



Enterprise Infrastructure for the Cloud

ezNcrypt

v2.1

Gazzang ezNcrypt™ User Guide

Copyright©2011 by Gazzang, Inc.

INTRODUCTION

Gazzang ezNcrypt is for Linux users who want to implement an “at rest” data encryption solution and store their encryption keys in Gazzang’s Key Storage System (KSS). The Flex and Database Editions, by default, use Gazzang’s high availability KSS servers, while the Enterprise Edition lets you set up your own KSS server.

CONTENTS

Introduction	2
Installation	3
Access Control Management	9
Using the Service	11
Executing Scripts	15
Changing the Encryption Key	16
Utilities	17
Uninstalling	18
ezNcrypt Settings.....	19

SYSTEM REQUIREMENTS

- Linux kernel 2.6.19 or higher*
- Supported Linux Systems:
 - CentOS
 - Red Hat
 - Ubuntu
 - Debian
 - Fedora
 - openSUSE
 - Slackware
 - CloudLinux
 - Scientific Linux
- MySQL Server 4.x or 5.x
- eCryptfs module (bundled with supported kernels)
- keyutils
- eCryptfs-utils

*Red Hat & CentOS can use 2.6.18-92 or higher

While this product has been extensively tested, every environment is different, and it is strongly suggested that you thoroughly test ezNcrypt before installing it into your production environment.

Please read this entire guide before installing ezNcrypt.

Troubleshooting	21
Appendix A: MySQL	22
Storage Engine Differences.....	22
InnoDB Support	22
Accessing the ezNcrypt Directory	22
MySQL Log Encryption	23
Appendix B: Apache	24
Apache Docs Encryption.....	24
License Agreement.....	25

INSTALLATION

1. Download the ezNcrypt software

Using your internet browser, download the ezNcrypt software from the Gazzang download site: www.gazzang.com/download.

The ezNcrypt software is distributed as a **.run** file in the following format: **ezncrypt-2.1-linux-installer.run** (for Linux 32bit). You must download the file that corresponds to your architecture.

2. Run the installer script

You can either run the interactive installer (default option) or the automated installer (see page 7).

Interactive Installer

To start the interactive installer, run the installation script with the appropriate privileges:

```
su -  
#(insert root password)  
./ezncrypt-2.1-linux-installer.run
```

Or (depending on your distro):

```
sudo ./ezncrypt-2.1-linux-installer.run
```

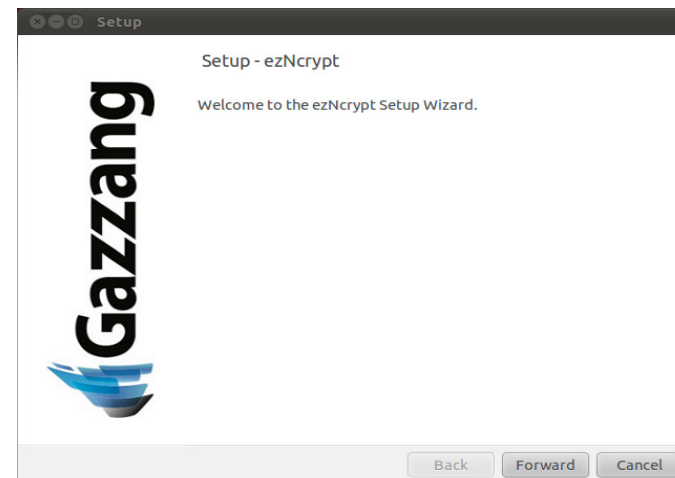
This section displays interactive installation using a graphical installation wizard. You must have a recent version of **Linux dialog** installed to use this method.

If **dialog** is not installed, the interactive installer uses the following conventions:

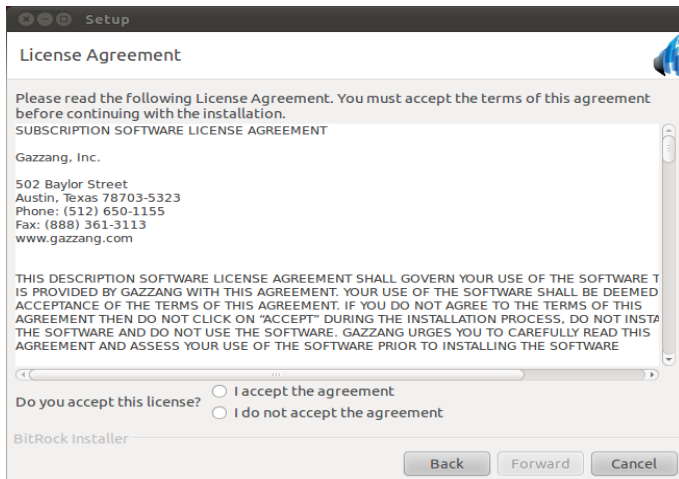
- When presented with a **yes** or **no** prompt, the default option is capitalized.
- When a response must be typed, the default response is surrounded by [square brackets]. This response will be used if you do not provide a response.

Note: You can abort the installation at any time by pressing **Ctrl-C**.

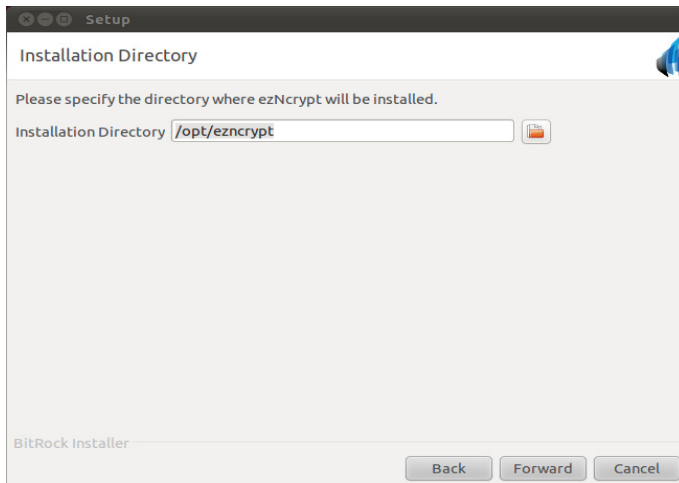
The installer starts the validation process. If your system passes, you will be welcomed to the installation of ezNcrypt.



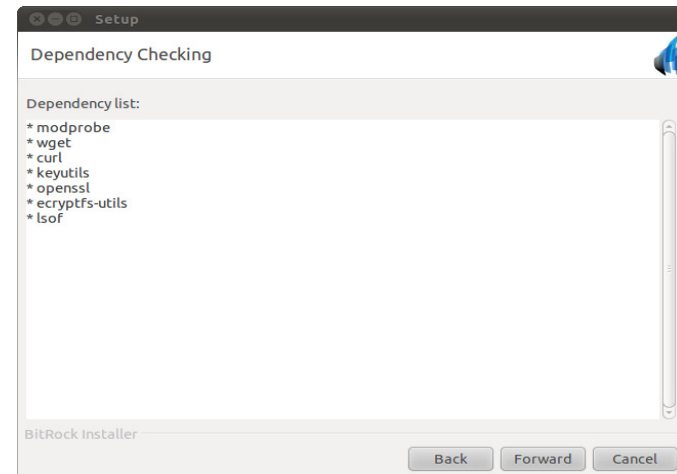
Click **Forward** to continue.



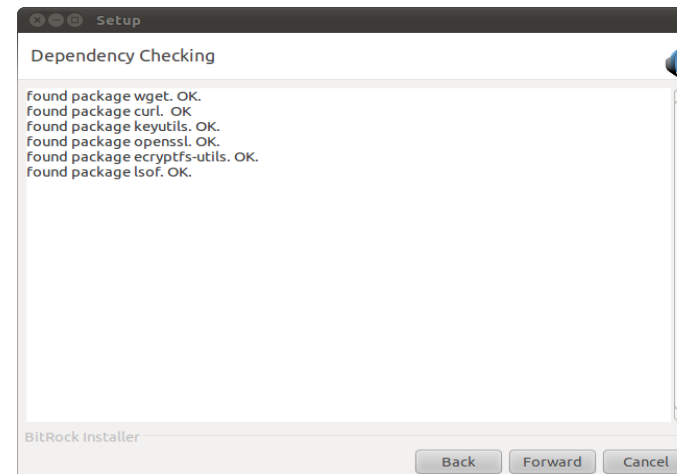
Read the License Agreement, and select the **I accept the agreement** option. Click **Forward** to continue.



Accept the default ezNcrypt installation directory (**/opt/ezNcrypt**) or specify another directory. Click **Forward** to continue.



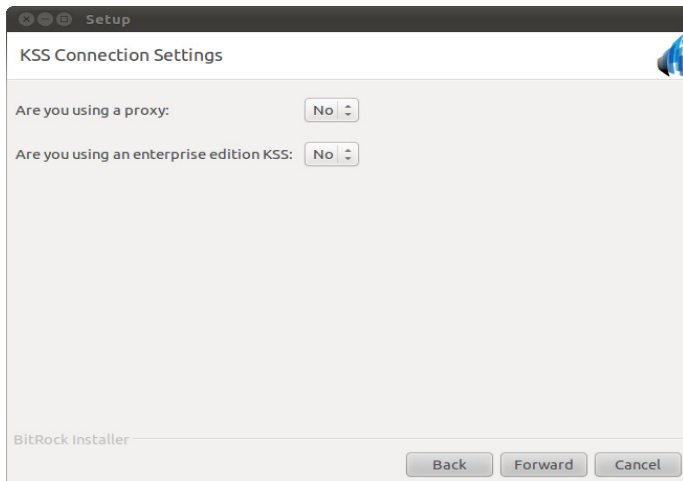
The installer displays a list of the necessary dependencies, and verifies whether or not the necessary dependencies are installed. At this point, you can stop the installer (**Ctrl-C**) and use your package manager to install the necessary dependencies, or you can click **Forward** to let the installer script call the package manager for you.



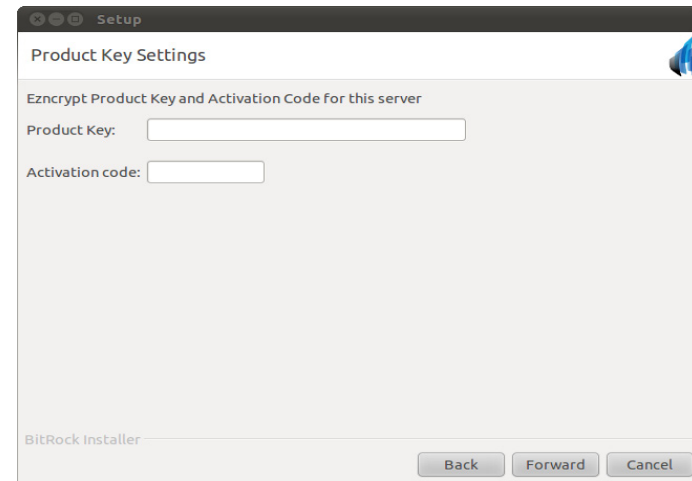
Click **Forward** to continue.



Select the desired cipher, and click **Forward** to continue.



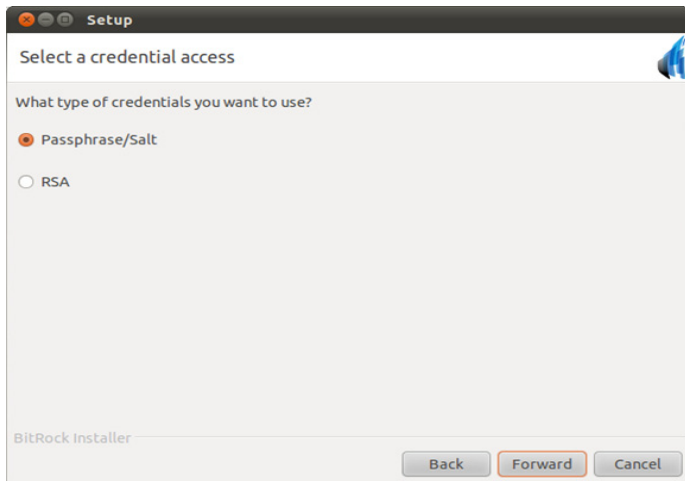
Provide the appropriate information, and click **Forward** to continue.



The installer prompts you for your ezNcrypt product key and activation code. These should have been provided to you by a Gazzang representative.

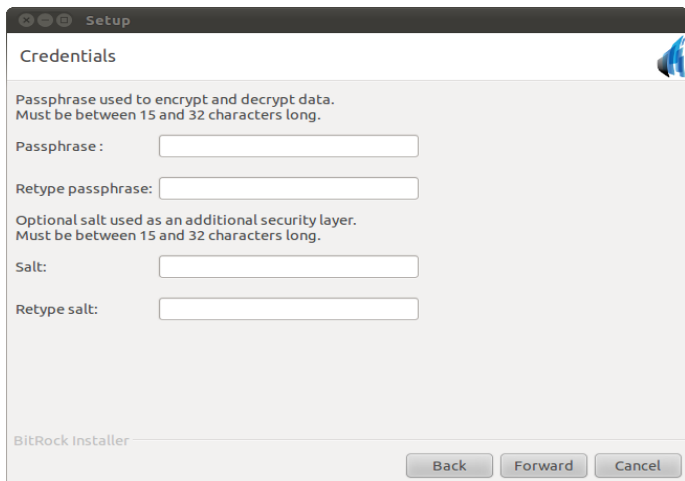
Note: It is important to carefully type the product key settings. For command line installation in most configurations, you can paste by middle-clicking in the terminal.

The installer will prompt you to choose between a passphrase and an RSA key to encrypt your data. Using a passphrase is the recommended option in most cases.

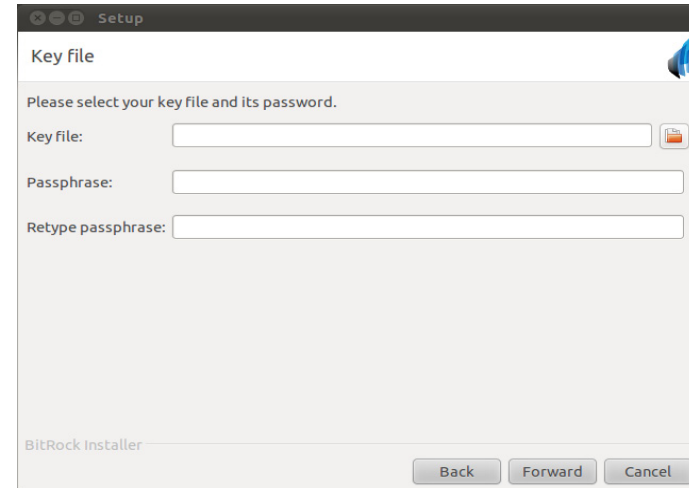


Note: If you have an existing Encryption Key Security policy and/or are using a hardware-based Key Storage module like TPM or HSM, you can use an RSA key and a random passphrase/salt will be generated for you. It is important that you store the random passphrase/salt in a secure place.

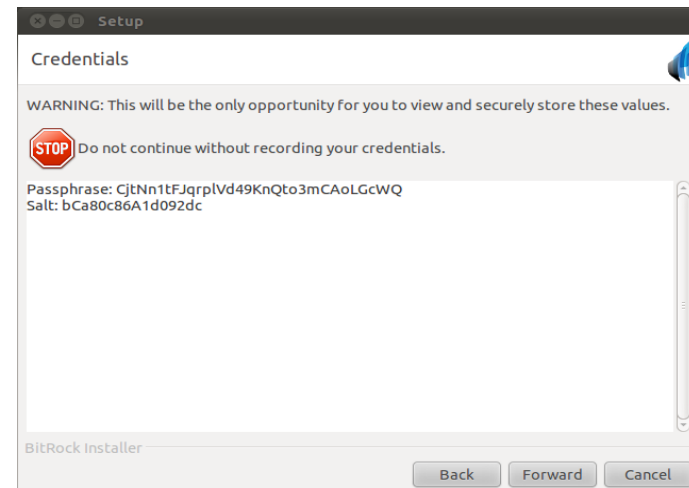
If you choose a passphrase, type a passphrase and/or salt. You can skip the salt configuration by leaving these fields blank, but if you choose to add a salt, this value must be between 15 and 32 characters long as does the passphrase.



If you choose RSA, type an RSA key file that you want to use, and type the RSA passphrase for this private key.



After providing an RSA, a random passphrase/salt will be generated.



WARNING: Write the passphrase down in a secure location so that you can always decrypt your data.

After providing the credentials, click **Forward** to continue. The installer will begin copying the ezNcrypt files to your system.

Note: If your distribution uses **apparmor**, the installer will attempt to properly configure it by editing the MySQL apparmor configuration file. This process is safe and only edits known apparmor configuration-files. If the files have been previously modified, or if the configuration file does not exist, the installer will abort. If this is the case, the installer script will display the changes that must be made to the configuration file. If you need further assistance, please contact support@gazzang.com.

Automated Installer

The automated installer lets you install from a command line, or by editing and executing the **install.options** installation file.

From a command line:

```
sudo ./ezncrypt-2.1-linux-installer.run --product_key YOUR_PRODUCT_KEY
--activation_code YOUR_ACTIVATION_CODE --passphrase YOUR_PASSPHRASE
--salt YOUR_SALT --mode unattended
```

Note: If you remove `--mode unattended`, the interactive installer will start, and the provided fields will be completed for you.

Or, edit the **install.options** configuration file, and issue the following command:

```
sudo ./ezncrypt-2.1-linux-installer.run --optionfile install.options
```

Sample **install.options** file:

```
product_key=YOUR_PRODUCT_KEY
activation_code=YOUR_ACTIVATION_CODE
passphrase=YOUR_PASSPHRASE
salt=YOUR_SALT
```

Note: In both cases, salt is optional.

3. Installation begins

The installer copies all necessary files and directories to your system. The default directory is `/usr/share/ezNcrypt`. Symbolic links that point to `/usr/share/ezncrypt/sbin` will be created on `/usr/sbin`.

ezNcrypt Files

Although all files will be in an exclusive ezNcrypt directory, the kernel module must be copied to your current kernel modules directory so that it can be properly loaded and unloaded.

The installer copies the `ezncryptfs.ko` file to the `/lib/modules/`uname -r`/kernel/fs/ezncryptfs` directory.

A start-up ezNcrypt script is used to start the service when you restart your computer. The installer copies this script to your current init scripts directory located in `/etc/init.d/`.

Data Directories

The default directory where your encrypted data is stored is `/var/lib/ezncrypt`. This directory contains the following directories:

```
(PRIVATE) /var/lib/ezncrypt/.encrypted_private
```

Private directory to store all encrypted data.

```
(PRIVATE) /var/lib/ezncrypt/.noencrypted_private
```

Private directory to store all un-encrypted data.

```
(PUBLIC) /var/lib/ezncrypt/ezncrypted
```

Public directory for encrypted data access. This directory must be used if you need to delete, copy, or move files.

WARNING: Do not use the private directories to edit or move files.

Configuration Files

The installer creates the */etc/ezncrypt* directory where all configuration settings are saved.

Do not delete any file from the */etc/ezncrypt* directory. These files provide necessary information to the ezNcrypt application.

ACCESS CONTROL MANAGEMENT

Gazzang ezNcrypt manages file system permissions through an access control list (ACL). This ACL is a security access control (created by Gazzang) that permits a Linux process to access a file or directory handled by ezNcrypt.

The ACL is set by rules. These rules tell the file system whether or not a Linux process has access right permissions to read/write a specific ezNcrypt path.

A rule is defined as follows:

TYPE @CATEGORY PATH PROCESS

Command	Description
TYPE	Tells the file system to allow or deny a process. It can have either of the following values: ALLOW or DENY .
@CATEGORY	Specifies the directory entry point to start. For example: @httpd , @mysql
PATH	Specifies the right permissions of a specific path. For example: * , www/*.htaccess
PROCESS	Specifies the process or command name for the rule.

All rules are stored in an Encrypted Policy File along with a set of process signatures used by ezNcrypt to authenticate a Linux process. This file is encrypted with the ezNcrypt key provided by the user at installation.

Adding Rules

Rules can be added in two ways: passing a rule as a parameter, or using a policy file. A policy file is the fastest way to add many rules because it only asks for the security key one time.

To add a rule by passing it as a parameter:

```
[root@host]# /usr/sbin/ezncrypt-access-control -a "ALLOW @mysql * /usr/sbin/mysqld"
```

Argument	Description
-a	Tells ezNcrypt to add a new rule to the current ezNcrypt policy.

To add two or more rules using a policy file:

```
[root@host]# /usr/sbin/ezncrypt-access-control -a -f my-policy-file
```

Argument	Description
-a	Tells ezNcrypt to add a new rule to the current ezNcrypt policy.
-f	Specifies the policy file.

Policy file example:

```
ALLOW @mysql * /usr/sbin/mysqld
ALLOW @log * /usr/sbin/mysqld
ALLOW @apache * /usr/lib/apache2/mpm-prefork/apache2
```

You can override the policy file with a set of rules stored in an input FILE:

```
[root@host]# /usr/sbin/ezncrypt-access-control -o -f my-policy-file
```

Argument	Description
-o	Tells ezNcrypt to override the current ezNcrypt policy file with the new policy FILE.
-f	Specifies the policy file.

Deleting Rules

Rules are deleted in two ways: passing a rule as a parameter, or by passing the line where the rule resides on the file. This line can be seen by issuing the list option (see Printing Rules).

To delete a rule by passing it as a parameter:

```
[root@host]# /usr/sbin/ezncrypt-access-control -d "ALLOW @mysql * /usr/sbin/mysqld"
```

Argument	Description
-d	Tells ezNcrypt to delete a rule from the current ezNcrypt policy.

To delete a rule by passing a line number:

```
[root@host]# /usr/sbin/ezncrypt-access-control -d -l 2
```

Argument	Description
-d	Tells ezNcrypt to delete a rule from the current ezNcrypt policy.
-l	Specifies the number line of the rule.

Printing Rules

You can view all rules added to the ezNcrypt policy file by issuing the following command:

```
[root@host]# /usr/sbin/ezncrypt-access-control -p
ALLOW @mysql * /usr/sbin/mysqld
ALLOW @log * /usr/sbin/mysqld
ALLOW @apache * /usr/lib/apache2/mpm-prefork/apache2
```

You can also send the rules directly to a file:

```
[root@host]# /usr/sbin/ezncrypt-access-control -p -f policy-backup
```

The above methods show you just the rules stored in the ezNcrypt policy file. The following option also shows extra information about the organization of the policy file:

```
[root@host]# /usr/sbin/ezncrypt-access-control -L
# - Type      Category Path                Process
1  ALLOW      @mysql  *                  /usr/sbin/mysqld
2  ALLOW      @log    *                  /usr/sbin/mysqld
3  ALLOW      @apache *                  /usr/lib/apache2/mpm-
prefork/apache2
```

Loading Rules

The following command reloads the access control list (ACL) to the ezNcrypt file system. This loads the updated ACL rules to the file system about which processes will have access to ezNcrypt:

```
[root@host]# /usr/sbin/ezncrypt-access-control -r
```

Updating a Process Signature

All rules have a reference to a process signature that is used to authenticate the process into the file system. If the file system detects a signature that is different than the one stored in the ACL, the Linux process will be denied and treated as an untrusted process.

There are times when this process signature must be updated, for example, due to a software upgrade. When the signature must be updated, the ezNcrypt administrator reauthenticates the process on the ACL by executing the *ezNcrypt-acl -u* command:

```
[root@host]# /usr/sbin/ezncrypt-access-control -u
ALERT: Hash for '/usr/sbin/mysqld' has changed.
```

If you need to know about signature changes or process errors, you can see them by listing the rules:

```
[root@host]# /usr/sbin/ezncrypt-access-control -L
# - Type      Category Path                Process
1  !! ALLOW      @mysql  *                  /usr/sbin/mysqld
2  !! ALLOW      @log    *                  /usr/sbin/mysqld
3  EE ALLOW      @apache *                  /usr/lib/apache2/mpm-prefork/apache2
```

The '!!' character warns you about a process signature issue. These processes must be updated.

The 'EE' character warns you about a process read error such as the process does not exist or there is a permission issue.

USING THE SERVICE

ezNcrypt Service

To use the encryption utilities, you must start the ezNcrypt service. The `ezncrypt-service` command gives you the option to start, stop, restart, and review the ezNcrypt status using the following syntax:

```
[root@host] # /usr/sbin/ezncrypt-service [OPTIONS]
[start|stop|restart|status]
```

Option	Description
start	Starts the ezNcrypt service.
stop	Stops the ezNcrypt service.
restart	Restarts the ezNcrypt service.
status	Reports whether or not the ezNcrypt service is running.

Argument	Description
-f	Force ezNcrypt stop (skip warnings).
-p	Prompts to type the ezNcrypt key manually.
-h	Shows ezNcrypt_service help.

To start the service, issue the following command. This command retrieves the encryption passphrase from the KSS server:

```
[root@host] # /usr/sbin/ezncrypt-service start
ezncrypt | Checking system dependencies
ezncrypt | checking encryption directories
keymgr | Retrieving key from KSS
| > Encryption password retrieved from KSS
ezncrypt | starting service
| > using "aes_256" cipher algorithm
| done!
access | Loading access control list
| done!
ezncrypt | Thank you for using ezncrypt.
log | File: /var/log/ezncrypt/ezncrypt-service.log
```

To stop the service, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-service stop
ezncrypt | Checking system dependencies
ezncrypt | WARNING: ezncrypt service will be stopped.
| Continue? (y/N) y
| Are you sure? (y/N) y
ezncrypt | stopping service
| done!
log | File: /var/log/ezncrypt/ezncrypt-service.log
```

To restart the service, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-service restart
ezncrypt | Checking system dependencies
ezncrypt | initiating ezncrypt server STOP process
| ezncrypt system is not running.
ezncrypt | initiating ezncrypt server START process
ezncrypt | checking encryption directories
keymgr | Retrieving key from KSS
| > Encryption password retrieved from KSS
ezncrypt | starting service
| > using "aes_256" cipher algorithm
| done!
access | Loading access control list
| done!
ezncrypt | Thank you for using ezncrypt.
log | File: /var/log/ezncrypt/ezncrypt-service.log
```

To see the service status, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-service status
ezncrypt | Checking system dependencies
** ezncrypt system is UP and running **
log | File: /var/log/ezncrypt/ezncrypt-service.log
```

Encrypting MySQL Tables

To encrypt tables on a selected database, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-mysql --encrypt DATABASE [TABLE]
```

This command encrypts the selected tables by moving them to the following encrypted directory: `/var/lib/ezncrypt/ezncrypted`. This command also creates the appropriate links to the files that MySQL expects to be in the mysql data directory.

Note: You can use MySQL wildcards to encrypt the desired tables. If you do not specify the % wildcard or the table name to be encrypted, all of the tables in the DATABASE will be encrypted, as in the following example.

```
[root@host]# /usr/sbin/ezncrypt-mysql --encrypt employees
ezncrypt | Checking system dependencies
          | Verifying ezncrypt license
mysql    | Please provide a MySQL username & password
          | Enter username: root
          | Enter password:
mysql    | Looking for 'employees' on MySQL
          | done!
keymgr   | Retrieving key from KSS
          | > Encryption password retrieved from KSS
          | generating keys
          | done!
ezncrypt | checking encryption status
          | done!
          | WARNING: MySQL will be stopped while encryptng data.
          | Continue? (Y/n)
mysql    | stopping mysql service
          | done!
ezncrypt | preparing database directory
          | moving data to encryption directory
          | This can take a while. Please be patient
          | > /var/lib/ezncrypt/ezncrypted/mysql/employees
          | done!
ezncrypt | encrypting selected tables
          | > encrypting 'MyISAM/departments'
          | > encrypting 'MyISAM/dept_emp'
          | > encrypting 'MyISAM/dept_manager'
          | > encrypting 'MyISAM/employees'
          | > encrypting 'MyISAM/salaries'
          | > encrypting 'MyISAM/titles'
          | done!
mysql    | starting mysql service
          | waiting
          | done!
ezncrypt | congratulations. you have encrypted your MySQL tables
log      | File: /var/log/ezncrypt/ezncrypt-mysql.log
```

You can also use the `ezncrypt-mysql -e database table` command to encrypt a table. Issue the `ezncrypt-mysql -h` command for help.

Note: The `DEFAULT_ENCRYPTION` variable located at `/etc/ezncrypt/ezncrypt.conf` lets you set the encryption of all new tables created on your database. This value can be **yes** (default) or **no**.

Decrypting MySQL Tables

To decrypt tables on a selected database, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-mysql --decrypt DATABASE [TABLE]
```

Note: You can use MySQL wildcards to decrypt the desired tables.

For the decrypt command, the passphrase is required for security purposes. For example, the following command decrypts every table in the DATABASE:

```
[root@host]# /usr/sbin/ezncrypt-mysql --decrypt employees
ezncrypt | Checking system dependencies
mysql    | Please provide a MySQL username & password
          | Enter username: root
          | Enter password:
mysql    | Looking for 'employees' on MySQL
          | done!
key      | Type your encryption/decryption key
          | type passphrase:
          | type salt:
          | generating keys
          | done!
ezncrypt | checking encryption status
          | done!
          | WARNING: MySQL will be stopped while decryptng data.
          | Continue? (Y/n)
mysql    | stopping mysql service
          | done!
ezncrypt | checking disk space
          | done!
ezncrypt | decrypting selected tables
          | > decrypting 'MyISAM/departments'
          | > decrypting 'MyISAM/dept_emp'
          | > decrypting 'MyISAM/dept_manager'
          | > decrypting 'MyISAM/employees'
          | > decrypting 'MyISAM/salaries'
          | > decrypting 'MyISAM/titles'
          | done!
ezncrypt | returning database to MySQL directory
          | moving data to mysql directory
          | This can take a while. Please be patient
          | > /var/lib/mysql/employees
          | done!
mysql    | starting mysql service
          | waiting
          | done!
ezncrypt | congratulations. you have decrypted your MySQL tables
log      | File: /var/log/ezncrypt/ezncrypt-mysql.log
```

This command decrypts the selected tables, and if there are no more encrypted tables, the command also copies the unencrypted database to its original directory, */var/lib/mysql*, replacing the symbolic link.

Show Encrypted Databases

To show information about encrypted databases and tables on the file system, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-mysql --show
```

Example:

```
[root@host] # /usr/sbin/ezncrypt-mysql --show
ezncrypt | Checking system dependencies
ezncrypt | MySQL Databases
          | employees (6 encrypted, 0 unencrypted)
```

The following example assumes that you have **dept_manager**, **salaries**, and **titles** tables encrypted:

```
[root@host] # /usr/sbin/ezncrypt-mysql -s employees
ezncrypt | Checking system dependencies
ezncrypt | Tables encrypted
          | - dept_manager
          | - salaries
          | - titles
ezncrypt | Tables unencrypted
          | - departments
          | - employees
          | - dept_emp
```

Encrypting a File

To encrypt a file or directory (like an Apache configuration file), issue the following command:

```
[root@host] # /usr/sbin/ezncrypt --encrypt @CATEGORY FILE|DIRECTORY
```

Argument	Description
@CATEGORY	Tells ezNcrypt where to save the encrypted file or directory. This is the main entry path of ezNcrypt.
FILE DIRECTORY	Specifies a file or directory to encrypt.

Example:

```
[root@host]# /usr/sbin/ezncrypt --encrypt @apache /etc/apache2/conf.d/
ezncrypt | Checking system dependencies
          | Verifying ezncrypt license
          | getting information about location
          |   > path: /var/lib/ezncrypt/ezncrypted/apache
ezncrypt | Checking encryption status
          | done!
          keymgr | Retrieving key from KSS
          |   > Encryption password retrieved from KSS
          | generating keys
          | done!
          backup | backing up data
          |   > Backup disabled. Use -b to enable backup.
ezncrypt | encrypting files
          |   > checking disk space
          |   > encrypting /etc/apache2/conf.d
          | done!
ezncrypt | congratulations. you have encrypted your Files!!
          log | File: /var/log/ezncrypt/ezncrypt.log
```

Decrypting a File

To decrypt a directory or file, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt --decrypt @CATEGORY FILE|DIRECTORY
```

Argument	Description
@CATEGORY	Tells ezNcrypt where the encrypted file or directory is stored. This is the main entry path of ezNcrypt.
FILE DIRECTORY	Specifies a file or directory to decrypt.

Example:

```
[root@host]# /usr/sbin/ezncrypt --decrypt @apache /etc/apache2/conf.d/
ezncrypt | Checking system dependencies
          | getting information about location
          | > path: /var/lib/ezncrypt/ezncrypted/apache
ezncrypt | Checking encryption status
          | done!
          | key | Type your encryption/decryption key
          | type passphrase:
          | type salt:
          | generating keys
          | done!
ezncrypt | decrypting files
          | > checking disk space
          | > /etc/apache2/conf.d
          | done!
ezncrypt | congratulations. you have decrypted your Files!!
          | log | File: /var/log/ezncrypt/ezncrypt.log
```

Encrypt/Decrypt Standalone Files

To display ezNcrypt command help, use the `-h` option, as follows:

```
[root@host] # /usr/sbin/ezncrypt -h
```

Command Usage:

```
ezncrypt options|command [command_opts] [command_args]
```

Option	Description
-e, --encrypt	Encrypt a file or directory.
-d, --decrypt	Decrypt a file or directory.

Argument	Description
-f, --file	Tells the ezNcrypt command to encrypt standalone files.
-i, --in=FILE	Source file to encrypt.

Argument	Description
-o, --out=FILE	Destination file (encrypted).

To encrypt standalone files that are compatible with ezNcrypt encryption, issue the following command:

Note: The passphrase used in this command can be different from the system passphrase.

```
[root@host] # /usr/sbin/ezncrypt --encrypt --file --in=SOURCE --out=DEST
```

Example:

```
[root@host] # /usr/sbin/ezncrypt --encrypt --file --in=my_test_file --
out=my_test file encrypted
ezncrypt | Checking system dependencies
          | key | Type your encryption/decryption key
          | type passphrase:
          | type salt:
ezncrypt | encrypting file
          | > file encrypted on 'my_test_file_encrypted'
          | log | File: /var/log/ezncrypt/ezncrypt.log
```

To decrypt standalone files that are compatible with ezNcrypt decryption, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt --decrypt --file --in=SOURCE --out=DEST
```

Example:

```
[root@host] # /usr/sbin/ezncrypt --decrypt --file --
in=my_test_file encrypted --out=my_test_file decrypted
ezncrypt | Checking system dependencies
          | key | Type your encryption/decryption key
          | type passphrase:
          | type salt:
ezncrypt | decrypting file
          | > file decrypted on 'my_test_file_decrypted'
          | log | File: /var/log/ezncrypt/ezncrypt.log
```

EXECUTING SCRIPTS

Gazzang ezNcrypt can authenticate any Linux command, but there are occasions when you may need to grant a command line script access to the ezNcrypt directory for a single execution without making the directory accessible all of the time.

For example, if you want to grant access to a shell script that copies files from ezNcrypt, you must allow */bin/bash* so that you can run the script. Unfortunately, this allows any shell script, or Linux command, to have access to ezNcrypt.

For this reason, the following command lets the shell execute the script and give access to ezNcrypt:

```
[root@host]# /usr/sbin/ezncrypt-run SCRIPT
```

The following is an example of the *list_databases.sh* shell script:

```
#!/bin/bash
ls -l /var/lib/ezncrypt/ezncrypted/mysql
```

If you issue this command without the ezNcrypt helper, then ezNcrypt will deny access to the script. (The ezNcrypt service must be running.)

```
[root@host]# ./list-databases.sh
ls: cannot open directory /var/lib/ezncrypt/ezncrypted/mysql: Permission
denied
```

Instead, use the ezNcrypt helper to issue the command. ezNcrypt will ask you for the key to permit access to the script.

```
[root@host] # /usr/sbin/ezncrypt-run ./list-databases.sh
passphrase:
salt:
total 4
drwx----- 2 mysql mysql 4096 May 19 16:36 employees
```

The ezNcrypt helper is useful for running others types of scripts such as Perl, PHP, Python, etc.

CHANGING THE ENCRYPTION KEY

To change the key (PASSPHRASE or RSA), all databases, views, and files previously encrypted should be decrypted using the proper commands.

Decrypt All Tables

```
[root@host] # /usr/sbin/ezncrypt-mysql --decrypt DBNAME %
```

In this command, *DBNAME* is the name of the Database and ``%`` works as a MySQL wildcard. This wildcard makes the command act on all tables and views in the *DBNAME* database.

Decrypt Files or Directories

```
[root@host] # /usr/sbin/ezncrypt --decrypt @CATEGORY FILE/DIRECTORY
```

In this command, *@CATEGORY* is the rule used to encrypt the file or directory, *FILE* is the file name to be decrypted, and *DIRECTORY* is the directory to be decrypted.

Change Encryption Key

```
[root@host] # /usr/sbin/ezncrypt-change-key
```

This command first attempts to stop the ezNcrypt service (you must confirm this action) and then prompts you for your old credentials. The command then prompts you for your new salt and passphrase and safely stores them. After this is complete, the command restarts the ezNcrypt service.

To display command help, use the `-h` option, as follows:

```
[root@host] # /usr/sbin/ezncrypt-change-key -h
```

Encrypt All Tables

You can now encrypt your tables with your new credentials. To do so, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-mysql --encrypt DATABASE TABLES
```

Note: You can use MySQL wildcards to encrypt the desired tables.

UTILITIES

The following commands are provided to validate the passphrase locally and against the KSS, and to display the processes that access ezNcrypt.

Check Key

To validate a given key against the KSS and current encrypted data, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-check-key
```

Example:

```
[root@host] # /usr/sbin/ezncrypt-check-key
ezncrypt-check-key performs checking of your encryption keys (passphrase
and salt)
passphrase:
salt:
Checking your encryption key - validating with your ezNcrypt local service
...
Checking your encryption key - validating with the KSS Server ...
SUCCESS: Your encryption key is validated properly with your ezNcrypt local
service
SUCCESS: Your encryption key is validated properly with KSS
```

Processes Accessing ezNcrypt

The *ezncrypt-top* command provides an ongoing look at process activity in real time. This command displays a list of the processes that read/write data on ezNcrypt. To display the top processes that access ezNcrypt, issue the following command:

```
[root@host] # /usr/sbin/ezncrypt-top
```

Example:

```
ezncrypt - 0 days 00:16:48 hours running
Disk: 76K encrypted, 68K unencrypted

  PID USER          FD   READ   WRITE TIME           PROCESS
-----
 3176 mysql          6  1524k    0k 00:00:30  mysqld
```

UNINSTALLING

WARNING: Ensure that all of your databases and files are decrypted and placed into the MySQL directory before uninstalling the software.

To uninstall ezNcrypt, you must stop the ezNcrypt service and run the uninstall script, or delete all of the ezNcrypt files manually, as follows:

```
[root@host] # /usr/sbin/ezncrypt-service stop  
su -  
#(insert root password)  
/opt/ezncrypt/uninstall
```

Or (depending on your distro):

```
sudo /opt/ezncrypt/uninstall
```

Or manually delete all ezNcrypt files:

```
rm -rf /usr/share/ezncrypt  
rm -f /usr/sbin/ezncrypt-*  
rm -f /usr/bin/ezncrypt-*  
rm -rf /etc/ezncrypt  
rm -rf /lib/modules/`uname -r`/kernel/fs/ezncryptfs  
rm -rf /var/lib/ezncrypt  
rm -f /etc/init.d/ezncrypt
```

EZNCRYPT SETTINGS

You can change any of the following ezNcrypt settings after installation. These settings are stored in the **/etc/ezncrypt/ezncrypt.conf** file.

[Encryption Settings]

CIPHER

Lets you change the encryption algorithm used for protecting your data. This setting can only be successfully modified if there is no encrypted data.

The supported algorithms vary from system to system.

Valid values: "aes_128", "aes_192", "aes_256", "twofish_128", "twofish_256", "blowfish_128", "blowfish_256"

DEFAULT_ENCRYPTION

Lets you set the encryption of all new tables created on your database server.

Valid values: "yes", "no"

[Mysql Settings]

MYSQLD

Lets you specify where the MySQL daemon start interface is located.

Valid value: *<complete path to mysqld file>*

MYSQLDAPMIN

Lets you specify the path of your *mysqladmin* command.

Valid value: *<complete path to mysqladmin file>*

MYSQL_BIN

Lets you specify the path of your MySQL binary file.

Valid values: *<complete path to mysql binary file>*

PASSWORD_TIMEOUT_DEFAULT: 60

Lets you specify how long to wait for the password prompt (in seconds).

Valid values: *<0 - 999>*

[Storage Settings]

This collection of settings allows you to specify personalized locations for ezNcrypt to use.

INSTALL_DIR DEFAULT: /usr/share/ezncrypt

Specifies the path where ezNcrypt is installed.

Valid value: *<path>*

DIR_VIRTUAL

Mount point for the ezNcrypt virtual directory.

Valid value: *<path>*

DIR_PRIVATE_ENCRYPTED

Mount point for ezNcrypt virtual encrypted directory.

Valid value: *<path>*

DIR_PRIVATE_NONENCRYPTED

Mount point for ezNcrypt virtual non-encrypted directory.

Valid value: *<path>*

DIR_BACKUP

Mount point for ezNcrypt backup directory.

Valid value: *<path>*

[Proxy Settings]

`proxy_host`

Lets you set a proxy for all of your ezNcrypt connections.

Valid value: "host:port"

`proxy_cred`

Lets you set a username and password for your proxy-enabled connections.

Valid value: "username:password"

[SSL mutual authentication settings]

To enable SSL mutual authentication, you must contact Gazzang to arrange the inclusion of your Certificate Authority credentials into the service.

SSL_CERTIFICATE_FILE DEFAULT:
`/etc/ssl/certs/server.cer`

Specifies the location of your server's certificate.

Valid value: `<path>`

SSL_CERTIFICATE_KEY_FILE DEFAULT:
`/etc/ssl/private/server.key`

Specifies the location of your server's key file.

Valid value: `<path>`

SSL_CERTIFICATE_KEY_PASS=passphrase

Specifies the passphrase of your server's ssl key.

Valid values: "ssl key passphrase"

SSL_CA_CERTIFICATE_FILE DEFAULT:
`/etc/ssl/certs/ca.crt`

Specifies the location of your server's certificate authority file.

Valid value: `<path>`

TROUBLESHOOTING

The following is a list of common errors and solutions:

ERROR | MYSQLD is not defined on ezncrypt configuration file.

This error is displayed when ezNcrypt is installed without previously installing MySQL. To correct this error, re-install ezNcrypt or run the ezNcrypt configuration utility with the following command:

```
[root@host] # ./ezncryptconfig-2.1-linux-installer.run
```

WARNING! table_name: 'NULL' engine is not supported

This error is due to Selinux configuration. To correct this error, issue the following command: *setenforce 0*

APPENDIX A: MYSQL

Storage Engine Differences

While ezNcrypt is a table-based abstraction layer that allows you to conveniently encrypt tables, it internally works in terms of files. This is not a problem when working with the MyISAM storage engine due to the way in which files are stored; however, when using InnoDB, the MySQL configuration must be changed so that files can be correctly handled by the ezNcrypt utilities.

InnoDB Support

To encrypt tables using the InnoDB engine, the tables must be separated into files. MySQL has a built-in option that allows each table to be stored in a separate file.

To activate this option, you must edit your MySQL configuration file, **/etc/my.cnf**, as follows:

1. Add the `innodb_file_per_table` option under the **[mysqld]** section.
2. Restart the MySQL server.

All newly created tables will be stored in separate files. To modify existing tables, issue the following command:

```
ALTER TABLE <table> ENGINE=InnoDB;
```

If tables were created before you updated the configuration file and you want those table to be encrypted, you must issue this command for each table. After successfully performing these steps, you can start encrypting InnoDB tables with ezNcrypt.

Accessing the ezNcrypt Directory

The ezNcrypt MySQL directory cannot be accessed by external users. This data is protected and you must have the proper key to access un-encrypted data.

If you attempt to read the directory contents, you will receive the following error message:

```
ls /var/lib/ezncrypt/ezncrypted/mysql
ls: cannot access /var/lib/ezncrypt/ezncrypted/mysql: Required key not available
```

To be allowed access to the data and make backups using utilities such as `cp`, `scp`, `rsync`, etc, you must execute **ezncrypt-load-key** with the proper privileges, and select option **1** to add the key to the current user, as follows:

```
[root@host]# /usr/sbin/ezncrypt-load-key
ezncrypt | Menu list
> | 1. Add encryption key to current user
> | 2. Exit
| Option: 1
key | Type your encryption/decryption key
| type passphrase:
| type salt:
ezncrypt | Adding key to user keyring
```

Provide the password that you use to encrypt and decrypt MySQL data.

After this, you will be allowed to access the ezNcrypt directory and see the un-encrypted files.

```
ls /var/lib/ezncrypt/ezncrypted/mysql
my_database
```

You can now copy or restore databases on this directory.

Example:

```
cp -R /var/lib/ezncrypt/ezncrypted/mysql/my_database $HOME/mysql_backups
```

To protect the directory after you are finished, you must clear the key ring:

```
keyctl clear @u
```

MySQL Log Encryption

Binary logs and other MySQL logs can be encrypted by running the ezNcrypt `-encrypt` command, as follows:

1. Stop MySQL.

```
/etc/init.d/mysqld stop
```

2. Encrypt any log, for example, `/var/log/mysql.err`.

```
[root@host] # /usr/sbin/ezncrypt --encrypt @log /var/log/mysql.err
ezncrypt | Checking system dependencies
          | Verifying ezncrypt license
          | getting information about location
          |   > path: /var/lib/ezncrypt/ezncrypted/log
ezncrypt | Checking encryption status
          | done!
keymgr   | Retrieving key from KSS
          |   > Encryption password retrieved from KSS
          | generating keys
          | done!
backup   | backing up data
          |   > Backup disabled. Use -b to enable backup.
ezncrypt | encrypting files
          |   > checking disk space
          |   > encrypting /var/log/mysql.err
          | done!
ezncrypt | congratulations. you have encrypted your Files!!
log      | File: /var/log/ezncrypt/ezncrypt.log
```

3. Restart MySQL.

```
/etc/init.d/mysqld start
```

Your logs are now encrypted into the `@log` category.

APPENDIX B: APACHE

Apache Docs Encryption

Apache web pages and apache logs can be encrypted by running the ezNcrypt `-encrypt` command, as follows:

1. Stop Apache.

```
/etc/init.d/apache2 stop
```

2. Encrypt `/var/www`

```
[root@host] # /usr/sbin/ezncrypt -e @www /var/www
ezncrypt | Checking system dependencies
          | Verifying ezncrypt license
          | getting information about location
          |   > path: /var/lib/ezncrypt/ezncrypted/www
ezncrypt | Checking encryption status
          | done!
          |
          | keymgr | Retrieving key from KSS
          |   > Encryption password retrieved from KSS
          | generating keys
          | done!
          |
          | backup | backing up data
          | This can take a while. Please be patient
          |   > backing up /var/www
          |   > File: /backup/mysql/2011-08-19/www.tar.gz
          | done!
ezncrypt | encrypting files
          |   > checking disk space
          |   > encrypting /var/www
          | done!
ezncrypt | congratulations. you have encrypted your Files!!
          | log | File: /var/log/ezncrypt/ezncrypt.log
```

3. Configure ACL Rules.

```
[root@host] # /usr/sbin/ezncrypt-access-control -a "ALLOW @www *
/usr/lib/apache2/mpm-prefork/apache2"
passphrase:
salt:
Rule added
```

4. Start Apache.

```
/etc/init.d/apache2 start
```

Your web pages are now encrypted into the `@www` category.

LICENSE AGREEMENT

GAZZANG LICENSE AGREEMENT ezNcrypt™ Software (All Editions) (v2011.1)

IMPORTANT – READ THIS LICENSE AGREEMENT (“AGREEMENT”) BEFORE CLICKING ON THE “ACCEPT” BUTTON, ENTERING “YES” IN RESPONSE TO THE ELECTRONIC LICENSE ACCEPTANCE INQUIRY, INSTALLING, OR ELECTRONICALLY DOWNLOADING. ANY OF THE ABOVE ACTIONS INDICATE ACCEPTANCE OF, AND LEGALLY BINDS YOUR COMPANY (“LICENSEE”), AND GAZZANG, INC. (“GAZZANG”) TO THE TERMS AND CONDITIONS SET FORTH BELOW. Licensee’s written approval is not a prerequisite to the validity or enforceability of this Agreement and no solicitation of any such written approval by or on behalf of Gazzang will be construed as an inference to the contrary.

This Agreement applies to the object code copy of the Gazzang ezNcrypt software which accompanied this Agreement, any license key or keys, and any subsequent software update Licensee receives of the foregoing, together with any included documentation (collectively, the “Software”).

This Agreement is divided into four parts: Part I: Terms and Conditions Applicable to Trial Licenses; Part II: Terms and Conditions Applicable to ezNcrypt Subscription Licenses; Part III: Terms and Conditions Applicable to the Provision of the Gazzang-hosted Key Storage System; Part IV: General Terms and Conditions Applicable to all Licensees. **ALL RIGHTS GRANTED TO LICENSEE UNDER PARTS II AND III WILL BE SUBJECT TO PAYMENT OF APPLICABLE LICENSE FEES.**

PART I: TERMS AND CONDITIONS APPLICABLE TO TRIAL LICENSE

1. **Trial License.** Licensee may order Trial Licenses for the Software on-line by downloading the Software and agreeing to these terms and conditions or pursuant to an agreement separately executed by the parties (“Trial License Order”). Subject to the terms and conditions of this Agreement, Gazzang hereby grants to Licensee a non-sublicensable, non-transferable, non-exclusive, royalty-free license to use the Software in object code form in accordance with the accompanying documentation solely for Licensee’s internal evaluation, development and testing purposes (“Trial License”) for a period of 30 days commencing on download of the Software by Licensee (“Trial Period”). Unless the Trial License is terminated by Licensee on or prior to the end of the Trial Period in written notice to Gazzang or on-line at the applicable Gazzang webpage, a Subscription License (as defined in Part II below) shall automatically commence at the then current fees for Subscription Licenses (viewable on Gazzang’s website) unless otherwise set forth in the Trial License Order or other Order pursuant to Part II below. In the event Licensee fails to pay for the Subscription License in accordance with Part II below upon receiving notice of the transition to the Subscription License and/or an invoice, as applicable, Gazzang may at any time thereafter disable the Software. By disabling the Software, this Agreement will be deemed terminated. Gazzang will have no obligation to provide technical support for the Software during the Trial Period.

2. **Warranty Disclaimer.** DURING THE TRIAL PERIOD, THE SOFTWARE IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT

LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.

3. **Limitation of Remedies and Damages. DURING THE TRIAL PERIOD,** NEITHER GAZZANG NOR ITS THIRD PARTY SUPPLIERS WILL BE RESPONSIBLE OR LIABLE WITH RESPECT TO ANY SUBJECT MATTER RELATING TO THE SOFTWARE UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY OR OTHER LEGAL OR EQUITABLE THEORY (A) FOR LOSS OR INACCURACY OF DATA OR COST OF PROCUREMENT OF SUBSTITUTE GOODS, SERVICES OR TECHNOLOGY, OR (B) FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL OR PUNITIVE DAMAGES INCLUDING, BUT NOT LIMITED TO LOSS OF REVENUES AND LOSS OF PROFITS. NOTHING IN THIS PART I WILL LIMIT GAZZANG’S LIABILITY FOR DEATH OR PERSONAL INJURY CAUSED BY GAZZANG’S NEGLIGENCE OR GAZZANG’S LIABILITY FOR FRAUD.

4. **No Indemnity.** During the Trial Period, Gazzang will not be liable to Licensee under any claim, suit or action, under any theory of or related to indemnity arising from the Software.

PART II: TERMS AND CONDITIONS APPLICABLE TO ezNcrypt STANDARD EDITION AND ezNcrypt ENTERPRISE EDITION

1. **Applicability.** The terms and conditions of this Part II and Part III will supersede the terms and conditions of Part I for Licensee’s orders of Subscription Licenses (as defined below), and the terms and conditions of Part IV will continue to apply.

2. **Orders.** In addition to Subscription Licenses automatically commencing at the end of the Trial Period, as set forth in Part I above, Licensee may also place orders for Subscription Licenses in accordance with Gazzang’s on-line order system or pursuant to a mutually executed order form between the parties (each, an “Order”).

3. **License Grant.** Subject to the terms and conditions of this Agreement, Gazzang grants Licensee a non-sublicensable, non-transferable, non-exclusive license (“Subscription License”) for a limited term as set forth on the Order (“Subscription Term”) to use the Software provided hereunder in object code form for production purposes in accordance with the documentation provided with the Software for the number of Servers (each, a “Server License”) set forth in an Order. Licensee has no right to receive any source code or design documentation relating to the Software.

4. **Disabling the Software.** Licensee’s right to use the Software is governed by a limiting device in the Software, which is designed to prevent its unlicensed use. In the event Licensee (i) fails to place an Order for a Subscription License at the end of the Trial Period, (ii) fails to timely pay for a Subscription License, (iii) fails to renew a Subscription License at the end of the Subscription Term, or (iv) is otherwise in breach of this Agreement, Gazzang may disable the Software or individual Server Licenses without notice to Licensee.

5. **Payment.** All rights granted to Licensee and obligations of Gazzang under this Part II and Part III will be subject to payment of applicable License Fees due upon commencement of the Subscription License, based on (i) the number of Servers

encrypted by the Software as set forth in the Order or Trial Period Order, as applicable, and (ii) any subsequent Servers which Licensee may add from time to time ("License Fees"). Except as set forth in Section 5 of this Part II, all fees due hereunder are nonrefundable. All amounts payable under this Agreement are exclusive of all sales, use, value-added, withholding, and other taxes and duties. Licensee will pay all such taxes and duties, except for taxes payable on Gazzang's net income. Except for invoices disputed in good faith, all past due amounts will incur interest at a rate equal to the lower of 1.0% per month or the highest rate permitted by law, beginning as of 15 days after the applicable due date. If at any time Licensee is delinquent (including during any grace periods) in the payment of License Fees, Gazzang may, in its discretion, disable any Server Licenses related to such unpaid License Fees. If an executed order contains different payment terms, those terms will apply.

6. Limited Warranty. Gazzang warrants for a period of 30 days from Licensee's first installation of the Software ("Warranty Period") that the Software will materially conform to Gazzang's then-current user documentation for such Software. This warranty covers only problems reported to Gazzang during the Warranty Period. Any liability of Gazzang for a breach of the foregoing warranty will be limited exclusively to Software repair or replacement or, if repair or replacement is commercially impractical, refund of the License Fee paid for the Software. EXCEPT FOR THE FOREGOING, ALL SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. FURTHER, GAZZANG DOES NOT WARRANT RESULTS OF USE OR THAT THE SOFTWARE IS BUG FREE OR THAT ITS USE WILL BE UNINTERRUPTED.

7. Limitation of Remedies and Damages. NOTWITHSTANDING ANYTHING IN THIS AGREEMENT OR OTHERWISE, NEITHER GAZZANG NOR ITS THIRD PARTY SUPPLIERS WILL BE LIABLE OR OBLIGATED WITH RESPECT TO ANY SUBJECT MATTER OF THIS AGREEMENT OR UNDER CONTRACT, NEGLIGENCE, STRICT LIABILITY OR ANY OTHER LEGAL OR EQUITABLE THEORY (I) FOR ANY AMOUNTS IN EXCESS IN THE AGGREGATE OF THE LICENSE FEES PAID TO GAZZANG OR ITS AUTHORIZED RESELLER BY LICENSEE WITH RESPECT TO THE SOFTWARE (AS EQUITABLY DETERMINED IN THE EVENT THE SOFTWARE IS BUNDLED WITH OTHER SOFTWARE DURING THE SIX MONTH PERIOD BEFORE THE CAUSE OF ACTION AROSE), (II) FOR ANY COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES OR RIGHTS; (III) FOR ANY INDIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL OR PUNITIVE DAMAGES; (IV) FOR INTERRUPTION OF USE OR LOSS OR CORRUPTION OF DATA; OR (V) FOR ANY MATTER BEYOND ITS REASONABLE CONTROL. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION AND EXCLUSIONS MAY NOT APPLY. NOTHING IN THIS AGREEMENT WILL LIMIT GAZZANG'S LIABILITY FOR FRAUD OR LIABILITY FOR DEATH OR PERSONAL INJURY CAUSED BY GAZZANG'S NEGLIGENCE. The provisions of this Agreement allocate the risks between Licensee and Gazzang. Gazzang's pricing reflects this allocation of risk and the limitations of liability specified herein.

8. Indemnification. Gazzang will defend and hold Licensee harmless from claims by third parties resulting from infringement by the Software of any United States patent or copyright or any misappropriation of any trade secret, provided Gazzang is promptly notified of any and all threats, claims and proceedings related thereto and given reasonable assistance and the opportunity to assume sole control over

defense and settlement; Gazzang will not be responsible for any settlement it does not approve in writing. The foregoing obligations do not apply with respect to Software or portions or components thereof (i) not supplied by Gazzang, (ii) that are modified by Licensee, (iii) combined with other products, processes or materials where the alleged infringement relates to such combination, (iv) where Licensee continues allegedly infringing activity after being notified thereof or after being informed of modifications that would have avoided the alleged infringement, or (v) where Licensee's use of such Software is not strictly in accordance with this Agreement. This provision will not survive the termination of this Agreement. If Licensee's use of any of the Software is, or in Gazzang's opinion is likely to be, enjoined due to the type of infringement specified in this Section 7, or if a claim is brought against Licensee due to the type of infringement specified in this Section 7, then Gazzang may, at its sole option and expense: (a) procure for Licensee the right to continue using such Software under the terms of this Agreement, (b) replace or modify such Software so that it is non-infringing and substantially equivalent or better in function to the enjoined Software, or (c) if options (a) and (b) above cannot be accomplished despite Gazzang's efforts, then Gazzang may terminate Licensee's rights and Gazzang's obligations hereunder with respect to such Software and remit to Licensee a prorata refund of the pre-paid License Fees for such Software corresponding to the portion of the then-current Subscription Term for such Software after the date of such termination.

9. Support and Maintenance. At no additional charge, Gazzang will provide Software maintenance and support services in accordance with Gazzang's standard Software Maintenance Program, set forth at www.gazzang.com during Gazzang's business hours.

PART III: TERMS AND CONDITIONS APPLICABLE TO THE PROVISION OF THE GAZZANG-HOSTED KEY STORAGE SYSTEM

1. Hosted Key Storage System Access. Subject to the terms and conditions of the Agreement, Gazzang will make the Hosted Key Storage System available to Licensee only (i) during the Subscription Term, (ii) by employees and consultants of Licensee, and (iii) for Licensee's internal business purposes and solely for use with the Software, as set forth in Gazzang's documentation, and may not be used by Licensee to provide services to third parties.

2. System Outages. This section is applicable to the delivery of the Hosted Key Storage System only. Gazzang will provide support 24 hours a day, 7 days a week to resolve emergency operational outages associated with the Hosted Key Storage System infrastructure, including server hardware and software, firewalls, load balancer and routers, systems administration, co-location services, and bandwidth provision, and critical, production-stopping errors in the Hosted Key Storage System.

3. Information Security. Gazzang will operate an information security program designed to protect Licensee data and utilizing industry standard policies and technologies. Gazzang will use third party hosting providers ("Third Party Hosting Providers") to host the Hosted Key Storage System. Gazzang shall ensure that any such third party hosting provider will have a current SAS 70 Type II report consisting of a comprehensive internal controls assessment report covering the internal controls and information security related to its hosting services, prepared by a third party auditor ("SAS 70 Type II Report"). In the event the SAS 70 Type II

Report is not reissued at least annually to the Third Party Hosting Provider being used for Licensee data, Gazzang agrees to replace such Third Party Hosting Provider as soon as practicable with another hosting services provider which has a current SAS 70 Type II Report.

4. Password and Encryption Key Administration. The Software administration tools used by Licensee to provide access to the Key Storage System will be password-protected and access by the Software to the Key Storage System will require encryption keys. Only Licensee personnel who have properly registered and received a login ID and password will be able to access the administration tools and encryption keys. Licensee will be solely responsible for administering, safeguarding and monitoring the use of login IDs and passwords and encryption keys. Upon the termination of employment of any personnel with access to administration tools and encryption keys, Licensee will immediately terminate access of the login ID of that individual to the administration tools and encryption keys. Gazzang will not recover any encryption keys if Licensee loses and is unable to recall any such encryption keys.

5. Customer Installed Key Storage System. In the event Customer purchases a version of the Software which includes a Customer-installed Key Storage System, the terms and conditions of such on-site Key Storage System license shall be as set forth in the On-Site Key Storage System Addendum at the URL referenced in the applicable Order.

PART IV: GENERAL TERMS AND CONDITIONS

1. Confidentiality. Each party acknowledges on its own behalf and on behalf of its officers, directors, employees, agents and consultants, and those of its affiliates ("Personnel"), that, during the term of this Agreement, it ("Receiving Party") may receive from or on behalf of the other party ("Disclosing Party") confidential and proprietary information relating to Disclosing Party ("Proprietary Information"). The Software and the documentation will be considered Gazzang's Proprietary Information. Business information, strategy, keys, operations information and related information disclosed by Licensee to Gazzang will be considered Licensee's Proprietary Information. During and after the term of this Agreement, Receiving Party will use the same degree of care to protect the Disclosing Party's Proprietary Information as it uses for its own Proprietary Information of like importance but in no event less than a reasonable standard of care. Proprietary Information will not include information that: (i) becomes public without breach of this Agreement by Receiving Party or its Personnel; (ii) was previously in the Receiving Party's possession (in written or other recorded form) with no obligation to maintain confidentiality; (iii) was received from a third party not under any obligation of confidentiality to Disclosing Party; or (iv) was developed by Receiving Party independently of, and without reference to, any Proprietary Information. Receiving Party will only permit access to Proprietary Information to those of its Personnel (a) who require access thereto for a purpose authorized by the Agreement and (b) who have signed confidentiality agreements or are otherwise bound by confidentiality obligations at least as restrictive as those contained herein.

2. Server. The term "Server" means a single computing system, including but not limited to a primary network server, a fail over server, or a virtual (or otherwise emulated) server, that is encrypted by the Software.

3. Restrictions. Except for one copy made solely for back-up purposes, Licensee may not copy the Software. Licensee must reproduce and include the copyright notice and any other notices that appear on the original Software on any copies and any media therefore. Licensee will not (and will not allow any third party to): (i) decompile, disassemble, or otherwise reverse engineer (except to the extent that applicable law prohibits reverse engineering restrictions) or attempt to reconstruct or discover any source code or underlying ideas or algorithms or file formats or programming or interoperability interfaces of the Software by any means whatsoever; (ii) remove any Software identification, copyright or other notices; (iii) provide, lease, lend, use for timesharing or service bureau purposes or otherwise use or allow others to use the Software to or for the benefit of third parties; (iv) modify, incorporate into or with other software or create a derivative work of any part of the Software; (v) disseminate performance information or analysis (including, without limitation, benchmarks) from any source relating to the Software; or (vi) remove or export from the United States or allow the export or re-export of any part of the Software or any direct Software thereof in violation of any restrictions, laws or regulations of the United States Department of Commerce, the United States Department of Treasury Office of Foreign Assets Control, or any other United States or foreign agency or authority. If a user of the software is an agency, department, or other entity of the United States Government ("Government"), the use, duplication, reproduction, release, modification, disclosure or transfer of the software, manuals, or any technical specifications, or any related documentation of any kind, including technical data, licensed in this Agreement, is restricted in accordance with Federal Acquisition Regulation ("FAR") 12.212 for civilian agencies and Defense Federal Acquisition Regulation Supplement ("DFARS") 227.7202 for military agencies. The Software and documentation licensed in this Agreement are commercial computer software and commercial computer software documentation. The use of the Software and documentation licensed under this Agreement is further restricted in accordance with the terms of this Agreement, or any modification thereto. The Software and documentation are licensed on the open market at market prices, and were developed entirely at private expense and without the use of any U.S. Government funds. Contractor/Manufacturer is Gazzang, Inc., 502 Baylor Street, Austin, Texas 78703.

4. Ownership. Notwithstanding anything else, as between Licensee and Gazzang, Gazzang retains all title to, and, except as expressly and unambiguously licensed herein, reserves all rights in the Software, all copies and derivative works thereof (by whomever made) and all related documentation and materials. Licensee acknowledges that Gazzang's third party licensors will be intended third party beneficiaries of this Agreement and will have the right to directly enforce against Licensee any Licensee obligations hereunder with respect to the Software to the extent the Software include licensed materials of such third party suppliers. Open source may be subject to additional license rights and restrictions, as set forth at [URL].

5. High Risk Activities. Licensee acknowledges that the Software is not intended for use in connection with any high risk or strict liability activity (including, without limitation, air travel, space travel, fire fighting, police operations, power plant operation, military operations, rescue operations, hospital and medical operations or the like) and Licensee agrees not to use or allow the use of the Software for or in connection with any such activity.

6. **Assignment.** Licensee may not assign or otherwise transfer in whole or in part or in any manner any rights, obligations, or any interest in or under this Agreement without Gazzang's prior written consent and any attempted assignment will be void. A merger or other acquisition by a third party will be treated as an assignment. Gazzang may at any time and without Licensee's consent assign all or a portion of its rights and duties under this Agreement. Subject to the foregoing, this Agreement will bind and inure to the benefit of the parties, their respective successors and permitted assigns.

7. **Open Source Licenses.** Any open source software provided hereunder will be provided pursuant to such open source software license terms and conditions. Upon reasonable notice to Licensee, Gazzang has the right to replace software provided to Licensee as part of open source software with software that has similar functionality. The license terms associated with open source software require that Gazzang provide copyright and license information to Licensee. A list of the open source software included in the Software or otherwise provided to Licensee and applicable license terms is available in Gazzang's on-line user documentation for the Software. Any provisions in this Agreement which differ from any open source software license are offered by Gazzang alone and not by any other party. In no event will the third party open source providers be liable for any special, direct, indirect, or consequential damages or any damages resulting from loss of use, data, or profits, whether in an action of contract, negligence, or other tortious action, arising out of or in connection with the use or performance of the open source software even if Gazzang or these providers have been advised of the possibility of such damages and whether or not such losses or damages are foreseeable.

8. **Term and Termination.** This Agreement is effective from the date Licensee downloads or installs the Software and will remain in force until terminated. The term of this Agreement with respect to Trial Software is governed by the key delivered with the Trial Software at the time of download. However, in no event will the term of the license for Trial Software be more than 30 days from the date of installation without express written approval from Gazzang. With respect to Software license under Part II above, the term of the Agreement will be for the agreed upon Subscription Term as set forth in the applicable order (normally, one year). Unless otherwise set forth in an Order, the Subscription Term will automatically renew at then-current License Fees, provided that if Licensee fails to pay the License Fees for such renewed Subscription Term on the renewal date or when due, if otherwise set forth in an Order, Gazzang may terminate this Agreement without notice and disable the Software. Licensee may terminate this Agreement at any time by destroying the documentation and the Software together with all copies and adaptations thereof. This Agreement will terminate immediately without notice from Gazzang if Licensee breaches this Agreement and fails to cure such breach within 30 days of notice from Gazzang. Upon termination of this

Agreement by Gazzang, Licensee will destroy all copies of the Software and documentation, and upon request, Licensee will certify such destruction to Gazzang. Part I Sections 2, 3, and 4, Part II Sections 4, 5, and 7, and Part IV Sections 1, 3, 4, 8, 9 and 10 of this Agreement will survive any termination hereof.

9. **Records and Inspection.** Licensee will conduct such internal audits as are reasonably required to verify continuing full compliance with this Agreement, maintain records with respect to Servers, and provide access to the Server locations or other applicable locations during normal business hours as requested by Gazzang on ten business days advance notice, from time to time, to permit personnel designated by Gazzang to verify such compliance.

10. **Miscellaneous.** This Agreement will be governed by, and interpreted in accordance with, the laws of the State of Texas (U.S.A.) exclusive of its choice of law provisions. This Agreement expressly excludes the United Nations Convention on Contracts for the International Sale of Goods. This Agreement sets forth the entire understanding and agreement between Licensee and Gazzang with respect to the subject matter hereof. NO VENDOR, DISTRIBUTOR, DEALER, RETAILER, SALES PERSON OR OTHER PERSON IS AUTHORIZED TO MODIFY THIS AGREEMENT OR TO MAKE ANY WARRANTY, REPRESENTATION OR PROMISE WHICH IS DIFFERENT THAN, OR IN ADDITION TO, THIS AGREEMENT ABOUT THE SOFTWARE OR ANY GAZZANG SERVICES. No waiver of any right under this Agreement will be effective unless in writing, signed by a duly authorized representative of Gazzang. Any modifications of this Agreement must be in writing and signed by both parties hereto. Each party will be and act as an independent contractor and not as an agent or partner of the other party for any purpose related to this Agreement. Neither party will have the authority to legally bind the other to any contract, proposal or other commitment or to incur any debt or create any liability on behalf of the other. Any notice required or permitted hereunder will be in writing and will be deemed to have been effectively given: (i) immediately upon personal delivery or facsimile transmission to the parties to be notified, (ii) one business day after deposit with a commercial overnight courier with tracking capabilities, or (iii) three days after deposit with the United States Postal Service, by registered or certified mail, postage prepaid to the respective addresses of the parties as set forth in the related electronic order. If any provision in this Agreement is held invalid or unenforceable, then that provision will be construed, limited, modified or, if necessary, severed, to the extent necessary, to eliminate its invalidity or unenforceability, and the other provisions of this Agreement will remain unaffected. Any pre-printed, additional or conflicting terms stated on purchase orders or acknowledgements of Licensee will be void and of no effect. English is the controlling language of this Agreement.

END