM.2)

An administrator and client node account type can change its own password after successfully logging onto the Server. An administrator with system privilege can

- create accounts;
- initialize passwords on newly created accounts; and
- change another account's password (be it another administrator or client node account's password) after the administrator has successfully logged onto the Server.

Passwords can be created at account creation time. Also, the TOE is capable of generating both authentication passwords and data encryption passwords for client node accounts via the B/A Client CLI. The password generator uses the FIPS 186-2 approved RNG in the IBM ICC library for generating passwords and generates 128-bit random passwords. Each client node stores the generated passwords locally for its client node account in the access protected TSM.PWD password file on AIX or in the access protected registry entry on Windows.

Copyright © IBM Corp. 2010-11

The SERVER_CONSOLE account is the exception. This account never has a password and, therefore, can never modify its password.

7.1.5 System Management (SM)

7.1.5.1 Roles (SM.1)

The TOE supports two roles (same as the 2 account types): administrator and client node. Roles and account types are used to control access to TSF data and user data once a user is logged into the TOE.

7.1.5.1.1 Administrators

Administrators monitor, manage, and maintain the TOE including managing client nodes and Storage Servers. The capabilities of administrators are controlled through administrative privilege classes which must be explicitly assigned to an administrative account.. Each administrator can have zero or more privilege classes assigned to them. Privilege classes influence the access control mechanisms of multiple object types within the TOE including TSF data (e.g. password management), backup and archive objects, and even the management of the privilege classes themselves. All administrators, regardless of their associated privilege classes, are trusted. The privilege classes are:

- **System** – Dominates all other privilege classes. An administrator with this privilege can manage all aspects of the TOE including all administrators associated with the Storage Server, all client nodes associated with the Storage Server, and the Storage Server.
- **Policy** -
    - **Unrestricted Policy** – Manage the backup and archive services for nodes assigned to any policy domain, including managing nodes, policies, and schedules.
    - **Restricted Policy** – Same capabilities as the Unrestricted Policy privilege except their authority is limited to specific policy domains.
- **Storage –**
    - **Unrestricted Storage** – Manage server storage, but not definition or deletion of storage pools, including managing the database and recovery log, managing TSM devices, managing, TSM storage.
    - **Restricted Storage** – Same as the Unrestricted Storage privilege except their authority is limited to specific storage pools and they cannot manage the database and recovery log.
- **Operator** – Control the immediate operation of the server and the availability of storage media, including managing client sessions and managing tape and tape library operations.
- **Node** – Access a backup-archive client to perform backup, archive, and restore operations.
- **Analyst** – Reset the counters that track server statistics.

Administrators with no privilege classes can only display information about administrators and update information about themselves, such as changing their own password.

The terms “unrestricted” and “restricted” associated with the policy and storage privilege classes are used to categorize the scope of these privileges, but are not actual privileges defined by TSM.

The figure below illustrates the privilege class hierarchy where the system privilege class has the highest privilege.

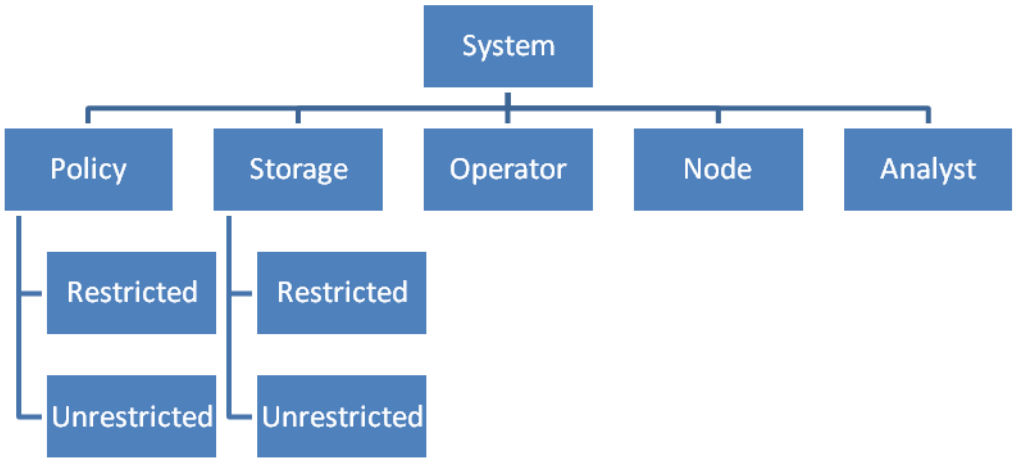

Figure 6 - Privilege class hierarchy

Although the administrative privilege classes are enumerated above, all administrators, regardless of the privilege classes associated with them, are considered trusted. The privilege classes simply provide a convenient way for administrators to limit their capability if they so desire.

Certain privilege classes also allow the association of Client Access Authorities to an administrator. Client access authorities are:

- **Client Owner** – Owns the data and has the right to gain access to the data regardless of the machine from which the data is requested. Can also delete file spaces and archive data. Can change the client node's password for which they have authority.
- **Client Access** – Can restore data only to the original client. Can only access the data from the owning client node.

The privilege classes that can have client access authorities associated with them are: system, policy, and node privileges. Administrators with system privilege and policy privilege always have client owner authority. An administrator with node privilege is given client access authority by default, but can be assigned client owner authority. Client Access Authorities only apply to administrators when logged into a B/A Client CLI.

7.1.5.1.2 Servers

At installation, IBM Tivoli Storage Manager provides a server options file named dsmserv.opt, which contains a set of default options to start the server. This server options file is protected by the environment such that only the administrator in the environment (i.e., the OS administrator) has permission to modify the server options file. The TSF allows a TOE administrator with system privilege to use the IBM Tivoli Storage Manager Console to edit the options specified in the server options file.

7.1.5.1.3 Client Nodes

Client nodes perform backup, archive, and restoration tasks using the Backup/Archive Client. A node is equivalent to a computer, as in the case of a Backup/Archive Client that is installed on a computer for file system backups.

Copyright © IBM Corp. 2010-11

8 Protection Profile Claims

8.1 PP Reference

This Security Target does not claim conformance with any Protection Profile that has been registered and / or evaluated.

Copyright © IBM Corp. 2010-11

9 Rationale

The rationale section provides additional information and demonstrates that the security objectives and the security functions defined in the previous chapter are consistent and sufficient to counter the threats defined in chapter 2.

9.1 Security Objectives Rationale

9.1.1 Security Objectives Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective is at least covered by one threat or policy.

Table 9-1 - Mapping Objectives to Threats and Policies

TOE Objective	Threat/Policy
O.AUTHENTICATE	T.BYPASS
O.SECURE_DEFAULTS	T.ACCESS, T.BYPASS
O.NOBYPASS	T.ACCESS, T.BYPASS
O.USERDATA	T.ACCESS, T.BYPASS
O.ADMDATA	T.ACCESS, T.BYPASS
O.PRIVS	P.PRIVS

Table 9-2 - Mapping Objectives for the Environment to Threats, Assumptions and Policies

Environment Objective	Threat/Assumption/Policy
OE.ADMIN	A.ADMIN
OE.PHYSICAL	A.PHYSICAL
OE.SERVER_RT OE.CLIENT_RT OE.PASSWORD OE.OS_SUPPORT	A.SERVER_RT A.CLIENT_RT P.PASSWORD A.OS_SUPPORT

9.1.2 Security Objectives Sufficiency

The following rationale provides justification that the security objectives are suitable to cover each individual assumption. Each security objective that traces back to an assumption about the use of the TOE, when achieved, actually contributes to the TOE achieving consistency with the assumption, and that if all security objectives that trace back to an assumption are achieved, the intended usage is supported.

Table 9-3 - Assumptions to Objectives Rationale

Assumption	Rationale for Objective
A.PHYSICAL	This assumption is addressed by the environmental objective OE.PHYSICAL which states those responsible for the TOE must ensure that the TOE and its underlying hardware and software are physically protected from unauthorized physical access and tampering. This also prevents unauthorized access to Server's keystores containing both public and private RSA keys. Furthermore, the server console, which does not require authentication, must be protected from unauthorized access.

Copyright © IBM Corp. 2010-11

Assumption	Rationale for Objective
A.ADMIN	This assumption is addressed by the environmental objective OE.ADMIN that states those responsible for the administration of the TOE must be trained such that they are capable of managing the TOE and the security of the information it contains and administrators are trustworthy. OE.ADMIN also states that client node users of the TOE are trustworthy.
A.SERVER_RT	The assumption on exclusive TOE use of the underlying machines for the TOE and prevent unauthorized access is achieved by the objective OE.SERVER_RT to implement corresponding measures for the Storage Server.
A.CLIENT_RT	The assumption that client node users protect their client node systems is achieved by the objective OE.CLIENT_RT to ensure management and training of users.
A.OS_SUPPORT	The assumption that the environment provides a reliable time stamp is achieved by the objective OE.OS_SUPPORT.

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented:

Table 9-4 - Organizational Security Policy to Objectives Rationale

OSP	Rationale for Objective
P.PRIVS	This OSP is address by the TOE objective O.PRIVS which states that the TOE must provide a mechanism to limit the scope of control of a given administrator.
P.PASSWORD	The OSP addresses the objective OE.PASSWORD which requires thatminimum security requirements for data encryption passwords exist and are enforced . By demanding that good quality data encryption passwords are used, the probability of guessing correctly the data encryption password is reduced to render correct guessing practically impossible.

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Table 9-5 - Sufficiency of Objectives to Counter Threats Rationale

Threat	Rationale for Objective
T.ACCESS	The threat that an authorized user of the TOE could gain unauthorized access to resources or information protected by the TOE, or perform operations for which no access rights have been granted, via user error, system error, or other actions is partially removed by O.NOBYPASS which states that the security policy enforcement functions must be invoked and succeed before allowing access to TOE protected objects and functions. It is also partially removed by O.USERDATA and O.ADMDATA which state that user data and TSF data must be protected from disclosure. Finally, O.SECURE_DEFAULTS ensures that only secure values are used by administrators when managing the configuration of the cryptographic functions.

Copyright © IBM Corp. 2010-11

T.BYPASS	The threat of exploitation of non-TSF portions of the TOE to circumvent the TSF is mitigated by the objective to enforce access control mechanisms before users and administrators can interact with the TOE, as expressed in O.NOBYPASS. (The Server Console does not enforce an access control mechanism.) The threat that an unauthorized user could view the backup, archive, and restore data as it crosses the network is removed by the objective O.USERDATA which states the TOE must protect backup, archive, and restore data from disclosure. The threat that an unauthorized user could view or modify TSF data transmitted between the administrative client and server or between the client node and server containing information protected by the TOE is removed by the objective O.ADMDATA. O.ADMDATA states that the TOE must protect TSF data from disclosure. The threat that an unauthorized user could use a spoof client node to obtain access to a server by pretending to be a different client node; thus, restoring data or changing client node information that the spoofing client node would normally not have access to is removed by O.AUTHENTICATE. O.AUTHENTICATE states the TOE must ensure that all users using client authentication not using the TSM server console, are identified and authenticated before being granted access to the TOE mediated resources. The threat that an unauthorized user could use a spoof server to obtain access to client node account information and administrative client account information such as passwords and to obtain backup and archive objects from a client node is removed by O.AUTHENTICATE. O.AUTHENTICATE states the Server must authenticate to the clients prior to the clients authenticating to the Server, which is accomplished through the use of TLSv1 and certificates. The threat that a malicious user could reduce the cryptographic strength of the cryptographic features of the TOE by succeeding in modifying the values of cryptographic attributes and replacing them by weak values is removed by O.SECURE_DEFAULTS. Each time a modification in the value of a cryptographic attribute occurs, the new value is examined to ensure that only values providing strong cryptography are accepted.

9.2 Security Functional Requirements Rationale

This chapter provides the rationale for the selection of security requirements.

9.2.1 Security Functional Requirements Coverage

The following table provides a mapping of security functional requirements to objectives, showing that each security functional requirement covers at least one objective and that each objective is covered by at least one security functional requirement.

Table 9-6 - Mapping Security Functional Requirements to TOE Security Objectives

SFR	Objective

SFR	Objective					
	O.AUTHENTICATE	O.PRIVS	O.NOBYPASS	O.USERDATA	O.ADMDATA	O.SECURE_DEFAULTS
FCS_CKM.1-sym	X			X	X	X
FCS_CKM.1-rsa	X			X	X	X
FCS_CKM.2-sym	X			X	X	X
FCS_CKM.2-rsa	X			X	X	X
FCS_CKM.4	X			X	X	X
FCS_COP.1sym	X			X	X	X
FCS_COP.1-rsa	X			X	X	X
FCS_COP.1-enc				X		X
FDP_ACC.1-enc				X		
FDP_ACC.1-obj				X		
FDP_ACC.1-prv		X				
FDP_ACF_EXT.1-enc				X		
FDP_ACF.1-obj				X		
FDP_ACF.1-prv		X				
FDP_ITT.1				X		
FDP_RIP_EXT.2-rnd				X		
FDP_RIP_EXT.2-enc				X		
FIA_AFL.1-nda	X					
FIA_AFL.1-ada	X					
FIA_ATD.1	X					
FIA_SOS.1	X					
FIA_SOS.2	X					
FIA_UAU.1	X		X			
FIA_UID.1	X		X			
FIA_USB.1	X					
FMT_MSA.1-enc				X		
FMT_MSA.1-obj				X		
FMT_MSA.2						X
FMT_MSA.3-enc				X		
FMT_MSA.3-obj				X		
FMT_MSA.3-prv		X				
FMT_MTD.1-acm					X	
FMT_MTD.1-ccm					X	
FMT_MTD.1-ipm					X	
FMT_MTD.1-cpm					X	
FMT_MTD.1-ppm	X					
FMT_MTD.1-scm					X	
FMT_SMF.1		X		X		
FMT_SMR.1		X		X		
FPT_ITT.1					X	
FTA_SSL.3	X					

Copyright © IBM Corp. 2010-11

9.2.2 Functional Requirements Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the TOE security functional requirements are suitable to meet and achieve the security objectives.

Table 9-7 - Mapping TOE objectives to SFRs for the TOE

Objective	Security Function Requirements for the TOE
O.AUTHENTICATE	FCS_CKM.1-sym & FCS_CKM.1-rsa require the TSF generate cryptographic keys for TLSv1 communications. FCS_CKM.2-sym & FCS_CKM.2-rsa require the TSF to distribute cryptographic keys for TLSv1 communications. FCS_CKM.4 requires that the TSF destroy cryptographic keys used for TLSv1 communications. FCS_COP.1-sym & FCS_COP.1-rsa require the TSF perform symmetric encryption and symmetric decryption when using TLSv1 communications. FMT_MTD.1-ppm allows the TOE to manage the password policy rules associated with authentication passwords. FIA_AFL.1-nda requires that the TSF detect authentication failures on client node accounts. FIA_AFL.1-ada requires that the TSF detect authentication failures on administrative accounts. FIA_ATD.1 requires the TSF maintain security attributes for users and assign to users during authentication. FIA_SOS.1 requires metrics for passwords FIA_SOS.2 requires password generation to specified metrics. FIA_UAU.1 requires user authentication before any TOE mediated actions can be performed everywhere except at the TSM server console. FIA_UID.1 requires user identification before any TOE mediated actions can be performed everywhere except at the TSM server console. FIA_USB.1 requires user–subject binding. FTA_SSL.3 requires the TSF to terminate an interactive session after a period of inactivity, thus, requiring the user to re-authenticate.
O.PRIVS	FDP_ACC.1-prv defines the existence of the privilege class management mechanism. FDP_ACF.1-prv defines the attributes of the privilege class management mechanism. FMT_MSA.3-prv defines that an administrator with system privilege class can

	define the initial values of the privilege classes for an administrator account. FMT_SMF.1 defines privilege class management as a TOE management function. FMT_SMR.1 defines the administrator management role.
O.NOBYPASS	FIA_UAU.1 requires user authentication before any TOE mediated actions can be performed everywhere except at the TSM server console. FIA_UID.1 requires user identification before any TOE mediated actions can be performed everywhere except at the TSM server console.
O.USERDATA	FCS_CKM.1-sym, FCS_CKM.1-rsa, FCS_CKM.2-sym, FCS_CKM.2-rsa, FCS_COP.1-sym, FCS_COP.1-rsa, and FDP_ITT.1 require the TSF to protect user data during transfer using TLS. FCS_COP.1-enc specifies that a user can encrypt backup and archive data on the B/A Client prior to transmitting it to the Server using a password only known to the B/A Client. FDP_ACC.1-obj, FDP_ACF.1-obj, FMT_MSA.1-obj, FMT_MSA.3-obj, FMT_SMF.1, and FMT_SMR.1 require the TSF to provide access control lists to collections of objects owned by the client node and to allow the client node to manage the access control lists. Additionally, FDP_ITT.1 supports the non-disclosure of these objects when transmitted to and from the Storage Server. FDP_ACC.1-obj, FDP_ACF.1-obj, FMT_MSA.1-obj, FMT_MSA.3-obj, FMT_SMF.1, and FMT_SMR.1 require the TSF to use FCS_COP.1-enc as an access control mechanism that prevents discloser of data contained in the backup and archive data objects. FDP_ACC.1-enc, FDP_ACF_EXT.1-enc, FMT_MSA.1-enc, FMT_MSA.3-enc, FMT_SMF.1 and FMT_SMR.1 require the TSF to support the encryption of user data that is stored as backup and archive data objects. FDP_RIP_EXT.2-rnd … The random number generator's state is reset when the module is unloaded and sensitive information in the buffers of a session is cleared before the buffer is released. Neither the buffer content nor the random number generator state are then any longer available. FDP_RIP_EXT.2-enc Cryptographic keys (symmetric keys and asymmetric key pairs) are destroyed when no longer used.
O.ADMDATA	FCS_CKM.1-sym, FCS_CKM.1-rsa, FCS_CKM.2-sym, FCS_CKM.2-rsa, FCS_COP.1-sym, FCS_COP.1-rsa, and FPT_ITT.1 require the TSF to protect administrative and TSF data during transfer using TLS. All FMT_MTD.1 iterations except FMT_MTD.1-ppm require the TSF to restrict the ability to initialize and modify TSF data.
O.SECURE_DEFAULTS	FMT_MSA.2 meets the objective by ensuring that only secure values are accepted as values of cryptographic attributes. The secure values are defined in the corresponding SFRs from the functional class FCS (Cryptographic Support). While FMT_MSA.2 is for ensuring that only secure values are accepted for cryptographic attributes, those secure values are defined in functional class FCS. The security attributes govern

	attributes for generating cryptographic keys as stated in FCS_CKM.1-sym and FCS_CKM-1-rsa, attributes for the distribution of keys as stated in FCS_CKM.2-sym and FCS_CKM.2-rsa, attributes for destroying cryptographic keys as stated in FCS_CKM.4, and attributes for governing the actual cryptographic operations as stated in FCS_COP.1-sym, FCS_COP.1-rsa, and FCS_COP.1-enc.

9.2.3 Security Requirements Dependency Analysis

The following table shows the dependencies between the security functional requirements for the TOE and their resolution in this Security Target.

SFRs in italic type setting show dependent SFRs that have not been resolved.

The following table shows that the TOE security functions specified in the TOE summary specification meet all the security functional requirements for the TOE and work together to satisfy the TOE security functional requirements.

Table 9-8 - Dependencies between TOE Security Functional Requirements

SFR	CC Part 2 Dependencies	Resolved
FCS_CKM.1-sym	[FCS_CKM.2; FCS_COP.1] FCS_CKM.4	FCS_COP.1-sym FCS_CKM.4
FCS_CKM.1-rsa	[FCS_CKM.2; FCS_COP.1] FCS_CKM.4	FCS_COP.1-rsa FCS_CKM.4
FCS_CKM.2-sym	[FDP_ITC.1; FDP_ITC.2; FCS_CKM.1] FCS_CKM.4	FCS_CKM.1-sym FCS_CKM.4
FCS_CKM.2-rsa	[FDP_ITC.1; FDP_ITC.2; FCS_CKM.1] FCS_CKM.4	FCS_CKM.1-rsa FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1; FDP_ITC.2; FCS_CKM.1]	FCS-CKM.1-*
FCS_COP.1-sym	[FDP_ITC.1; FDP_ITC.2; FCS_CKM.1] FCS_CKM.4	FCS_CKM.1-sym FCS_CKM.4
FCS_COP.1-rsa	[FDP_ITC.1; FDP_ITC.2; FCS_CKM.1] FCS_CKM.4	FCS_CKM.1-rsa FCS_CKM.4
FCS_COP.1-enc	FCS_CKM.1 FCS_CKM.4	No No
FDP_ACC.1-enc	FDP_ACF.1	FDP_ACF_EXT.1-enc
FDP_ACC.1-obj	FDP_ACF.1	FDP_ACF.1-obj
FDP_ACC.1-prv	FDP_ACF.1	FDP_ACF.1-prv
FDP_ACF_EXT.1-enc	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1-enc FMT_MSA.3-enc
FDP_ACF.1-obj	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1-obj FMT_MSA.3-obj
FDP_ACF.1-prv	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1-prv FMT_MSA.3-prv
FDP_ITT.1	[FDP_ACC.1; FDP_IFC.1]	FDP_ACC.1-obj
FDP_RIP_EXT.2-rnd	None	Yes
FDP_RIP_EXT.2-enc	None	Yes
FIA_AFL.1-nda	FIA_UAU.1	FIA_UAU.1
FIA_AFL.1-ada	FIA_UAU.1	FIA_UAU.1

Copyright © IBM Corp. 2010-11

SFR	CC Part 2 Dependencies	Resolved
FIA_ATD.1	None	Yes
FIA_SOS.1	None	Yes
FIA_SOS.2	None	Yes
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UID.1	None	Yes
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MSA.1-enc	[FDP_ACC.1; FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1-enc FMT_SMR.1 FMT_SMF.1
FMT_MSA.1-obj	[FDP_ACC.1; FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1-obj FMT_SMR.1 FMT_SMF.1
FMT_MSA.2	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	No No FMT_SMR.1
FMT_MSA.3-enc	*FMT_MSA.1* FMT_SMR.1	No FMT_SMR.1
FMT_MSA.3-obj	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1-obj FMT_SMR.1
FMT_MSA.3-prv	*FMT_MSA.1* FMT_SMR.1	No FMT_SMR.1
FMT_MTD.1-acm	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1-ccm	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1-ipm	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1-cpm	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1-ppm	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1-scm	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_SMF.1	None	Yes
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_ITT.1	None	Yes
FTA_SSL.3	None	Yes

The rationale for unresolved dependencies on TOE SFRs is given in the following table:

Table 9-9 – Unresolved TOE Security Function Requirements Dependency Rationale

SFR	Unresolved Dependency	Rationale
FCS_COP.1-enc	FCS_CKM.1 FCS_CKM.4	The cryptographic key is the data encryption password which is not created by the TOE but selected by the human user. As such, the TOE does not generate the key and the dependency is not applicable. Hence, the quality of the data encryption password is covered by P.PASSWORD. As the key is not managed by the TOE but created from the password entered by the human user,

SFR	Unresolved Dependency	Rationale
		there is no key destruction applicable to the TOE. Hence, FCS_CKM.4 is not applicable.
FMT_MSA.3-prv	FMT_MSA.1	The privilege class mechanism is not only used by the TOE to manage access to user data, it is also used to manage access to itself. Because of this, FDP_ACF.1-prv describes the management of privilege classes making FMT_MSA.1 redundant.
FMT_MSA.2	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1	The dependencies stated by the CC imply that this requirement should be scoped to security attributes associated with access control or information flow policies. However, there are other requirements which would equally benefit by requiring the use of secure values. One such example are Cryptographic attributes that are defined for cryptographic keys and cryptographic operations in functional class FCS (Cryptographic support). These may include the methods for generating or importing, distributing or accessing, and destroying cryptographic keys as well as key sizes and the actual cryptographic operations carried out by the TOE. The TOE ensures that values meeting these constraints and no other values are accepted for the stated cryptographic security attributes. Therefore, the TOE is applying FMT_MSA.2 not against a user data protection mechanism, but as a constraint on cryptographic values, thus making the dependencies upon access control and information flow requirements unnecessary.

9.2.4 Internal Consistency and Mutual Support of SFRs

Section 9.2.2 demonstrates how the security requirements work together to implement the individual objectives for the TOE and the environment. This section elaborates on the internal consistency and mutual support of the security requirements.

The main purpose of the TOE is to backup and archive data from one computer to a central server and to be able to retrieve the data when necessary. In order to do this in a secure manner, the TOE employs TLS to protect the communications between the client components of the TOE and the server component protecting both TSF and user data. TLS generate session keys (FCS_CKM.1-sym, FCS_CKM.1-rsa, FMT_MSA.2) when establishing a connection and distribute the keys (FCS_CKM.2-sym, FCS_CKM.2-rsa) in accordance to the protocols and using cryptographic operations (FCS_COP.1-sym, FCS_COP.1-rsa). Using TSL protects both TSF data (FPT_ITT.1) and user data (FDP_ITT.1). In addition, the Server will terminate a client session after a period of inactivity (FTA_SSL.3) properly destroying cryptographic keys once they are no longer needed (FCS_CKM.4). Further assurance on the security of cryptographic functions is obtained by ensuring that only values considered secure are accepted as cryptographic attributes (FMT_MSA.2).

The TOE supports mutual authentication of clients to the Server. Each client has the public key of the Server; therefore, it can validate the identity of the Server. The Server uses a user ID and password-based mechanism to identify (FIA_UID.1) and authenticate (FIA_UAU.1) a client. When a user properly authenticates, the TOE associates security attributes with the user (FIA_USB.1). If an account has 3 consecutive unsuccessful authentication attempts, the account is locked until an authorized administrator

Copyright © IBM Corp. 2010-11

unlocks the account (FIA_AFL.1-nda and FIA_AFL.1-ada). The user attributes associated with a user are defined in FIA_ATD.1 and user account attributes are managed by authorized administrators as defined by FMT_MTD.1-acm.

Authentication passwords have an expiration and composition quality requirements (FIA_SOS.1). The expiration and composition quality parameters are managed by an authorized administrator (FMT_MTD.1-ppm). Authentication passwords can also be generated by the TOE (FIA_SOS.2). This allows the client node software to automatically generate new passwords and to automatically change them when they expire when the client is running in an unattended mode. Authentication passwords are initialized by an authorized administrator (FMT_MTD.1-ipm) and can be modified as per FMT_MTD.1-cpm.

The B/A Client supports data encryption passwords used to encrypt backups and archives (FCS_COP.1-enc) before sending them to the Server. This password provides a security attribute for the access control mechanism (FDP_ACC.1-enc, FDP_ACF.1-enc) to the data by making it difficult for another user who may have access to the backups/archives from restoring the backups/archives. The management of this feature is described by FMT_MSA.1-enc, and FMT_MSA.3-enc.

A client node can control which other client node accounts can access its backups and archives (i.e., user data) through a list of rules (FDP_ACC.1-obj and FDP_ACF.1-obj). The management of the rules is described by FMT_MSA.1-obj and FMT_MSA.3-obj. The SFR FDP_ITT.1 also aids in supporting this by protecting the backups and archives from disclosure during transit between the B/A Client and Storage Server.

Administrator accounts have privilege classes and client authorities associated with their accounts, which functions as an access control mechanism controlling the degree of power an administrator has in the TOE (FDP_ACC.1-prv and FDP_ACF.1-prv). The management of this feature is described by FMT_MSA.3-prv, and FMT_MTD.1-acm.

The B/A Client and Storage Server have editable configuration files containing TSF data which are covered by FMT_MTD.1-ccm and FMT_MTD.1-scm.

Management of the TOE in general and the roles defined by the TOE defined by FMT_SMF.1 and FMT_SMR.1 and include all the management and roles specified in this section.

The TOE is designed to be non-bypassable. For example, the B/A Client and Administrative Client cannot access data on the Server until they successfully log into the Server. Once logged in, a client node cannot access backups and archives until the discretionary access control associated with the user data on the Server is enforced.

9.3 TOE Summary Specification Rationale

9.3.1 Security Functions Justification

The following tables show that the security functions specified in the TOE summary specification meet all the security functional requirements for the TOE and work together to satisfy the TOE security functional requirements.

Copyright © IBM Corp. 2010-11

Table 9-10 – Quick Mapping of TOE SFRs to TSF

SFR	Security Function											
	IA.1	IA.2	IA.3	IA.4	IA.5	IA.6	ACCESS.1	DATA_PROT.1	PM.1	PM.2	SM.1	(empty)
FCS_CKM.1-sym					X							
FCS_CKM.1-rsa					X							
FCS_CKM.2-sym					X							
FCS_CKM.2-rsa					X							
FCS_CKM.4					X							
FCS_COP.1-sym					X							
FCS_COP.1-rsa					X							
FCS_COP.1-enc								X				
FDP_ACC.1-enc								X				
FDP_ACC.1-obj							X					
FDP_ACC.1-prv											X	
FDP_ACF_EXT.1-enc								X				
FDP_ACF.1-obj							X					
FDP_ACF.1-prv											X	
FDP_ITT.1					X							
FDP_RIP_EXT.2-rnd					X							
FDP_RIP_EXT.2-enc					X							
FIA_AFL.1-nda			X									
FIA_AFL.1-ada			X									
FIA_ATD.1						X						
FIA_SOS.1									X			
FIA_SOS.2										X		
FIA_UAU.1	X											
FIA_UID.1	X											
FIA_USB.1	X											
FMT_MSA.1-enc								X				
FMT_MSA.1-obj							X					
FMT_MSA.2					X							
FMT_MSA.3-enc								X				
FMT_MSA.3-obj							X					
FMT_MSA.3-prv											X	
FMT_MTD.1-acm		X									X	
FMT_MTD.1-ccm											X	
FMT_MTD.1-ipm		X								X		
FMT_MTD.1-cpm										X		
FMT_MTD.1-ppm									X			
FMT_MTD.1-scm											X	
FMT_SMF.1		X	X				X		X	X	X	
FMT_SMR.1		X	X		X		X	X	X	X	X	
FPT_ITT.1					X							
FTA_SSL.3				X								

Copyright © IBM Corp. 2010-11

Table 9-11 - Mapping of TOE SFRs to TSF

SFR	Security Function
FCS_CKM.1-sym	The IA.5 security function includes the GSKit library that implements TLSv1. The GSKit library is responsible for the generation of the symmetric keys for the RC4, TDES, and AES algorithms.
FCS_CKM.1-rsa	The IA.5 security function includes the GSKit library that implements TLSv1. The GSKit library is responsible for the generation of the symmetric keys for the RSA algorithm.
FCS_CKM.2-sym	The IA.5 security function implements the TLS protocol's handshake which exchanges session keys as part of the protocol's initialization.
FCS_CKM.2-rsa	The IA.5 security function implements the TLS protocol's handshake which exchanges session keys as part of the protocol's initialization.
FCS_CKM.4	The IA.5 security function ensures that it clears (i.e., writes zeroes over) memory space used for cryptographic keys when a key is destroyed.
FCS_COP.1-sym	RC4, TDEA, and AES are implemented by the GSKit library that is part of the IA.5 security function.
FCS_COP.1-rsa	Digital signature generation and verification with RSA is implemented by the GSKit Library TLSv1 functionality which is a part of the IA.5 security function.
FCS_COP.1-enc	AES encryption is implemented in DATA_PROT.1.
FDP_ACC.1-enc	The Data Encryption SFP is implemented in DATA_PROT.1.
FDP_ACC.1-obj	The Stored Object Access Control SFP is implemented in ACCESS.1
FDP_ACC.1-prv	The Privilege Class Management SFP is implemented in SM.1.
FDP_ACF_EXT.1-enc	The encryption of data that is stored in backup and archive objects is performed in DATA_PROT.1.
FDP_ACF.1-obj	The rules of the Stored Object Access Control SFP are implemented in ACCESS.1.
FDP_ACF.1-prv	The rules of the Privilege Class Management SFP are implemented in SM.1.
FDP_ITT.1	The non-disclosure of user data is accomplished by the protection of data in transit using TLS which is part of the IA.5 security function.
FDP_RIP_EXT.2-rnd	The clearing of data associated with random number generator occurs within the GSKit which provides capabilities of IA.5.
FDP_RIP_EXT.2-enc	The clearing of data associated with sensitive data occurs withing the GSKit which provides capabilities of IA.5.
FIA_AFL.1-nda	The IA.3 security function implements the constraint that three consecutive failed logon attempts by a client node cause the account used by the client to become locked until an administrator unlocks the account.
FIA_AFL.1-ada	The IA.3 security function implements the constraint that three consecutive failed logon attempts by an administrative client cause the account used by the client to become locked until an administrator unlocks the account.
FIA_ATD.1	The administrator and client node account security attributes are defined in IA.6.
FIA_SOS.1	The password strength policy is implemented in PM.1.
FIA_SOS.2	The password generation policy is implemented in PM.2.
FIA_UAU.1	User authentication is implemented in IA.1.
FIA_UID.1	User identification is implemented in IA.1.
FIA_USB.1	User-subject binding is implemented in IA.1
FMT_MSA.1-enc	The client configuration information and encryption password are implemented in DATA_PROT.1.
FMT_MSA.1-obj	Management of the Stored Object Access Control SFP is implemented in ACCESS.1.

Copyright © IBM Corp. 2010-11

FMT_MSA.2	The IA.5 security function ensures that only secure values are accepted when cryptographic attributes are being set.
FMT_MSA.3-enc	Default values for the encryption configuration and encryption password are implemented in DATA_PROT.1
FMT_MSA.3-obj	Restrictive default values for the Stored Object Access Control SFP is implemented in ACCESS.1.
FMT_MSA.3-prv	The SM.1 security function forces privilege classes to be assigned to accounts. If no privilege class is assigned, the account is unprivileged.
FMT_MTD.1-acm	Privilege classes for accounts are set by the SM.1 security function, while other account management operations (e.g., creation, removal, lock/unlock) are implemented in IA.2.
FMT_MTD.1-ccm	Client configuration management is implemented in SM.1.
FMT_MTD.1-ipm	The creation of initial passwords is implemented in PM.2, while the enforcement that only administrators with system privilege can create accounts is implemented in IA.2.
FMT_MTD.1-cpm	Modification of account passwords is implemented in PM.2.
FMT_MTD.1-ppm	Password policy management is implemented in PM.1.
FMT_MTD.1-scm	Server configuration control is implemented in SM.1 through the enforcement of privilege classes to protect the server options file. It is further supported by the enforcement of access controls on the server options file that is performed by the environment.
FMT_SMF.1	TSF management functionality for data used by each security function is implemented by each security function. IA.2 implements account management functionality, IA.3 implements an ability to lock accounts for too many failed logon attempts, ACCESS.1 implements the management of attributes of the stored object access control SFP, PM.1 implements password policy management functionality, PM.2 implements password management, and SM.1 implements client configuration management, server configuration management, and privilege class management.
FMT_SMR.1	The definition of security roles is implemented in SM.1. The use of security roles to define various operations permitted by each role is implemented in the security function IA.2, IA.3, IA.5, ACCESS.1, DATA_PROT.1, PM.1, PM.2, and SM.1.
FPT_ITT.1	TSF data in transfer is protected by an encrypted channel between the client node CLI and Server and between the Administrative Client CLI and Server in IA.5.
FTA_SSL.3	Session termination is implemented in IA.4

9.3.2 Mutual Support of Security Functions

The primary goal of the TOE is to transfer user data from a client system to a trusted server where the server is a backup/archive mechanism for the client system, and to provide protection of this user data during transport to and from the server as well as to protect the user data residing on the server from access by unauthorized users.

To protect the data while on the server side, the TOE requires all client node users and administrative client users to logon using individual accounts (IA.1). To deter people from guessing an account's password, the server side employs an account lockout mechanism (IA.3).

The system supports a mutual authentication mechanism whereby a client node authenticates the server through TLS using the server's public/private keys (IA.5) and the server authenticates the client nodes

using an account name and password (IA.1, IA.6). The TLS mechanism is used by the TOE to aid in preventing the disclosure of TSF and user data (ACCESS.1) while in transit between a client and the server.

When a user logs in, the TOE creates a session between the client and the server. As a security feature, the TOE implements an inactivity timeout feature (IA.4) causing the session to be automatically terminated after a period of inactivity.

The authentication passwords are required to conform to certain password complexity rules including password expiration (PM.1). The authentication passwords can also be generated by the TOE (PM.2).

Once the user data is on the server, each client node account can control which other client node accounts can access the data, if any, through a list of access rules (ACCESS.1). Additionally, a client node can encrypt the backups and archives with a private password (DATA_PROT.1) making the objects more difficult to interpret in case of accidental disclosure. These data encryption passwords are generated by the TOE (PM.2).

The TOE provides management features to manage user account data (IA.2, PM.1). It defines two types of users (administrators and client node users) and provides a privilege mechanism to limit the power of a given administrator (SM.1, IA.6).

As a result:

- No sessions using the B/A Client CLI or Administrative Client CLI can be established without proper authentication.
- Sessions between the B/A Client CLI and Administrative Client CLI are protected from disclosure.
- Backups and archives transferred to the server have discretionary access control (DAC) while residing on the server where the DAC can be managed by the user.
- Only authorized users can manage and control TSF data.

Copyright © IBM Corp. 2010-11

10 References

[CC] Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 3, July 2009, Parts 1 to 3.

[CEM] Common Methodology for Information Technology Security Evaluation, CCMB-2009-07-004, Evaluation methodology, Version 3.1 Revision 3, July 2009.

[EPRAND] EUROPEAN PATENT APPLICATION EP 1 081 591 A2, Random number generator, Application number: 00114754.5, Date of publication: 07.03.2001 Bulletin 2001/10.

[FIPS140-2] FIPS PUB 140-2: SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES, Issued May 25, 2001, including CHANGE NOTICES (12-03-2002).

[FIPS186-2] FIPS PUB 186-2: DIGITAL SIGNATURE STANDARD (DSS), including Change Notice, January 27, 2000

[FIPS197] FIPS PUB 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES), November 26, 2001.

[RFC1321] IETF RFC 1321: The MD5 Message-Digest Algorithm, including the Erratum for RFC 1321, April 1992

[RFC2246] IETF RFC 2246: The Transport Layer Security (TLS) Protocol Version 1.0

[RFC2313] IETF RFC 2313: PKCS #1: RSA Encryption Version 1.5

[RFC3268] IETF RFC 3268: Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)

Copyright © IBM Corp. 2010-11

11 Abbreviations

Abbreviation	Description
AES	Advanced Encryption Standard
AIX	Advanced Interactive Executive
ANSI	American National Standards Institute
API	Application Programming Interface
CC	Common Criteria
CCIMB	Common Criteria Interpretations Management Board
CD	Compact Disc
CLI	Command Line Interface
CRL	Certificate Revocation List
DES	Data Encryption Standard
EE	End-entiry
FIPS	Federal Information Processing Standard
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISO	International Standards Organization
NAS	Network-Attached Storage
NDMP	Network Data Management Protocol
OSP	Organizational Security Policy
PDF	Portable Data Format
PP	Protection Profile
RSA	Rivest-Shamir-Adleman
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
ST	Security Target
TCP	Transmission Control Protocol
TDEA	Triple Data Encryption Algorithm
TLS	Transport Layer Security
TOE	Target of Evaluation
TOE	Target of Evaluation
TRNG	True random number generator
TSF	TOE Security Functions
TSM	Tivoli Storage Manager

Copyright © IBM Corp. 2010-11