

BLOOMBASE CONFIDENTIAL



# Bloombase StoreSafe – Interoperability and Exit Form: IBM Security Key Lifecycle Manager (SKLM)

**BLOOMBASE<sup>®</sup>**

This document contains information of a proprietary nature. **ALL INFORMATION CONTAINED HEREIN SHALL BE KEPT IN CONFIDENCE.** None of this information shall be divulged to persons other than Bloombase employees authorized by the nature of their duties to receive such information, or individuals or organizations authorized by Bloombase research and development in accordance with existing policy regarding the release of company information

**BLOOMBASE CONFIDENTIAL**

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, people and events depicted herein are fictitious and no association with any real company, organization, product, person or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Bloombase.

Bloombase may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Bloombase, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

This document is the property of Bloombase. No exploitation or transfer of any information contained herein is permitted in the absence of an agreement with Bloombase, and neither the document nor any such information may be released without the written consent of Bloombase.

© 2017 Bloombase, Inc.

Bloombase, Spitfire, StoreSafe are either registered trademarks or trademarks of Bloombase in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Document No.: Bloombase InteropLab - Interoperability and Certification – Interoperability and Exit Form - IBM SKLM - vo.95.doc

**BLOOMBASE CONFIDENTIAL**

# Table of Contents

Table of Contents	3
1. Document Revision History	6
2. Introduction	8
3. Certification Details	9
3.1 Products Tested	9
3.2 KMIP Features Supported	10
4. Certification Test Cases and Results	12

**BLOOMBASE CONFIDENTIAL**

5. Test Environment Setup	17
5.1 IBM SKLM Configuration	17
IBM SKLM Web Management Console	17
5.2 Bloomberg StoreSafe Configuration	18
Setting up Integration with KMIP Key Manager – IBM SKLM	20
KMIP Key Manager High Availability Clustering Configuration	24
CIFS Secure Storage Configuration	24
NFS Secure Storage Configuration	27
iSCSI Secure Storage Configuration	30
FCP Secure Storage Configuration	33
6. Interoperability Test Cases	37
6.1 Test Cases for Key Generation	37
6.1.1 Key Generation at IBM SKLM	37
6.1.2 Key Access at IBM SKLM	40
6.1.3 Key Access at IBM SKLM Replication Model Clone Node	42
6.2 Test Cases for CIFS Data Encryption and Decryption	44
6.2.1 SMB Authentication at CIFS Secure Storage	44
6.2.2 List CIFS Secure Storage	45
6.2.3 Connect and Access CIFS Secure Storage	46
6.2.4 Mount CIFS Secure Storage	46
6.2.5 Write to CIFS Secure Storage	48
6.2.6 Read from CIFS Secure Storage	49
6.2.7 Delete from CIFS Secure Storage	49
6.3 Test Cases for NFS Data Encryption and Decryption	51
6.3.1 Network Access Control at NFS Secure Storage	51
6.3.2 List NFS Secure Storage	51
6.3.3 Connect and Access NFS Secure Storage	51
6.3.4 Mount NFS Secure Storage	51
6.3.5 Write to NFS Secure Storage	52
6.3.6 Read from NFS Secure Storage	53
6.3.7 Delete from NFS Secure Storage	53
6.4 Test Cases for iSCSI Data Encryption and Decryption	53
6.4.1 Discover iSCSI Secure Storage	53
6.4.2 Connect to iSCSI Secure Storage	54
6.4.3 Dump in iSCSI Secure Storage Raw Device	54

**BLOOMBASE CONFIDENTIAL**

6.4.4 Dump out iSCSI Secure Storage Raw Device	55
6.4.5 Create Filesystem on iSCSI Secure Storage	55
6.4.6 Write to Filesystem on iSCSI Secure Storage	56
6.4.7 Read from Filesystem on iSCSI Secure Storage	56
6.4.8 Delete from Filesystem on iSCSI Secure Storage	57
6.5 Test Cases for FCP Data Encryption and Decryption	57
6.5.1 Initialize FCP Secure Storage Disk	57
6.5.2 Dump in FCP Secure Storage Raw Device	57
6.5.3 Dump out FCP Secure Storage Raw Device	58
6.5.4 Create Filesystem on FCP Secure Storage	58
6.5.5 Write to Filesystem on FCP Secure Storage	60
6.5.6 Read from Filesystem on FCP Secure Storage	60
6.5.7 Delete from Filesystem on FCP Secure Storage	61
7. Interoperability and Certification Confirmation	62
8. References	63

BLOOMBASE CONFIDENTIAL

# 1. Document Revision History

Revision	Date	Revised By	Comments
0.9	2017-1-3	Calvin Susanto, Bloombase	Initial draft for partner review
0.91	2017-1-13	Calvin Susanto, Bloombase	Interoperability test results documented
0.92	2017-1-16	Calvin Susanto, Bloombase	Interoperability test configuration documented
0.93	2017-1-16	Calvin Susanto, Bloombase	KMIP details documented
0.94	2017-1-16	Calvin Susanto,	Format changes

**BLOOMBASE CONFIDENTIAL**

---

		Bloomberg	
0.95	2017-1-16	Calvin Susanto, Bloomberg	Format changes

---

---

**BLOOMBASE CONFIDENTIAL**

## **2. Introduction**

The Bloombase StoreSafe– Interoperability and Exit Form: IBM SKLM documents the evaluation of the Bloombase solution with IBM Security Key Lifecycle Manager (SKLM) for data-at-rest encryption.

This Exit Form is meant to augment an existing Use Cases and Certification document (UCC) between Bloombase and IBM. The use cases documented in the UCC document are captured here in Pass/Fail test scenarios.

All test scenarios must pass for the solution to be certified.



**BLOOMBASE CONFIDENTIAL**

## **3. Certification Details**

Partner Company Name: IBM

Usage Details: The IBM SKLM will be used as the OASIS KMIP-compliant key manager with Bloombase StoreSafe for encryption of data-at-rest of unified storage systems and services.

Testing Date: January 9, 2017

Bloombase Representative(s): Michael Brew, Calvin Susanto

Partner Representative(s): Rinkesh Bansal, Mahesh Paradkar

### **3.1 Products Tested**

## BLOOMBASE CONFIDENTIAL

Item	Version	Details
IBM SKLM	2.7	OASIS KMIP-compliant key manager
Bloomberg StoreSafe	3.5	Data-at-rest encryption security software appliance for NAS, SAN, object stores and cloud storage service endpoints

## 3.2 KMIP Features Supported

Item	Details
KMIP Specification	<ul style="list-style-type: none"><li>• 1.0</li><li>• 1.1</li></ul>
Transport Protocol	<ul style="list-style-type: none"><li>• TLS 1.0</li><li>• TLS 1.1</li><li>• TLS 1.2</li></ul>
Authentication Mechanism	X.509 certificate
KMIP Operations	<ul style="list-style-type: none"><li>• Locate</li><li>• Get</li><li>• Get Attributes</li><li>• Get Attribute List</li><li>• Add Attribute</li><li>• Modify Attribute</li><li>• Delete Attribute</li></ul>

**BLOOMBASE CONFIDENTIAL**

---

KMIP Objects

- Certificate
  - Public Key
  - Private Key
  - Symmetric Key
-

BLOOMBASE CONFIDENTIAL

## 4. Certification Test Cases and Results

#	Test Case	Required Result	Pass   Fail	Test Case Number	Notes
1	Documentation: Documents to cover deployment and use of Bloombase-IBM solution				
2	Interoperability Testing			6	
	<b>Key Management</b>			6.1	

**BLOOMBASE CONFIDENTIAL**

Key generation at IBM SKLM	Key generated	Pass	6.1.1
Key access at IBM SKLM	Key retrieved	Pass	6.1.2
Key access at IBM SKLM replication model clone node	Key retrieved	Pass	6.1.3
<b>CIFS Data Encryption and Decryption</b>			6.2
SMB authentication at CIFS secure storage	Login success	Pass	6.2.1
List CIFS secure storage	List of shares returned	Pass	6.2.2
Connect and access CIFS secure storage	Share connected	Pass	6.2.3
Mount CIFS secure storage	Share mounted	Pass	6.2.4
Write to CIFS secure storage	File written to share	Pass	6.2.5
Read from CIFS secure storage	File read from share	Pass	6.2.6

## BLOOMBASE CONFIDENTIAL

Delete from CIFS secure storage	File deleted from share	Pass	6.2.7
<b>NFS Data Encryption and Decryption</b>			6.3
Network access control at NFS secure storage	Access success	Pass	6.3.1
List NFS secure storage	List of shares returned	Pass	6.3.2
Connect and access NFS secure storage	Share connected	Pass	6.3.3
Mount NFS secure storage	Share mounted	Pass	6.3.4
Write to NFS secure storage	File written to share	Pass	6.3.5
Read from NFS secure storage	File read from share	Pass	6.3.6
Delete from NFS secure storage	File deleted from share	Pass	6.3.7
<b>iSCSI Data Encryption and Decryption</b>			6.4
Discover iSCSI secure storage	iSCSI disk discovered	Pass	6.4.1

## BLOOMBASE CONFIDENTIAL

Connect to iSCSI secure storage	Disk connected	Pass	6.4.2
Dump in iSCSI secure storage raw device	Data written to disk	Pass	6.4.3
Dump out iSCSI secure storage raw device	Data read from disk	Pass	6.4.4
Create filesystem on iSCSI secure storage	Filesystem created on disk	Pass	6.4.5
Write to filesystem on iSCSI secure storage	File written to filesystem	Pass	6.4.6
Read from filesystem on iSCSI secure storage	File read from filesystem	Pass	6.4.7
Delete from filesystem on iSCSI secure storage	File deleted from filesystem	Pass	6.4.8
<b>FCP Data Encryption and Decryption</b>			6.5
Initialize FCP	Disk initialized	Pass	6.5.1

**BLOOMBASE CONFIDENTIAL**

secure storage disk				
	Dump in FCP secure storage raw device	Data written to disk	Pass	6.5.2
	Dump out FCP secure storage raw device	Data read from disk	Pass	6.5.3
	Create filesystem on FCP secure storage	Filesystem created	Pass	6.5.4
	Write to filesystem on FCP secure storage	File written to filesystem	Pass	6.5.5
	Read from filesystem on FCP secure storage	File read from filesystem	Pass	6.5.6
	Delete from filesystem on FCP secure storage	File deleted from filesystem	Pass	6.5.7
3	Final Results		Pass	



**BLOOMBASE CONFIDENTIAL**

## **5. Test Environment Setup**

### **5.1 IBM SKLM Configuration**

#### **IBM SKLM Web Management Console**

For this interoperability test, the test SKLM is provided by IBM test team and is hosted at IBM SoftLayer cloud infrastructure.

Bloombase StoreSafe client certificate is trusted by this Stand-Alone IBM SKLM with IP address 173.193.147.234 and port 5696.

## BLOOMBASE CONFIDENTIAL



IBM Security Key Lifecycle Manager

User ID:

Password:

 Licensed Materials - Property of IBM (c) Copyright IBM Corp. 2008, 2016 All Rights Reserved. IBM, the IBM logo, ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [Copyright and trademark information](#).

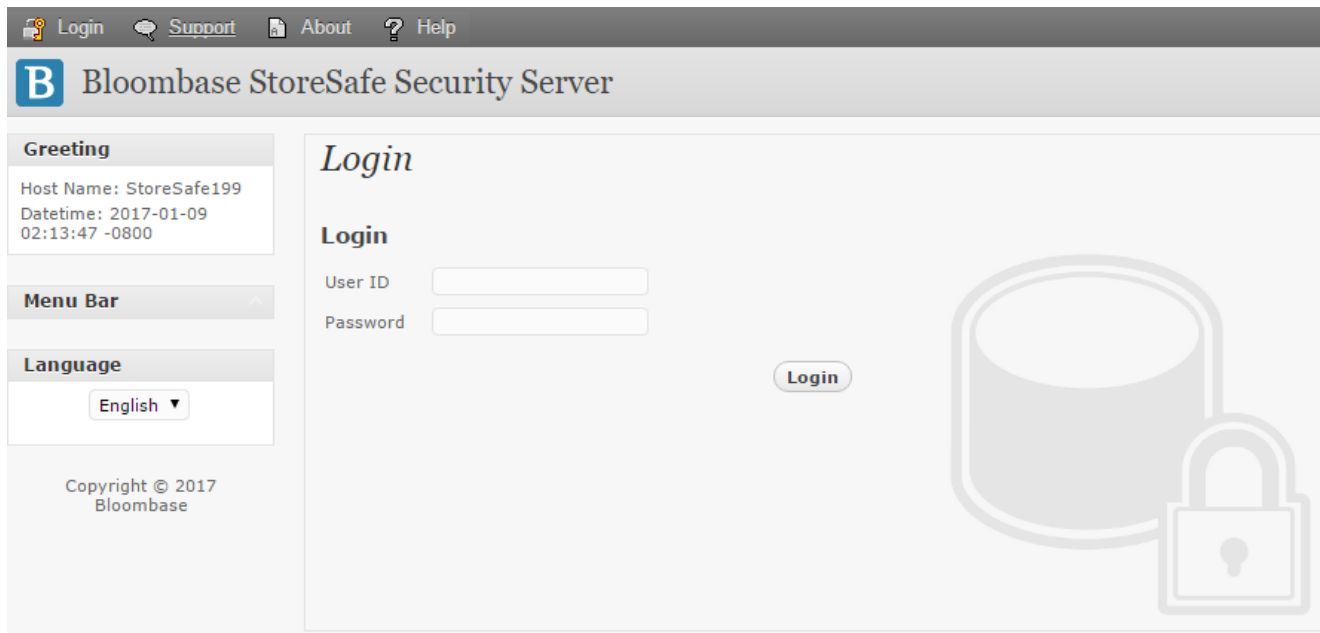
The exact same SKLM is used to test the interoperability between Bloomberg StoreSafe and IBM SKLM high availability cluster.

IBM test team has provided both “Master” and “Clone” SKLM server instances for the testing.

## 5.2 Bloomberg StoreSafe Configuration

Bloomberg StoreSafe Web Administration Console can be accessed using web browser at [https://<StoreSafe\\_IP>:8443](https://<StoreSafe_IP>:8443). For this test, Bloomberg StoreSafe with 192.168.12.199 as IP address is used and can be accessed at <https://192.168.12.199:8443>. Bloomberg StoreSafe web management console login page will prompt and request for login credentials.

**BLOOMBASE CONFIDENTIAL**



After login, the Main dashboard page of Bloombase StoreSafe web management console will display all system and server necessary information.

**BLOOMBASE CONFIDENTIAL**

**Bloombase StoreSafe Security Server**

**Greeting**  
 Host Name: StoreSafe199  
 User: admin  
 Datetime: 2017-01-09 01:46:28 -0800

**Menu Bar**

- System
- Operation
- Network Security
- High Availability
- Administration
- Key Management
- StoreSafe Configurations
- Storage

**Language**  
 English

Copyright © 2017 Bloombase

**Main**

**System Information**

Product Name	Bloombase StoreSafe Security Server	Version	3.4.6.19a
Host Name	StoreSafe199 / localhost	System Up Since	2017-01-08 23:48:12 -0800
Host Addresses	1 tun0 10.11.0.82 2 ens32 fe80:0:0:0:250:56ff:fe87:9859, 192.168.1.51, 192.168.12.199		
Licensee	CN=SPFSSF2666 O=Bloombase\ Inc. C=US	Serial Number	9830
Validity	<input checked="" type="checkbox"/>	Perpetuality	<input checked="" type="checkbox"/>

**Server Information**

Operating System	Linux amd64 3.10.0-327.el7.ssfc.x86_64	Processors	1
Memory Utilization	5%	Total Memory	776,667,136
Max Memory	4,151,836,672	Free Memory	537,608,400
Disk Space Utilization	9%	Total Disk Space	40,207,929,344
Used Disk Space	3,732,643,840	Free Disk Space	36,475,285,504

**Application Status**

Application Status

Last Shutdown Time  
 Last Standby Time  
 Last Startup Time 2017-01-08 23:48:22 -0800

**Setting up Integration with KMIP Key Manager – IBM SKLM**

In this interoperability test effort, Bloombase StoreSafe serves as the client of IBM SKLM for encryption key access to deliver data-at-rest encryption services. To enable the built-in Bloombase KeyCastle to utilize the key in the network attached IBM SKLM, KMIP service configuration has to be setup. KMIP service configuration can be accessed from Bloombase StoreSafe web management console, under Menu Bar -> Key Management -> OASIS KMIP Key Manager.

BLOOMBASE CONFIDENTIAL

The screenshot shows the Bloombase StoreSafe Security Server web interface. At the top, there is a navigation bar with links for Main, Logout, Support, About, and Help. Below this is the page title 'Bloombase StoreSafe Security Server'. On the left side, there is a 'Greeting' box with the following information: Host Name: StoreSafe199, User: admin, Datetime: 2017-01-09 01:48:41 -0800. Below the greeting is a 'Menu Bar' with a list of categories: System, Operation, Network Security, High Availability, Administration, and Key Management. Under 'Key Management', there are several sub-items: Bloombase KeyCastle, Hardware Security Module, OASIS KMIP Key Manager, Find Key Wrapper, and Create Key Wrapper. The main content area is titled 'List KMIP Key Manager' and features a table with the following headers: Name, Model, Host Address, and Port. An 'Add' button is located below the table headers. The background of the main content area has a faint watermark of a padlock and a key.

Push “Add” button in order to create a new KMIP trust. Choose “IBM SKLM” model and input IBM SKLM details including IP address and port. For this test, IBM SKLM has been configured to provide KMIP service using IP address 173.193.147.234. IBM SKLM service is trusted by adding the IBM SKLM server certificate to Bloombase StoreSafe’s trust KMIP Server Keystore and Bloombase StoreSafe client key-pair is generated and uploaded to KMIP Client Keystore.

BLOOMBASE CONFIDENTIAL

Main Logout Support About Help

# Bloombase StoreSafe Security Server

### Greeting

Host Name: StoreSafe199  
User: admin  
Datetime: 2017-01-10 20:53:17 -0800

### Menu Bar

- System
- Operation
- Network Security
- High Availability
- Administration
- Key Management
- Bloombase KeyCastle
- Hardware Security Module
- OASIS KMIP Key Manager
- Find Key Wrapper
- Create Key Wrapper
- StoreSafe Configurations
- Storage

### Language

English

Copyright © 2017 Bloombase

## Modify KMIP Key Manager

**Modify KMIP Key Manager**

Name:

Model:

Host Addresses:

Port:

Timeout:  ms

Retry Count:

Retry Wait Time:  ms

Username:

Password:

### Client Keystore

Subject Name: CN=Bloombase  
Serial Number: 2c754aef  
Issuer Name: CN=Bloombase  
Valid Start Date: 2017-01-09  
Valid End Date: 2026-11-18

Client Keystore File:  No file chosen

Pin:

### Trust Certificate

Subject Name: CN=sslserver  
Serial Number: 02be3a5dd622  
Issuer Name: CN=sslserver  
Valid Start Date: 2016-11-20  
Valid End Date: 2023-04-23

Trust Certificate File:  No file chosen

**BLOOMBASE CONFIDENTIAL**

After configuration is complete, push “Test” button to check whether Bloombase StoreSafe and IBM SKLM communication is established. When “Test” button is pushed, IBM SKLM will authenticate Bloombase StoreSafe and Bloombase StoreSafe will send a query to IBM SKLM. If connection test between client-server returns “Success”, IBM SKLM is ready to use as Key Manager for Bloombase StoreSafe.

*Modify KMIP Key Manager*

**Modify KMIP Key Manager**

Name:

Model:

Host Addresses:

Port:

Timeout:  ms

Retry Count:

Retry Wait Time:  ms

Username:

Password:

Test Results :

173.193.147.234 : Success

Push “Submit” button to save the configuration and the configured KMIP server can be viewed under Menu Bar -> Key Management -> OASIS KMIP Key Manager -> Find.

Main Logout Support About Help

**B** Bloombase StoreSafe Security Server

**Greeting**

Host Name: StoreSafe199  
User: admin  
Datetime: 2017-01-10 20:55:34 -0800

**Menu Bar**

System  
Operation

*List KMIP Key Manager*

**List KMIP Key Manager**

	Name	Model	Host Address	Port
1	IBM_SKLM		173.193.147.234	5696

**BLOOMBASE CONFIDENTIAL****KMIP Key Manager High Availability Clustering Configuration**

Considering IBM SKLM high availability clustering with replication, Bloomberg StoreSafe also supports multi-server KMIP configuration. This can be used when there are more than 1 SKLM configured for high availability purpose. In below image, we assume there are three SKLM server; 1 master (173.193.147.234) and 2 clones (173.193.147.235 and 173.193.147.236). Bloomberg StoreSafe will try to communicate with the 1<sup>st</sup> IP which is master (173.193.147.234). If it is failed it will try the next IP which is the 1<sup>st</sup> clone (173.193.147.235). If it happens both master and 1<sup>st</sup> clone are unable to be reached, the 2<sup>nd</sup> clone (173.193.147.236) will take over.

*Modify KMIP Key Manager*

**Modify KMIP Key Manager**

Name	<input type="text" value="IBM_SKLM"/>
Model	<input style="border: none; background-color: #f0f0f0; padding: 2px 5px;" type="text" value="IBM SKLM"/>
Host Addresses	<input type="text" value="173.193.147.234, 173.193.147.235, 173.193.147.236"/>
Port	<input type="text" value="5696"/>
Timeout	<input type="text" value="30000"/> ms
Retry Count	<input type="text" value="1"/>
Retry Wait Time	<input type="text" value="3000"/> ms
Username	<input type="text"/>
Password	<input type="password"/>

**CIFS Secure Storage Configuration**

Physical storage namely 'CIFS01' is configured to be secured by Bloomberg StoreSafe using encryption. Backend Physical Storage can be configured under Menu Bar -> Storage -> Physical Storage.



**BLOOMBASE CONFIDENTIAL**

CIFS backend storage is provisioned by Windows Server 2012 (192.168.100.157) with share, namely CIFS01.

The screenshot shows a web interface titled "Modify Storage Configuration" with two tabs: "Physical Storage" and "Permissions". The "Physical Storage" tab is active, displaying the following configuration details:

Physical Storage Configuration	
Name	CIFS01
Description	
Physical Storage Type	Remote
Type	Common Internet File System (CIFS)
Host	192.168.100.157
Share Name	CIFS01
Read Size	
Write Size	
Mount Hard	<input type="checkbox"/>
User	Administrator
Password	
Options	
Virtual Storage	CIFS01
Owner	admin
Last Update Datetime	2017-01-11 00:37:08 -0800

At the bottom of the form are three buttons: "Submit", "Delete", and "Close". A large, faint watermark of a storage tank and a padlock is visible on the right side of the interface.

Bloombase StoreSafe CIFS secure storage, named CIFS01, is configured to secure "CIFS01" backend physical storage.

**BLOOMBASE CONFIDENTIAL**

### Virtual Storage Status

**Virtual Storage**

Name CIFS01  
Status   
Active   
Type File

**Physical Storage**

Name CIFS01  
Type Remote  
Host 192.168.100.157  
Share CIFS01

**Active Share Status**

Share Name CIFS01  
Storage Type Remote  
Storage Path //192.168.100.157/CIFS01  
Startup Time 2017-01-11 00:55:44 -0800  
Sessions 0 



[Refresh](#) [Stop](#) [Restart](#) [Close](#)

Files under Bloombase StoreSafe CIFS secure storage is encrypted using key `CIFS_Key01` that has been created and managed at IBM SKLM.

**BLOOMBASE CONFIDENTIAL**

### Modify Virtual Storage Handler

Virtual Storage | Protection | Access Control | Permissions

#### Virtual Storage Protection

Protection Type:

#### Encryption Keys

	Key Name	Last Update Datetime
1 <input type="checkbox"/>	CIFS_Key01	2017-01-11 00:55:21 -0800

#### Cryptographic Cipher

Cipher Algorithm:

Bit Length:



### NFS Secure Storage Configuration

Physical storage namely 'NFS01' is configured to be secured by Bloombase StoreSafe using encryption. Backend Physical Storage can be configured under Menu Bar -> Storage -> Physical Storage.

NFS backend storage is provisioned by CentOS 7 (192.168.100.156) with share name /NFS01.


**BLOOMBASE CONFIDENTIAL**

### Modify Storage Configuration

**Physical Storage** | **Permissions**

#### Physical Storage Configuration

Name	<input type="text" value="NFS01"/>
Description	<input type="text"/>
Physical Storage Type	<input type="text" value="Remote"/>
Type	<input type="text" value="Network File System (NFS)"/>
Host	<input type="text" value="192.168.100.156"/>
Share Name	<input type="text" value="/NFS01"/>
Read Size	<input type="text"/>
Write Size	<input type="text"/>
Synchronous	<input type="checkbox"/>
Mount Hard	<input type="checkbox"/>
Options	<input type="text"/>
Virtual Storage	NFS01
Owner	admin
Last Update Datetime	2017-01-11 00:37:29 -0800



Bloombase StoreSafe NFS secure storage, named `NFS01`, is provisioned to secure “`NFS01`” backend physical storage.

**BLOOMBASE CONFIDENTIAL**

### Virtual Storage Status

**Virtual Storage**

Name NFS01  
Status   
Active   
Type File

**Physical Storage**

Name NFS01  
Type Remote  
Host 192.168.100.156  
Share /NFS01

**Active Share Status**

Share Name NFS01  
Storage Type Remote  
Storage Path 192.168.100.156:/NFS01  
Startup Time 2017-01-12 00:32:57 -0800  
Sessions 6 🔍

[Refresh](#) [Stop](#) [Restart](#) [Close](#)



Files under Bloombase StoreSafe NFS secure storage is encrypted using key NFS\_Key01 that has been created and managed at IBM SKLM.

**BLOOMBASE CONFIDENTIAL**

*Modify Virtual Storage Handler*

Virtual Storage   Protection   Access Control   Permissions

**Virtual Storage Protection**

Protection Type

**Encryption Keys**

	Key Name	Last Update Datetime
1 <input type="checkbox"/>	NFS_Key01	2017-01-11 01:00:01 -0800

**Cryptographic Cipher**

Cipher Algorithm

Bit Length

### iSCSI Secure Storage Configuration

Physical storage namely 'iSCSI01' is configured to be secured by Bloombase StoreSafe using encryption. Backend Physical Storage can be configured under Menu Bar -> Storage -> Physical Storage.

iSCSI backend storage is provisioned by OpenFiler (192.168.100.154). To configure the backend iSCSI storage to be encrypted, discover the iSCSI target at OpenFiler and configure the iSCSI backend storage.

**BLOOMBASE CONFIDENTIAL**

### Modify Storage Configuration Discovery

Physical Storage | **iSCSI** | iSCSI Discovery | Permissions

#### iSCSI Discovery

Host 192.168.100.154  
Port 3260

#### Target List

	Target	Portal	Port
1	iqn.2006-01.com.openfiler:tsn.196477af9e05	192.168.100.154	3260

Discover Close

### Modify Storage Configuration

Physical Storage | **Permissions**

#### Physical Storage Configuration

Name

Description

Physical Storage Type

Type

Device ID [max 8 chars]

Options

Device  🔍 🗑️

Virtual Storage

Owner

Last Update Datetime

Submit Delete Close

Bloombase StoreSafe iSCSI secure storage, named `iSCSI01`, is provisioned to secure “`iSCSI01`” backend physical storage.

**BLOOMBASE CONFIDENTIAL**

*Virtual Storage Status*

**Virtual Storage**

Name	iSCSI01
Status	<input checked="" type="checkbox"/>
Active	<input checked="" type="checkbox"/>
Type	iSCSI

**Physical Storage**

Name	iSCSI01
Type	Device
Device	4f504e46494c45527a71345854592d556a4b632d42656e32

**Active Share Status**

Share Name	iSCSI01
Storage Type	Device
Storage Path	Target : iSCSI01 LUN 1:[sdc];
Sessions	1

Refresh Stop Start Resync Close

The screenshot shows a web-based interface for managing storage. It is titled 'Virtual Storage Status'. Under the 'Virtual Storage' section, the name is 'iSCSI01', status is checked, active is checked, and type is 'iSCSI'. The 'Physical Storage' section shows the same name, type 'Device', and a long hexadecimal device ID. The 'Active Share Status' section shows 'Share Name' as 'iSCSI01', 'Storage Type' as 'Device', 'Storage Path' as 'Target : iSCSI01 LUN 1:[sdc];', and 'Sessions' as '1' with a key icon. At the bottom, there are five buttons: 'Refresh', 'Stop', 'Start', 'Resync', and 'Close'. A large, faint watermark of a storage cylinder and a padlock is visible in the background.

Disk contents inside Bloombase StoreSafe iSCSI secure storage is protected using key iSCSI\_Key01 that has been created and managed at IBM SKLM.



**BLOOMBASE CONFIDENTIAL**

*Modify Virtual Storage Handler*

Virtual Storage Protection

Protection Type

Encryption Keys

	Key Name	Last Update Datetime
1 <input type="checkbox"/>	iSCSI_Key01	2017-01-11 01:05:49 -0800

Remove

Cryptographic Cipher

Cipher Algorithm

Bit Length

Submit Close

**FCP Secure Storage Configuration**

Physical storage namely 'FC01' is configured to be secured by Bloombase StoreSafe using encryption. Backend Physical Storage can be configured under Menu Bar -> Storage -> Physical Storage.



FCP backend storage configuration is provisioned by SAN Storage (192.168.100.140).


**BLOOMBASE CONFIDENTIAL**

### Modify Storage Configuration

**Physical Storage** | **Permissions**

#### Physical Storage Configuration

Name	<input type="text" value="FC01"/>
Description	<input type="text"/>
Physical Storage Type	<input type="text" value="Device"/>
Type	<input type="text" value="BLOCKIO"/>
Device ID [max 8 chars]	<input type="text" value="0"/>
Options	<input type="text" value="MPIO"/>
Device	600c0ff000139da524f26e5801000000  
Virtual Storage	FC01
Owner	admin
Last Update Datetime	2017-01-12 12:43:04 +0800



Bloombase StoreSafe FCP secure storage, named FC01, is configured to secure “FC01” backend physical storage.

**BLOOMBASE CONFIDENTIAL**

*Virtual Storage Status*

**Virtual Storage**

Name	FC01
Status	<input checked="" type="checkbox"/>
Active	<input checked="" type="checkbox"/>
Type	FC

**Physical Storage**

Name	FC01
Type	Device
Device	600c0ff000139da524f26e5801000000

**Active Share Status**

Share Name	FC01
Storage Type	Device
Storage Path	Target : 21:00:00:24:ff:54:5f:ca LUN 1:[dm-3]; Target : 21:00:00:24:ff:54:61:ee LUN 1:[dm-3];
Sessions	0 🔑

Refresh Stop Start Resync Close

Disk contents inside Bloombase StoreSafe FCP secure storage is protected using key FC\_Key01 that has been created and managed at IBM SKLM.

BLOOMBASE CONFIDENTIAL

### Modify Virtual Storage Handler

Virtual Storage | Protection | Access Control | Permissions

#### Virtual Storage Protection

Protection Type: Privacy ▼

#### Encryption Keys

	Key Name	Last Update Datetime
1 <input type="checkbox"/>	FC_Key01	2017-01-12 12:43:51 +0800

#### Cryptographic Cipher

Cipher Algorithm: AES XTS ▼  
Bit Length: 256 ▼



BLOOMBASE CONFIDENTIAL

## 6. Interoperability Test Cases

### 6.1 Test Cases for Key Generation

#### 6.1.1 Key Generation at IBM SKLM

Login to the Bloombase StoreSafe web management console and expand the “Key Management” menu. Launch “Create Key Wrapper” tool and change to “Modify Key Source” tab. Select Key Source Type as “OASIS KMIP Key Manager” and configured Key Manager “`IBM_SKLM`”.

When “`IBM_SKLM`” is chosen, Bloombase StoreSafe will send several KMIP operations to list all keys stored in IBM SKLM; including Get & Get Attributes.

**BLOOMBASE CONFIDENTIAL**

*Modify Key Source*

Key Wrapper   **Modify Key Source**   Permissions

**Modify Key Source**

Type   OASIS KMIP Key Manager ▼

**OASIS KMIP Key Manager**

Key Manager   IBM\_SKLM ▼

Object  

Refresh

Submit   Close

Leave the Object as blank and change to “Key Wrapper”. Input key name and push “Generate” button to generate KMIP key object at the configured IBM SKLM cluster. When “Generate” button is pushed, Bloombase StoreSafe will send several KMIP operations required for key generation; including Create, Get, Register, Add Attributes, Get Attributes, Destroy, and Activate.


BLOOMBASE CONFIDENTIAL

### Modify Key Wrapper

Key Wrapper    Modify Key Source    Permissions

**Modify Key Wrapper**

Name	<input type="text" value="CIFS_Key01"/>
Type	Symmetric
Active	<input checked="" type="checkbox"/>
KMIP Key Manager	IBM_SKLM
KMIP UUID	
KMIP Key Name	
KMIP Key State	
Key Bit Length	256 ▼
Owner	admin
Last Update Datetime	




Push “Submit” button to store the key created by Bloombase StoreSafe in IBM SKLM.

### Modify Key Wrapper

Key Wrapper    Modify Key Source    Permissions

**Modify Key Wrapper**

Name	<input type="text" value="CIFS_Key01"/>
Type	Symmetric
Active	<input checked="" type="checkbox"/>
KMIP Key Manager	IBM_SKLM
KMIP UUID	KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e
KMIP Key Name	CIFS_Key01
KMIP Key State	Active
Key Bit Length	256
Owner	admin
Last Update Datetime	



**BLOOMBASE CONFIDENTIAL**

Push “Find” button to list all keys that are ready to use to provide data at-rest encryption. In this test effort, four different symmetric keys (CIFS\_Key01, NFS\_Key01, iSCSI\_Key01, FC\_Key01) are generated by Bloombase StoreSafe and stored in IBM SKLM.

**Find Key Wrapper**

Name:  Type:  Active:  CA:

Find Reset Add

Name	Type	Key Source Type	Active	Status	CA	Subject DN	Issuer DN	Effective Datetime	Expiry Datetime	Last Update Datetime
1 CIFS_Key01	Symmetric	KMIP	<input checked="" type="checkbox"/>			UUID=KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e	KMIP=IBM_SKLM			2017-01-11 00:43:48 -0800
2 FC_Key01	Symmetric	KMIP	<input checked="" type="checkbox"/>			UUID=KEY-58f5efd-4f74585e-3a61-4a01-856b-fe4f80ecf192	KMIP=IBM_SKLM			2017-01-11 00:52:11 -0800
3 NFS_Key01	Symmetric	KMIP	<input checked="" type="checkbox"/>			UUID=KEY-58f5efd-6f4cb622-cd01-403b-b0f2-39c4fe87807e	KMIP=IBM_SKLM			2017-01-11 00:49:21 -0800
4 iSCSI_Key01	Symmetric	KMIP	<input checked="" type="checkbox"/>			UUID=KEY-58f5efd-5b120089-8bde-4af8-a284-bba16a21138a	KMIP=IBM_SKLM			2017-01-11 00:50:36 -0800

### 6.1.2 Key Access at IBM SKLM

Login to the Bloombase StoreSafe web management console and expand the “Key Management” menu. Launch “Create Key Wrapper” tool and change to “Modify Key Source” tab. Select Key Source Type as “OASIS KMIP Key Manager” and configured Key Manager “IBM\_SKLM”. Under object, list of all keys that are stored in IBM SKLM can be found. Choose the key that will be provisioned to Bloombase StoreSafe.



BLOOMBASE CONFIDENTIAL

### Modify Key Source

Key Wrapper   **Modify Key Source**   Permissions

**Modify Key Source**

Type   OASIS KMIP Key Manager

**OASIS KMIP Key Manager**

Key Manager   IBM\_SKLM

Object

- CIFS\_Key01 [KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e]
- BB\_KEY01 [KEY-58f5efd-599455fd-51fc-4cef-a460-9b8d35606571]
- bb\_key02 [KEY-58f5efd-46c34d82-3cc1-403a-a069-d8d6154b63f0]
- Bloombase\_CIFS\_Key [KEY-58f5efd-ea50f8f3-e8a0-4274-95fe-06080747cdfb]
- Bloombase\_NFS\_Key [KEY-58f5efd-a8783256-b62e-4d65-8eeb-e3dd3b7a4d1a]
- Bloombase\_FC\_Key [KEY-58f5efd-31a58f29-0c3b-4b97-a6f2-b8467058cc39]
- Bloombase\_iSCSI\_Key [KEY-58f5efd-2f3ddd6d-92c4-4ab7-9ab7-d354836ffdfc]
- CIFS\_Key01 [KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e]
- NFS\_Key01 [KEY-58f5efd-6f4cb622-cd01-403b-b0f2-39c4fe87807e]
- iSCSI\_Key01 [KEY-58f5efd-5b120089-8bde-4af8-a284-bba16a21138a]
- FC\_Key01 [KEY-58f5efd-4f74585e-3a61-4a01-856b-fe4f80ecf192]

Change to “Key Wrapper” tab after choosing the key, key information will be shown.

### Modify Key Wrapper

Key Wrapper   **Modify Key Source**   Permissions

**Modify Key Wrapper**

Name	CIFS_Key01
Type	Symmetric
Active	<input checked="" type="checkbox"/>
KMIP Key Manager	IBM_SKLM
KMIP UUID	KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e
KMIP Key Name	CIFS_Key01
KMIP Key State	Active
Key Bit Length	256
Owner	admin
Last Update Datetime	

Submit   Close

**BLOOMBASE CONFIDENTIAL**

Push “Submit” button to complete the linkage of key from IBM\_SKLM.

*Modify Key Wrapper*

Key Wrapper    **Modify Key Source**    Permissions

**Modify Key Wrapper**

Name: CIFS\_Key01

Type: Symmetric

Active:

KMIP Key Manager: IBM\_SKLM

KMIP UUID: KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e

KMIP Key Name: CIFS\_Key01

KMIP Key State: Active

Key Bit Length: 256

Owner: admin

Last Update Datetime:

Submit    Close

### 6.1.3 Key Access at IBM SKLM Replication Model Clone Node

Interoperability test between Bloombase StoreSafe and IBM SKLM HA is done by configuring IBM SKLM as Clone. Clone SKLM is limited to read only operation in which key generation is forbidden. In this test, Bloombase StoreSafe will get the key stored in IBM SKLM “Clone” which will be used to encrypt and un-encrypt data.

Login to the Bloombase StoreSafe web management console and expand the “Key Management” menu. Launch “Create Key Wrapper” tool and change to “Modify Key Source” tab. Select Key Source Type as “OASIS KMIP Key Manager” and configured Key Manager “IBM\_SKLM”. Under object, list of all keys that are stored in IBM SKLM can be found. Choose the key that will be provisioned to Bloombase StoreSafe.

BLOOMBASE CONFIDENTIAL

### Modify Key Source

Key Wrapper   **Modify Key Source**   Permissions

**Modify Key Source**

Type   OASIS KMIP Key Manager

**OASIS KMIP Key Manager**

Key Manager   IBM\_SKLM

Object

- CIFS\_Key01 [KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e]
- BB\_KEY01 [KEY-58f5efd-599455fd-51fc-4cef-a460-9b8d35606571]
- bb\_key02 [KEY-58f5efd-46c34d82-3cc1-403a-a069-d8d6154b63f0]
- Bloombase\_CIFS\_Key [KEY-58f5efd-ea50f8f3-e8a0-4274-95fe-06080747cdfb]
- Bloombase\_NFS\_Key [KEY-58f5efd-a8783256-b62e-4d65-8eeb-e3dd3b7a4d1a]
- Bloombase\_FC\_Key [KEY-58f5efd-31a58f29-0c3b-4b97-a6f2-b8467058cc39]
- Bloombase\_iSCSI\_Key [KEY-58f5efd-2f3ddd6d-92c4-4ab7-9ab7-d354836ffdfc]
- CIFS\_Key01 [KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e]
- NFS\_Key01 [KEY-58f5efd-6f4cb622-cd01-403b-b0f2-39c4fe87807e]
- iSCSI\_Key01 [KEY-58f5efd-5b120089-8bde-4af8-a284-bba16a21138a]
- FC\_Key01 [KEY-58f5efd-4f74585e-3a61-4a01-856b-fe4f80ecf192]

Change to “Key Wrapper” tab after choosing the key, key information will be shown.

### Modify Key Wrapper

Key Wrapper   **Modify Key Source**   Permissions

**Modify Key Wrapper**

Name	CIFS_Key01
Type	Symmetric
Active	<input checked="" type="checkbox"/>
KMIP Key Manager	IBM_SKLM
KMIP UUID	KEY-58f5efd-8dac43e5-f18a-4581-99f8-03e9709e390e
KMIP Key Name	CIFS_Key01
KMIP Key State	Active
Key Bit Length	256
Owner	admin
Last Update Datetime	

Submit   Close

**BLOOMBASE CONFIDENTIAL**

Push “Submit” button to complete the linkage of key from IBM\_SKLM.

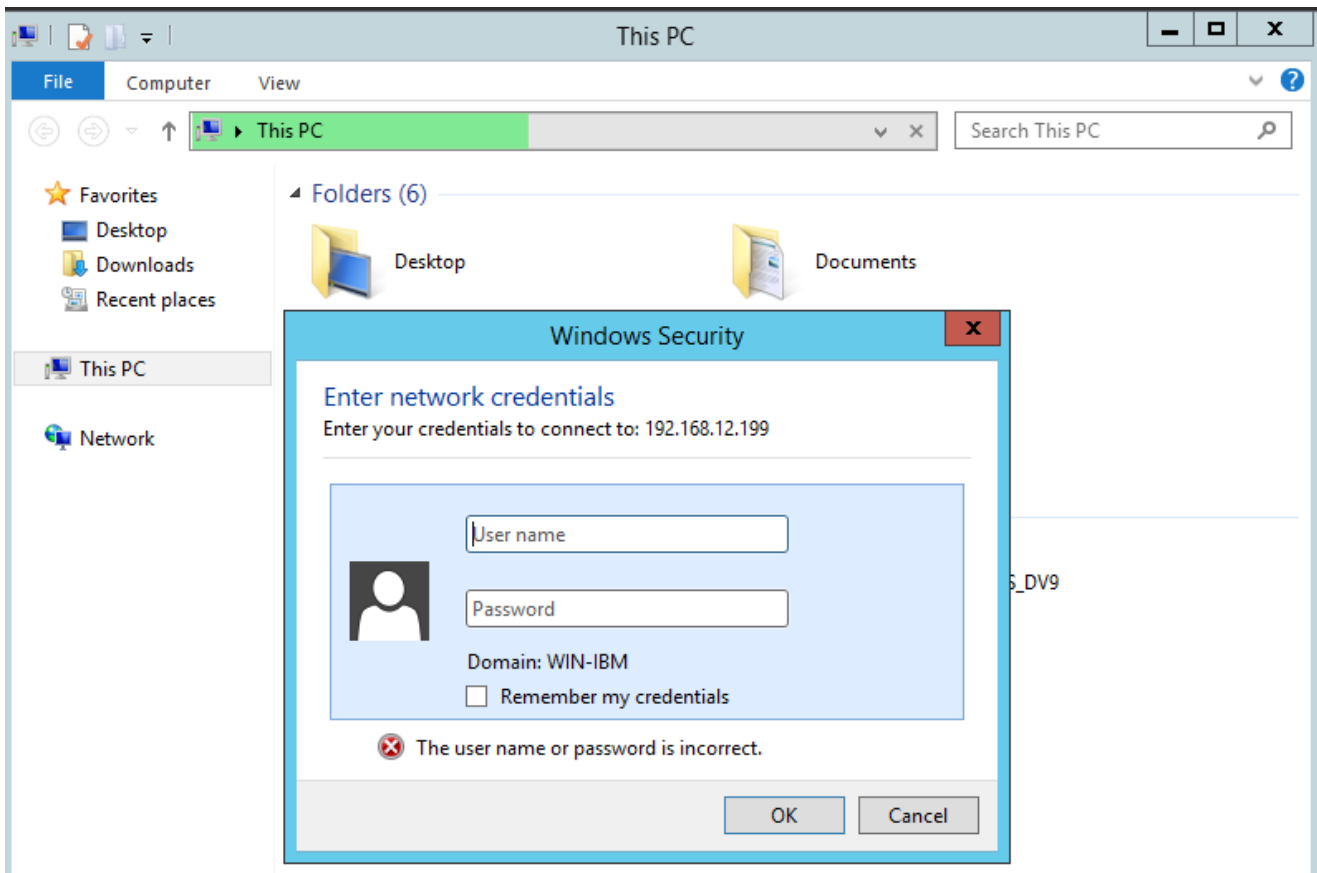
Interoperability test between Bloombase StoreSafe and “Clone” in IBM SKLM replication model maintain the “read-only” characteristics of “Clone” in IBM SKLM replication model. As a client, Bloombase StoreSafe is able to get and use the key which is stored in IBM SKLM to encrypt and un-encrypt data, but will not be able to make changes to existing objects in the clone IBM SKLM.

## 6.2 Test Cases for CIFS Data Encryption and Decryption

### 6.2.1 SMB Authentication at CIFS Secure Storage

Login to the CIFS secure storage delivered by Bloombase StoreSafe by accessing \\192.168.12.199 using Windows File Explorer on Microsoft Windows 7 Client. A dialog box will be displayed and prompts for credentials for user authentication.

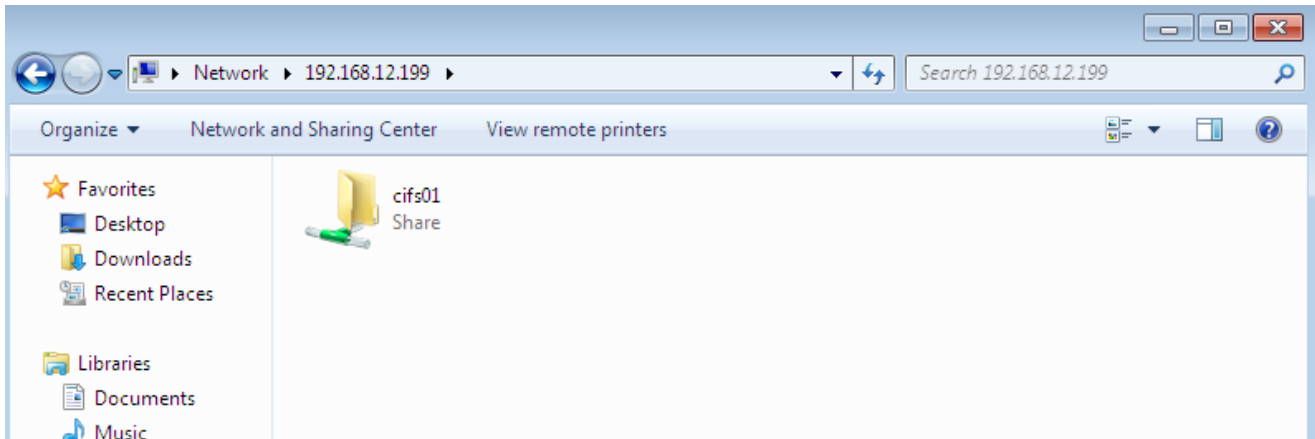
**BLOOMBASE CONFIDENTIAL**



### 6.2.2 List CIFS Secure Storage

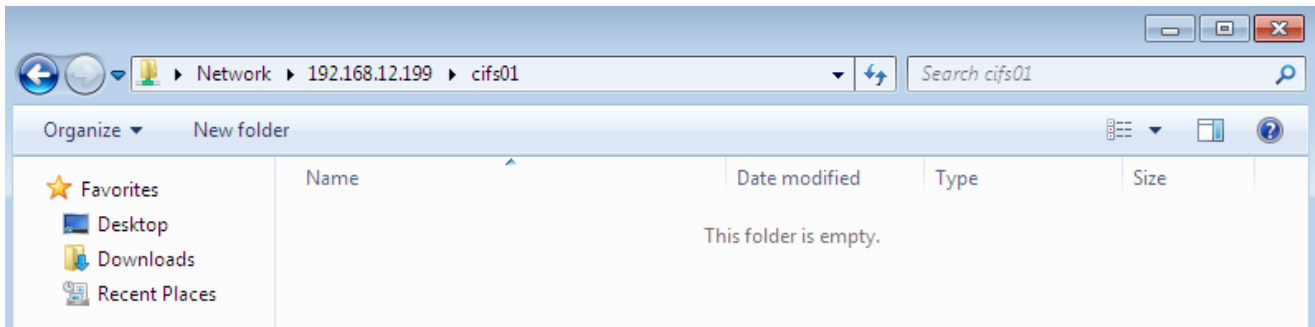
Once user successfully signs on to Bloombase StoreSafe as an authorized SMB user, the available Bloombase StoreSafe CIFS secure storage will be listed.

**BLOOMBASE CONFIDENTIAL**



### 6.2.3 Connect and Access CIFS Secure Storage

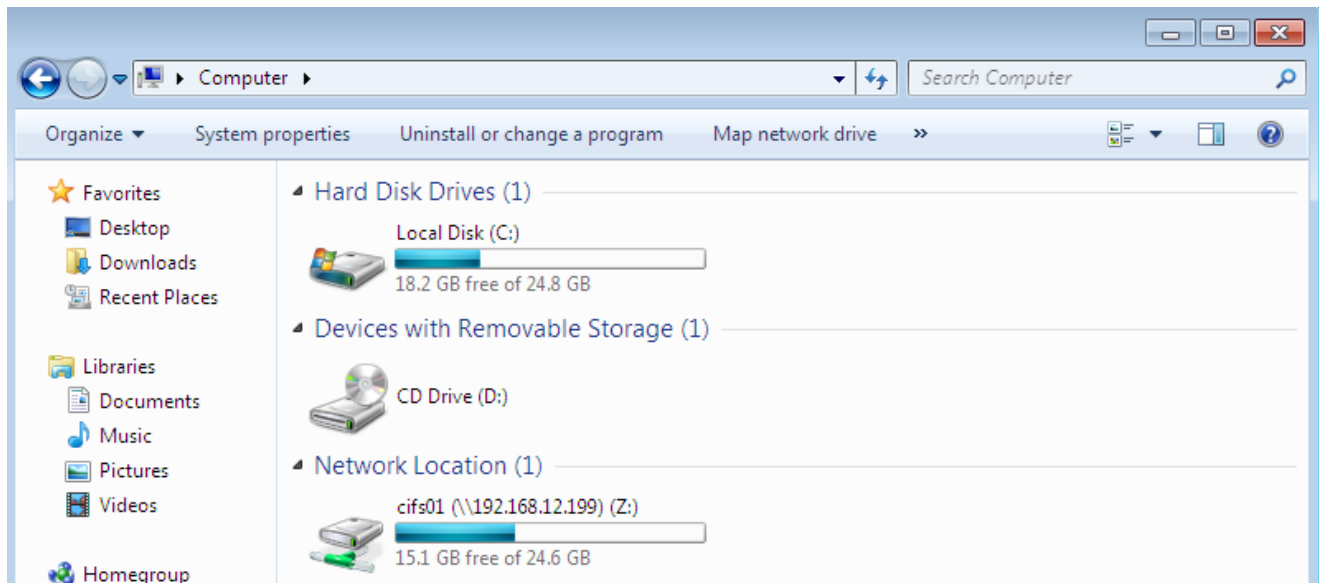
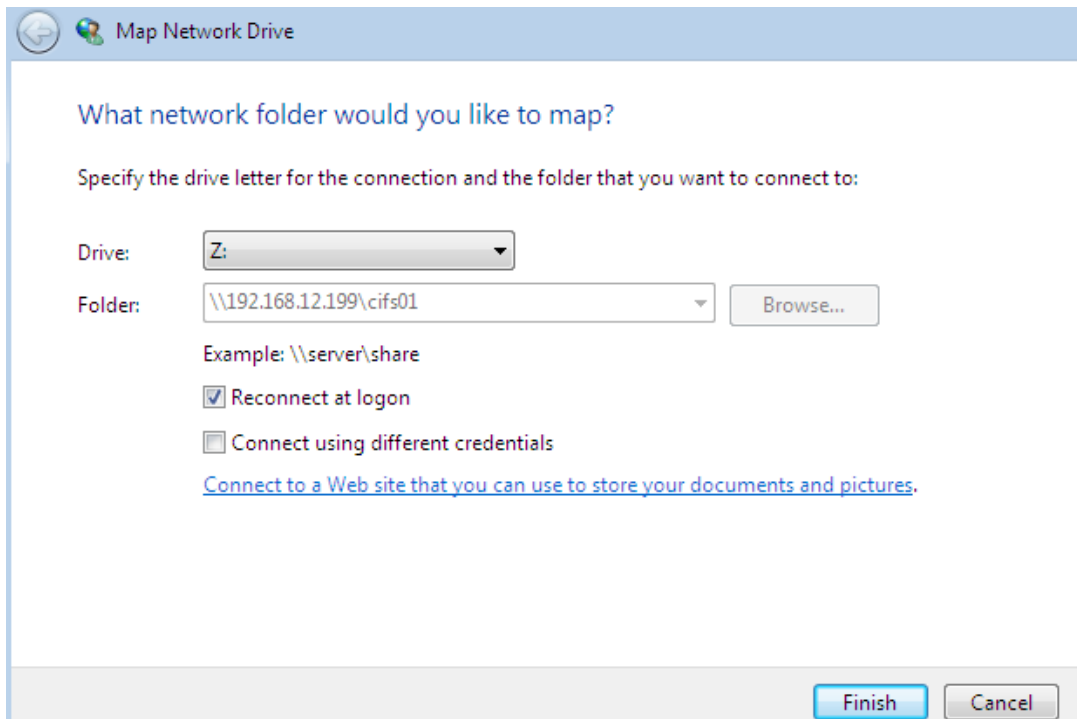
Authorized SMB user is able to list and access contents inside Bloombase StoreSafe CIFS secure storage.



### 6.2.4 Mount CIFS Secure Storage

Mount Bloombase StoreSafe CIFS secure storage by using File Explorer, Map Network Drive wizard will be shown as it is supposed to be mounting any normal network storage. Choose drive letter and push “Finish” button, Bloombase StoreSafe CIFS secure storage has been successfully mapped as a network drive.

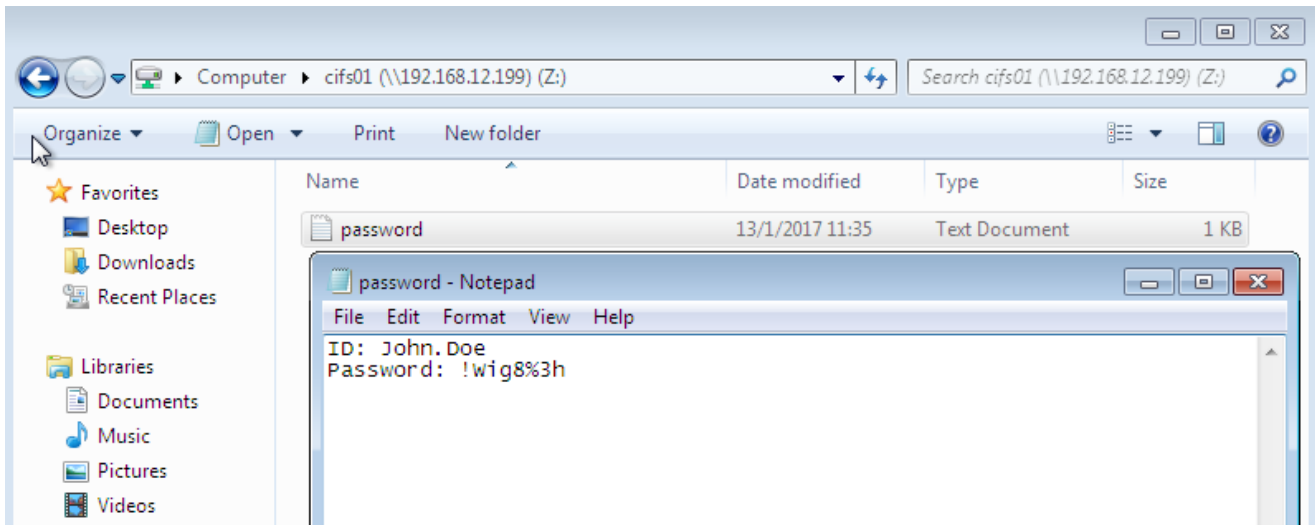
**BLOOMBASE CONFIDENTIAL**



**BLOOMBASE CONFIDENTIAL**

### 6.2.5 Write to CIFS Secure Storage

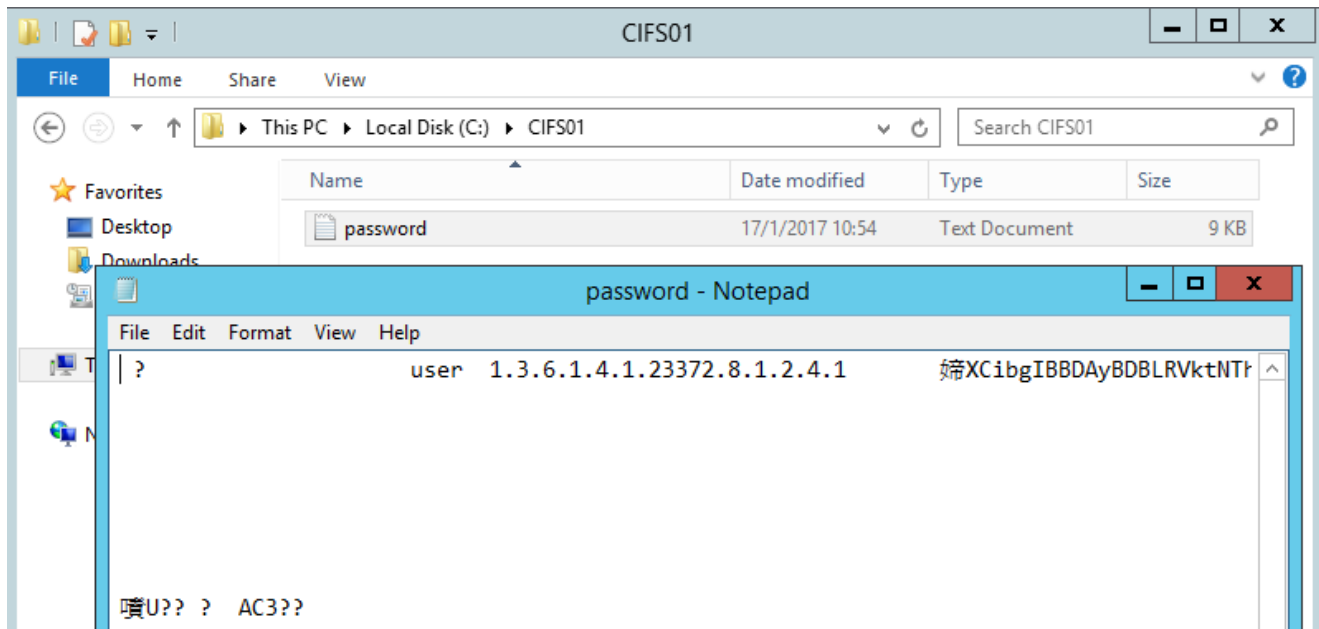
Create a new text document, named “password.txt”, inside Bloombase StoreSafe CIFS secure storage.



Once the file is saved, it is encrypted on the fly and stored physically at backend storage in natural ciphertext format. File is encrypted by key `CIFS_Key01` that has been created and managed at IBM SKLM.



**BLOOMBASE CONFIDENTIAL**



**6.2.6 Read from CIFS Secure Storage**

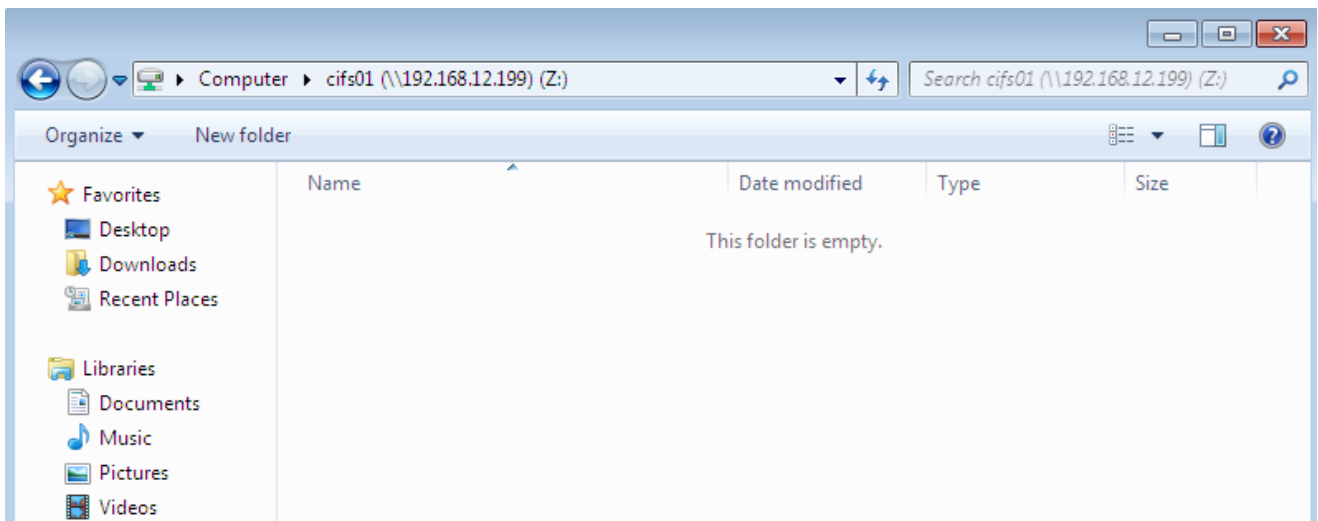
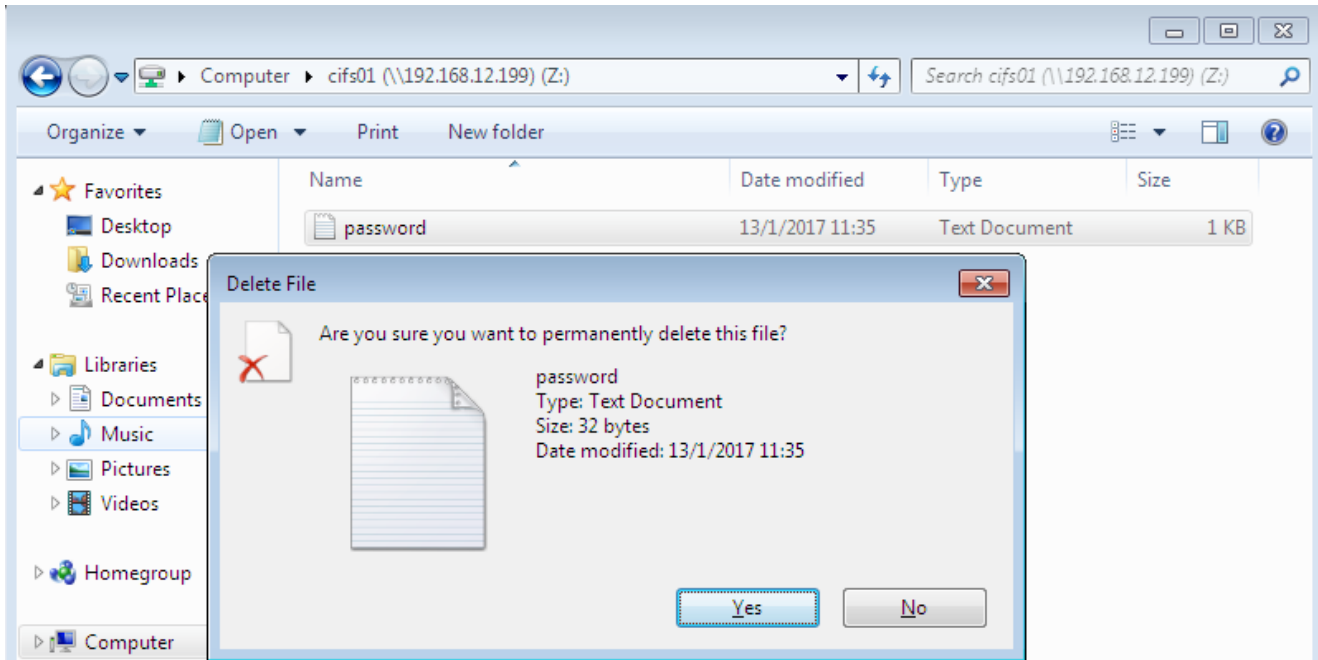
Read the newly created file in Bloombase StoreSafe CIFS secure storage, cipher-text file is retrieved physically from backend storage and un-encrypted on the fly. File is successfully presented and read as virtual clear-text data. File is un-encrypted using key `CIFS_Key01` that has been created and managed at IBM SKLM.

```
Z:\>type password.txt
ID: John.Doe
Password: !Wig8x3h
```

**6.2.7 Delete from CIFS Secure Storage**

Deleting file from Bloombase StoreSafe CIFS secure storage as if normal virtual-plain file. Delete file on Bloombase StoreSafe CIFS secure storage will automatically delete file at backend storage.

**BLOOMBASE CONFIDENTIAL**



## BLOOMBASE CONFIDENTIAL

## 6.3 Test Cases for NFS Data Encryption and Decryption

### 6.3.1 Network Access Control at NFS Secure Storage

Mount Bloomberg StoreSafe NFS secure storage on CentOS 7 client. Authorized client is able to access Bloomberg StoreSafe NFS secure storage without returning any access denied error message.

```
[root@localhost ~]# mount -t nfs 192.168.12.199:/NFS01 /mnt/NFS01
[root@localhost ~]#
```

### 6.3.2 List NFS Secure Storage

List all contents of the mount point which mounted to Bloomberg StoreSafe NFS secure storage.

```
[root@localhost ~]# ls -l /mnt/NFS01
total 0
```

### 6.3.3 Connect and Access NFS Secure Storage

Authorized client is able to access Bloomberg StoreSafe NFS secure storage without returning any access denied or error message.

```
[root@localhost ~]# mount | grep NFS01
192.168.12.199:/NFS01 on /mnt/NFS01 type nfs (rw,relatime,vers=3,rsize=65536,wsiz=65536,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,mountaddr=192.168.12.199,mountvers=3,mountport=20048,mountproto=udp,local_lock=none,addr=192.168.12.199)
[root@localhost ~]# cd /mnt/NFS01/
[root@localhost NFS01]#
```

### 6.3.4 Mount NFS Secure Storage

Client host with authorized IP address is able to get Bloomberg StoreSafe NFS secure storage mounted as a mount point which operates almost like a filesystem.

BLOOMBASE CONFIDENTIAL

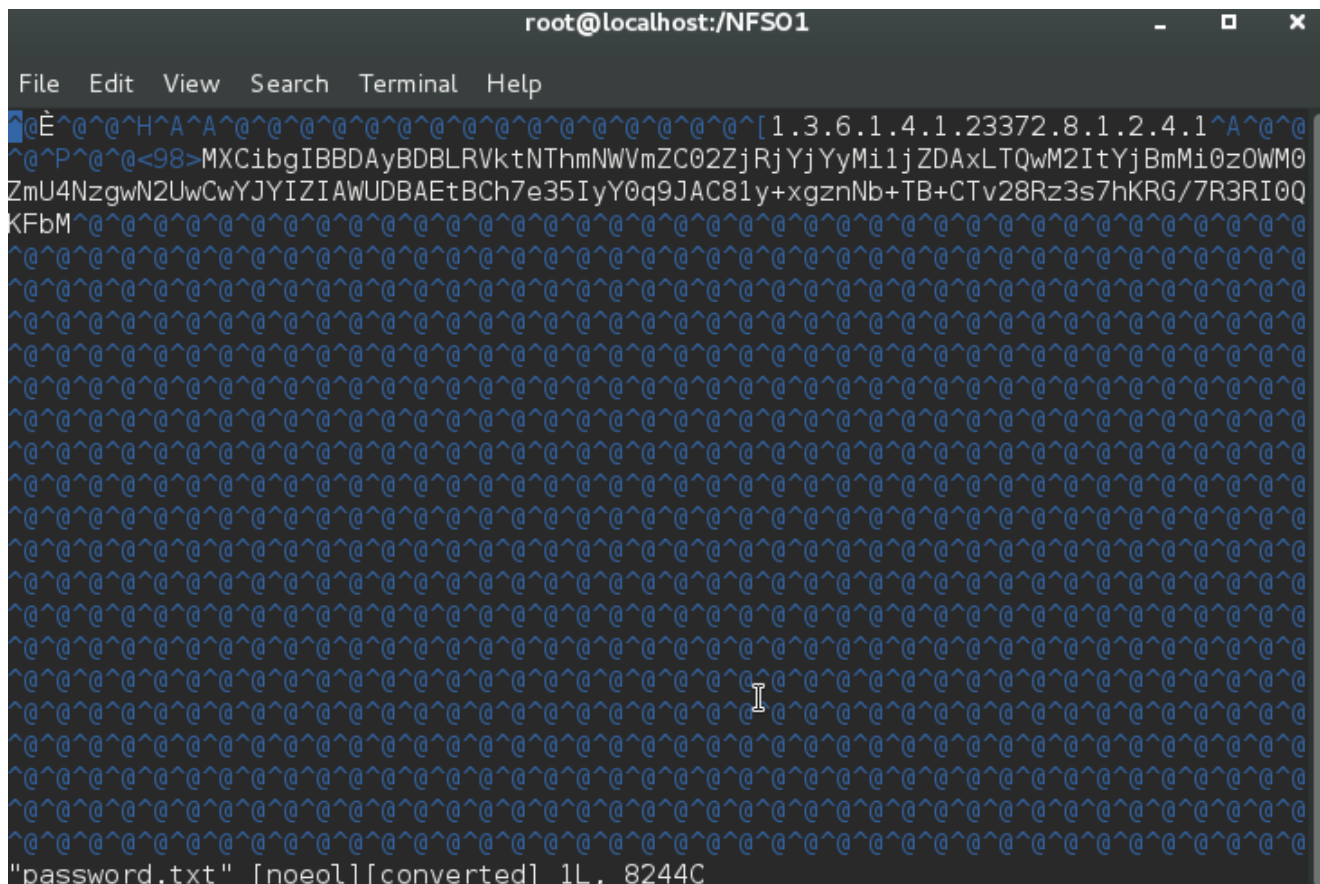
```
[root@localhost ~]# mount -t nfs 192.168.12.199:/NFS01 /mnt/NFS01
[root@localhost ~]# mount | grep NFS01
192.168.12.199:/NFS01 on /mnt/NFS01 type nfs (rw,relatime,vers=3,rsize=65536,wsz=65536,namlen=255,hard,proto=tcp,timeo=600,retr=2,sec=sys,mountaddr=192.168.12.199,mountvers=3,mountport=20048,mountproto=udp,local_lock=none,addr=192.168.12.199)
```

### 6.3.5 Write to NFS Secure Storage

Create a new text document, named “password.txt”, in the mount-point referencing Bloombase StoreSafe NFS secure storage.

```
[root@localhost ~]# echo "John.Doe u&G43tg%" > /mnt/NFS01/password.txt
[root@localhost ~]# ls /mnt/NFS01/password.txt
/mnt/NFS01/password.txt
```

File is encrypted on the fly and stored physically at the backend storage. File is encrypted using key NFS\_Key01 that has been created and managed at IBM SKLM.



**BLOOMBASE CONFIDENTIAL**

### 6.3.6 Read from NFS Secure Storage

Read the newly created file from mount-point referencing to Bloomberg StoreSafe NFS secure storage, cipher-text file is retrieved physically from backend storage and un-encrypted on the fly. File is successfully presented and read as virtual clear-text data. File is un-encrypted using key `NFS_Key01` that has been created and managed at IBM SKLM.

```
[root@localhost ~]# cat /mnt/NFS01/password.txt
John.Doe u&G43tg%
```

### 6.3.7 Delete from NFS Secure Storage

Delete file from mount-point referencing to Bloomberg StoreSafe NFS secure storage as if normal virtual-plain file. Deleting file from mount-point referencing to Bloomberg StoreSafe NFS secure storage will automatically delete file at backend storage.

```
[root@localhost ~]# rm /mnt/NFS01/password.txt
rm: remove regular file '/mnt/NFS01/password.txt'? y
[root@localhost ~]# ls -l /mnt/NFS01/
total 0
```

## 6.4 Test Cases for iSCSI Data Encryption and Decryption

### 6.4.1 Discover iSCSI Secure Storage

Discover Bloomberg StoreSafe iSCSI secure storage on CentOS 7 with authorized IP network address returns discovery and listing of Bloomberg StoreSafe iSCSI secure storage.

```
[root@localhost ~]# iscsiadm -m discovery -tst -p 192.168.12.199:3260
192.168.12.199:3260,1 iSCSI01
```

**BLOOMBASE CONFIDENTIAL**

### 6.4.2 Connect to iSCSI Secure Storage

Authorized client is able to connect and access Bloombase StoreSafe iSCSI secure storage without any permission denied error message.

```
[root@localhost ~]# iscsiadm -m node -p 192.168.12.199:3260 -T iSCSI01 --login
Logging in to [iface: default, target: iSCSI01, portal: 192.168.12.199,3260] (multiple)
Login to [iface: default, target: iSCSI01, portal: 192.168.12.199,3260] successful.
```

### 6.4.3 Dump in iSCSI Secure Storage Raw Device

Write and dump in contents to Bloombase StoreSafe iSCSI secure storage as raw device which gets encrypted and persisted physically at backend storage.

```
[root@localhost dev]# dd if=/dev/zero of=/dev/sdd bs=4k count=2
2+0 records in
2+0 records out
8192 bytes (8.2 kB) copied, 0.00936727 s, 875 kB/s
```

Backend volume is encrypted by using key `iSCSI_Key01` that has been created and managed at IBM SKLM.

```
00000a00 29 bd 7d b6 7c ba 11 f6 69 62 f2 2d 40 1a 07 4a |).|.|.ib.-@..J|
00000a10 bb 4e 72 0d c6 33 ef 02 8b 1b 3d 64 77 66 57 a5 |.Nr..3....=dwfW.|
00000a20 65 d5 44 cb 2e 2f cf 6a 53 a0 23 8d 70 b9 28 f3 |e.D../.jS.#.p.(.|
00000a30 40 10 d6 46 4a df 6d 56 44 f3 f1 83 b1 c1 67 68 |@..FJ.mVD.....gh|
00000a40 02 54 a5 e1 89 b7 02 ad c5 30 26 ea fe 40 e7 04 |.T.....0&...@..|
00000a50 7e 2c f4 f9 a4 2f 8a 97 56 ac c0 26 90 6a 39 a5 |~,.../.V..&.j9.|
00000a60 13 ff 87 ca 06 51 be c5 5f f0 d1 a8 99 8c 08 a1 |.....Q.....|
00000a70 ce 76 71 cd 76 d6 7f 44 7b 43 3c 98 77 58 13 93 |.vq.v..D{C<.wX..|
00000a80 4d 09 7e f0 cb ae d4 32 cd 49 df 89 7a 0e 45 5e |M.~....2.I..z.E^|
00000a90 b0 e7 d8 07 4f 9c 86 8f 12 32 0f 3c 10 95 9e a7 |....0....2.<....|
00000aa0 28 13 3c 6c e4 d2 e1 4a 60 6c a7 86 3a 8b ba a1 |(<l...J`l...:..|
00000ab0 1f a6 1c fc 02 d9 72 8a af db 1a d3 a2 1c e3 17 |.....r.....|
00000ac0 33 35 eb 81 59 31 7c 5a 7b 48 e8 f4 7d a7 ca 4a |35..Y1|Z{H..}..J|
00000ad0 02 1a d9 09 58 f5 75 a9 2f 55 3f f5 b5 b2 86 13 |....X.u./U?.....|
00000ae0 ac 3f 39 e4 db 12 d6 b0 28 32 29 cd 04 f2 9f 9f |.?9.....(2).....|
00000af0 b0 88 01 21 2b da 39 c2 8c de 76 12 4b d8 ce 93 |...!+.9...v.K...|
00000b00 3c 7d e4 d5 bc 1d 9c 9c 5d 65 e1 54 e9 2a e1 e2 |<}.....]e.T.*..|
```

## BLOOMBASE CONFIDENTIAL

#### 6.4.4 Dump out iSCSI Secure Storage Raw Device

Dump out contents from Bloomberg StoreSafe iSCSI secure storage as raw storage device which gets un-encrypted on the fly and retrieved physically from backend storage.

```
[root@localhost dev]# dd if=/dev/sdd of=/var/tmp/images.tar.gz bs=4k count=2  
2+0 records in  
2+0 records out  
8192 bytes (8.2 kB) copied, 0.501861 s, 16.3 kB/s
```

#### 6.4.5 Create Filesystem on iSCSI Secure Storage

Create partition on mounted Bloomberg StoreSafe iSCSI secure storage and create filesystem.

```
[root@localhost dev]# fdisk /dev/sdd  
Welcome to fdisk (util-linux 2.23.2).  
  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.  
  
Device does not contain a recognized partition table  
Building a new DOS disklabel with disk identifier 0xaf93092a.  
  
Command (m for help): n  
Partition type:  
   p   primary (0 primary, 0 extended, 4 free)  
   e   extended  
Select (default p): p  
Partition number (1-4, default 1): 1  
First sector (2048-108003327, default 2048):  
Using default value 2048  
Last sector, +sectors or +size{K,M,G} (2048-108003327, default 108003327):  
Using default value 108003327  
Partition 1 of type Linux and of size 51.5 GiB is set  
  
Command (m for help): w  
The partition table has been altered!  
  
Calling ioctl() to re-read partition table.  
Syncing disks.
```

**BLOOMBASE CONFIDENTIAL**

```
[root@localhost dev]# mkfs.ext3 /dev/sdd1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=128 blocks
3375104 inodes, 13500160 blocks
675008 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
412 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

#### 6.4.6 Write to Filesystem on iSCSI Secure Storage

Create a new text document, named `password.txt`, in the mount-point referencing Bloomberg StoreSafe iSCSI secure storage. File is encrypted on the fly and stored physically at the backend storage. File is encrypted using key `iSCSI_Key01` that has been created and managed at IBM SKLM.

```
[root@localhost ~]# echo "John.Doe u&G43tg%" > /mnt/iSCSI01/password.txt
[root@localhost ~]# ls /mnt/iSCSI01/password.txt
/mnt/iSCSI01/password.txt
```

#### 6.4.7 Read from Filesystem on iSCSI Secure Storage

Read the newly created file from mount-point referencing to Bloomberg StoreSafe iSCSI secure storage, cipher-text file is retrieved physically from backend storage and un-encrypted on the fly. File is successfully presented and read as virtual clear-text data. File is un-encrypted using key `iSCSI_Key01` that has been created and managed at IBM SKLM.

```
[root@localhost ~]# cat /mnt/iSCSI01/password.txt
John.Doe u&G43tg%
```



## BLOOMBASE CONFIDENTIAL

### 6.4.8 Delete from Filesystem on iSCSI Secure Storage

Delete file from mount-point referencing to Bloomberg StoreSafe iSCSI secure storage as if normal virtual-plain file. Deleting file from mount-point referencing to Bloomberg StoreSafe iSCSI secure storage will automatically delete file at backend storage.

```
[root@localhost ~]# rm /mnt/iSCSI01/password.txt
rm: remove regular file '/mnt/iSCSI01/password.txt'? y
[root@localhost ~]# ls -l /mnt/iSCSI01
total 16
drwx----- 2 root root 16384 Jan 12 16:55 lost+found
```

## 6.5 Test Cases for FCP Data Encryption and Decryption

### 6.5.1 Initialize FCP Secure Storage Disk

Client with authorized HBA WWNs and proper SAN switch zoning will be able to access Bloomberg StoreSafe FCP secure storage as a disk and initialize the disk.

```
Disk /dev/sde: 50.0 GB, 49999970304 bytes, 97656192 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 524288 bytes
Disk label type: dos
Disk identifier: 0xa22ec068
```

### 6.5.2 Dump in FCP Secure Storage Raw Device

Write and dump in contents to Bloomberg StoreSafe FCP secure storage as raw device which gets encrypted and persisted physically at backend storage.

```
[root@bb008 dev]# dd if=/dev/zero of=/dev/sde bs=4k count=2
2+0 records in
2+0 records out
8192 bytes (8.2 kB) copied, 0.000276755 s, 29.6 MB/s
```

**BLOOMBASE CONFIDENTIAL**

Backend volume is encrypted by using key FC\_Key01 that has been created and managed at IBM SKLM.

```

000075e0 0c 00 cf 68 3e 81 48 46 09 18 86 a5 22 1f 39 52 |...h>.HF....".9R|
000075f0 14 d2 a4 cd 4a 29 a5 e2 1a 58 11 8f b3 c6 d4 10 |...J)...X.....|
00007600 2f ee f6 4a 90 c5 10 aa 4b 8d b5 67 7a de bb 15 |/..J...K..gz...|
00007610 6d 67 4b e8 6f af 9b 40 17 57 b8 f0 ef 42 37 f2 |mgK.o..@.W...B7.|
00007620 9c 0e 75 34 95 f6 72 d3 10 88 8e f8 a1 96 3f 57 |..u4..r.....?W|
00007630 34 f1 d3 96 d6 c5 93 34 49 50 51 0f 6c 3e 22 82 |4.....4IPQ.l>".|
00007640 25 dd 29 ba ff ff 64 39 40 14 9a ed aa 8f ef 38 |%.)...d9@.....8|
00007650 4f fe 01 2a 78 e8 79 b0 95 7f fd 53 bd 6c aa 36 |O..*x.y....S.l.6|
00007660 bd c9 26 56 bf a3 15 11 24 da b8 fa b9 2e 77 f0 |..&V....$.....w.|
00007670 3e f5 f4 87 75 c0 12 a4 16 de 7e 96 51 fc 08 49 |>...u.....~.Q..I|
00007680 2b 2c bb 87 bf f2 5c b3 04 d8 b0 fd fa d5 f7 ae |+,...\.....|
00007690 ff a0 af 0f d2 72 f8 24 92 75 bb 1c d1 21 b8 41 |....r.$,u...!.A|
000076a0 ce 61 0c b3 34 e3 e4 5a d1 9c b4 f5 bd 6a 1b 6e |.a..4..Z.....j.n|
000076b0 21 d9 f9 cc d1 35 17 55 1a 25 19 02 79 3a 95 27 |!....5.U.%.y:.'|
000076c0 f3 8b 93 94 c5 d1 27 0b 4e 67 d5 5e 30 88 e5 51 |.....'.Ng.^0..Q|
000076d0 a1 a4 2d fb 15 a8 eb b5 59 85 ac 6c aa dd 8a b6 |..-.....Y..l....|
000076e0 52 38 92 5e 57 0b 16 16 68 2e 72 44 c5 a9 0a 3d |R8.^W...h.rD...=|
000076f0 4c 68 0e a3 dc af 65 b1 5d 26 fa 91 74 b1 69 da |Lh....e.]&..t.i.|
00007700 c8 c0 2a 7f 4b f2 66 a2 59 fb 4b 50 e2 3f be 0e |..*.K.f.Y.KP?...|
00007710 ab 9b 4f 9a 7e 8d 7e 91 f0 48 bb b4 1a 79 5c 62 |..O.~.~..H...y\b|
00007720 1a 33 c3 15 3c 9b 9b 79 0f 59 48 19 2c 77 61 30 |.3..<..y.YH.,wa0|
00007730 50 b4 36 97 e7 f0 74 d9 ca 90 d9 41 68 84 2e df |P.6...t....Ah...|
00007740 30 05 a8 9e 1d 20 38 64 9b d7 b3 3e 71 f7 aa 4d |0.... 8d...>q..M|
00007750 14 70 67 bf 16 e7 8c b6 81 ee b2 ec aa 0b f3 3d |.pg.....=|
    
```

### 6.5.3 Dump out FCP Secure Storage Raw Device

Dump out contents from Bloombase StoreSafe FCP secure storage as raw storage device which gets un-encrypted on the fly and retrieved physically from backend storage.

```

[root@bb008 /]# dd if=/dev/sde of=/var/tmp/image.tar.gz bs=4k count=2
2+0 records in
2+0 records out
8192 bytes (8.2 kB) copied, 0.00133511 s, 6.1 MB/s
    
```

### 6.5.4 Create Filesystem on FCP Secure Storage

Create partition on mounted Bloombase StoreSafe FCP secure storage and create filesystem.

## BLOOMBASE CONFIDENTIAL

```
[root@bb008 /]# fdisk /dev/sde
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xec140678.

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-97656191, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-97656191, default 97656191):
Using default value 97656191
Partition 1 of type Linux and of size 46.6 GiB is set

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

## BLOOMBASE CONFIDENTIAL

```
[root@bb008 /]# mkfs.ext3 /dev/sde1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=128 blocks
3055616 inodes, 12206768 blocks
610338 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
373 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

### 6.5.5 Write to Filesystem on FCP Secure Storage

Create a new text document, named `password.txt`, in the mount-point referencing Bloomberg StoreSafe FCP secure storage. File is encrypted on the fly and stored physically at the backend storage. File is encrypted using key `FC_Key01` that has been created and managed at IBM SKLM.

```
[root@bb008 /]# mkdir /mnt/FC01
[root@bb008 /]# mount /dev/sde1 /mnt/FC01
[root@bb008 /]# mount | grep FC01
/dev/sde1 on /mnt/FC01 type ext3 (rw,relatime,errors=continue,user_xattr,barrier=1,data=ordered)
[root@bb008 /]# echo "John.Doe I43%t^r2" > /mnt/FC01/password.txt
[root@bb008 /]# ls -l /mnt/FC01/password.txt
-rw-r--r-- 1 root root 18 Jan 12 16:08 /mnt/FC01/password.txt
```

### 6.5.6 Read from Filesystem on FCP Secure Storage

Read the newly created file from mount-point referencing Bloomberg StoreSafe FCP secure storage, cipher-text file is retrieved physically from backend storage and un-encrypted on the fly. File is

**BLOOMBASE CONFIDENTIAL**

successfully presented and read as virtual clear-text data. File is un-encrypted using key `FC_Key01` that has been created and managed at IBM SKLM.

```
[root@bb008 /]# cat /mnt/FC01/password.txt
John.Doe I43%t^r2
```

### 6.5.7 Delete from Filesystem on FCP Secure Storage

Delete file from mount-point referencing Bloomberg StoreSafe FCP secure storage as if normal virtual-plain file. Deleting file from mount-point referencing Bloomberg StoreSafe FCP secure storage will automatically delete file at backend storage.

```
[root@bb008 /]# rm /mnt/FC01/password.txt
rm: remove regular file '/mnt/FC01/password.txt'? y
[root@bb008 /]# ls -l /mnt/FC01/
total 16
drwx----- 2 root root 16384 Jan 12 16:06 lost+found
```

BLOOMBASE CONFIDENTIAL

## **7. Interoperability and Certification Confirmation**

By completing this Exit Form, IBM confirms that Bloombase StoreSafe will be validated by the typical QA process employed by IBM solution testing procedure and that issues found with the solution will be reported to Bloombase.

BLOOMBASE CONFIDENTIAL

## 8. References

Bloombase StoreSafe, <http://www.bloombase.com/products/storesafe/>

IBM Security Key Lifecycle Manager, <http://www-03.ibm.com/software/products/en/key-lifecycle-manager>

IBM SoftLayer, <http://www.softlayer.com/>

IBM Bluemix, <https://www.ibm.com/cloud-computing/bluemix/>

Brocade 300 SAN Switch, <http://www.brocade.com/en/products-services/storage-networking/fibre-channel/300-switch.html>

HPE 1920-48G Ethernet Switch, <https://www.hpe.com/us/en/product-catalog/networking/networking-switches/pip.switches.7399514.html>

QLogic QLE2672, <http://www.qlogic.com/Products/adapters/Pages/FibreChannelAdapters.aspx>

**BLOOMBASE CONFIDENTIAL**

Red Hat Enterprise Linux, <http://www.redhat.com/en/technologies/linux-platforms/enterprise-linux>