CloudBerry Lab

#1 Cross-Platform Cloud Backup

March 2018

CloudBerry Lab Security

CloudBerry product security information

CloudBerry Product Security Information

Preliminary Details

1.1 Document control

Date	Name	Comment
06/10/2015	Julia T	Initial revision
07/22/2015	Julia T	Updated
08/07/2015	Julia T	Updated
09/15/2017	David G	Updated
03/02/2018	Robert Z	Updated

1.2 Legal notice

Introduction

This document describes data encryption, secure access and transferring options that CloudBerry Lab uses in its products. It includes what encryption algorithm is used, encryption key generation and decryption procedures.

Overview

CloudBerry Lab provides robust encryption functionality. It allows our customers to transfer and keep their data in a secure way on the cloud storage. It supports the use of the FIPS-compliant encryption algorithms for data encryption. For data transference purposes, CloudBerry Lab provides an option for a secure network channel.

Implementation

2.1. Technical data model

Let's consider a case wherein user data is stored on a corporate network, and the user is going to configure a backup plan to the cloud storage:

1. First, you specify a cloud storage account like Amazon S3 that will store the backup data. The account credentials are stored in the application settings file that is encrypted using the AES-128 algorithm.

2. Then you create a backup plan that contains the backup destination, specify whether to use compression, encryption and other options (retention, notification, and schedule). All these settings are stored in a plan configuration file. If encryption is used, the entered password is stored in an encrypted way (AES-128).

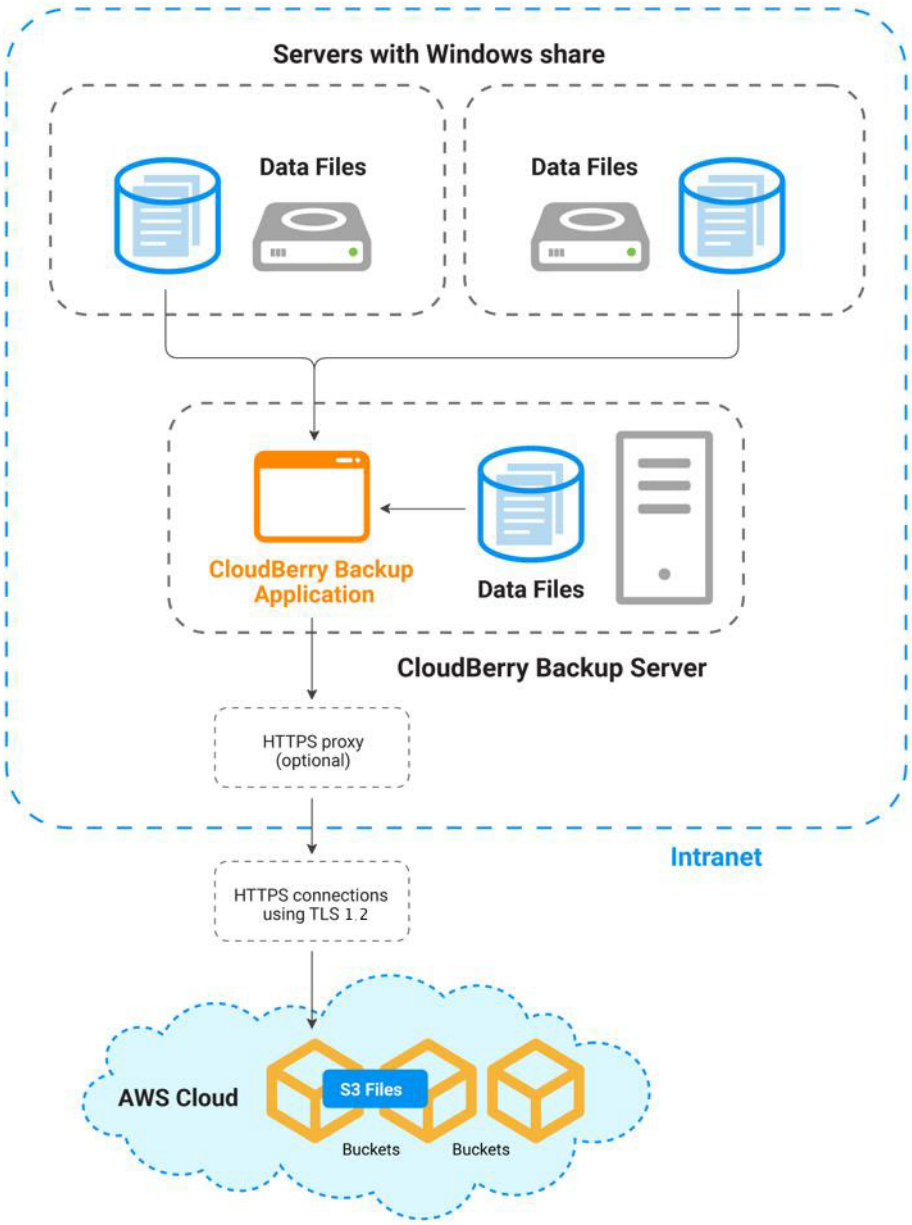
3. When you execute the first backup, it uploads all of the selected files and folders to the cloud storage (initial backup). When you execute subsequent backups, it scans selected folders and files and identifies if there are any new/modified files to perform an incremental backup.

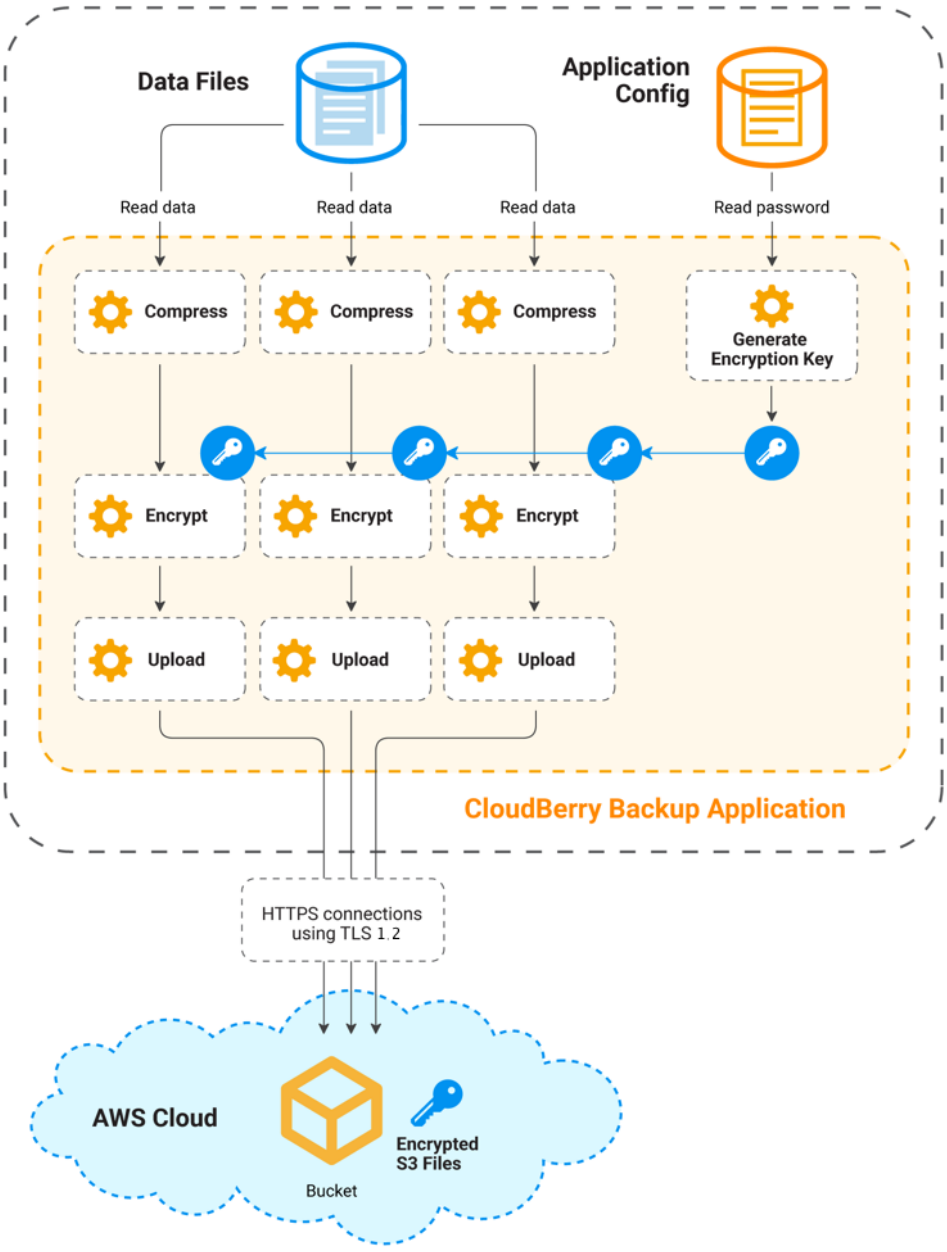
Note: we use a local repository (SQLite database) to track the list of backed up files. It allows to reduce a number of requests sent to the cloud storage. Namely, while scanning files and folders, we compare a list of files from the local repository (instead of making requests to the cloud) with your local data and then compose the list of files for the backup.

4. Once the list of files is ready, the plan starts uploading them into the cloud in multiple threads using HTTP(S) requests.

Note: to enable a secure channel, you need to select "Use SSL" option in the Storage Account dialog (Advanced settings).

Note: if you have a proxy server configured, you need to specify the proxy settings in Tools | Options |Proxy.Note: if you use compression/encryption, your files are compressed/encrypted prior to uploading. The following diagrams demonstrates the procedure:





5. CloudBerry Backup calculates an MD5 hash for each file you upload and sends this hash to the cloud storage service (e.g. Amazon S3). Amazon S3 calculates the hash on the server side and compares it with the hash we sent. If they do not match, it returns an error and upload fails. To enable MD5 check, you need to select the “Use MD5 checksum” option in Tools | Options | Connection.

Data Encryption

CloudBerry Lab supports concurrent encryption during uploading to the cloud storage as well as the following symmetric key encryption algorithms.

Our product supports the following encryption algorithms:

- AES-128;
- AES-192;
- AES-256;

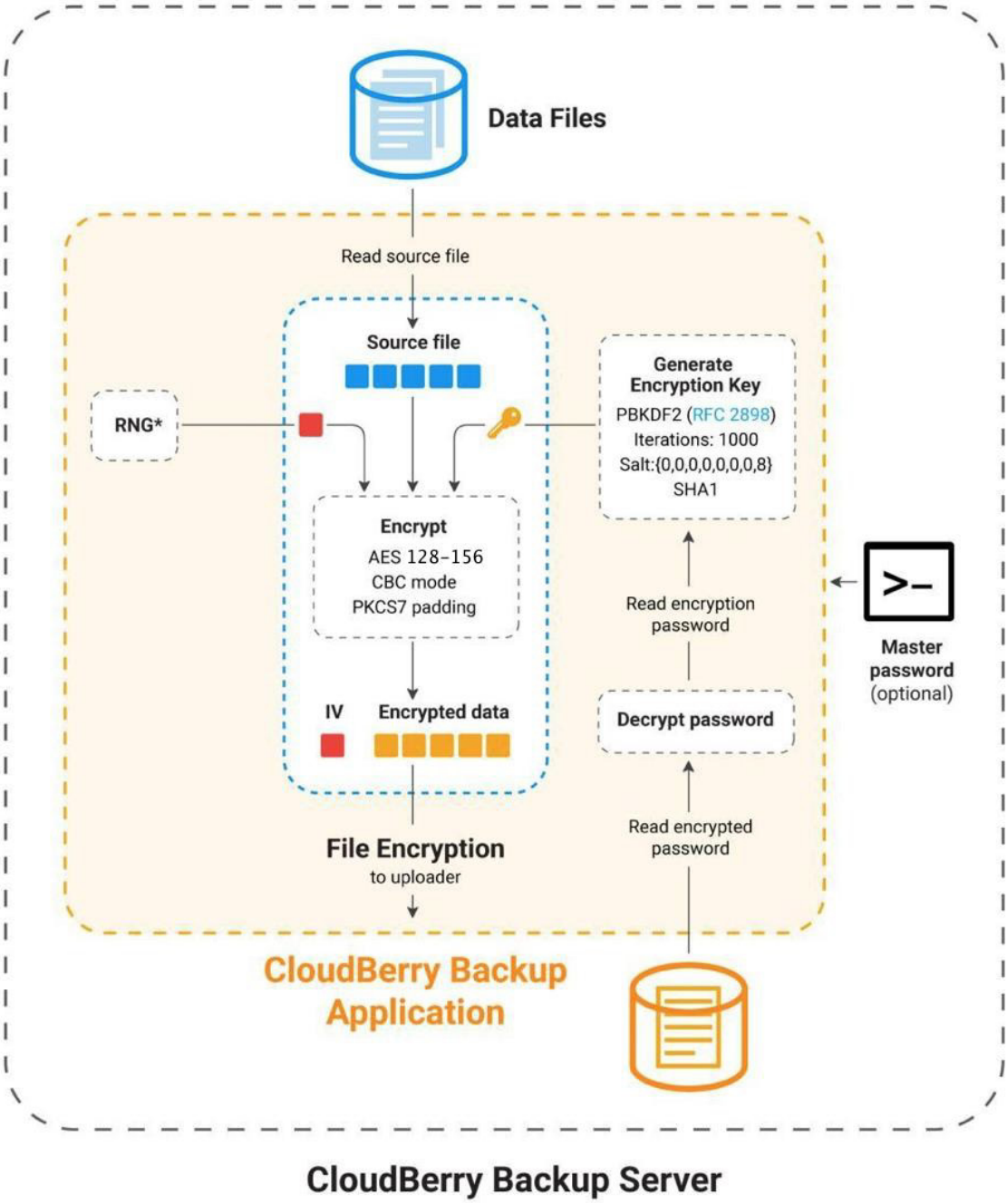
CloudBerry employs Microsoft's .NET API to use the aforementioned encryption algorithms.

Encryption Key

When encryption is enabled, the user is prompted to select an encryption algorithm and enter a password into the required text field. The password is subsequently converted into a cryptographic key and stored in the settings file. The password is encrypted using the AES-128 algorithm.

Data Flow

The following diagram illustrates the data flow:



*RNG —random number generator;

**IV – Initialization Vector (generated for each file).

Data Encryption Procedure

When the user enters a password into the GUI text field, this password is stored in the settings file that is encrypted using the AES-128 algorithm.

Upon upload, the program reads the encrypted user-provided password from the settings and generates an encryption key. To generate the encryption key, we use PBKDF2 (RFC2898): UTF8 (password), salt, iterations. Salt is a constant 8-byte zero array; iterations count is 1000.

The encryption algorithm is selected by the user in one the Backup Wizard's steps (e.g. AES-256). Encryption runs in CBC mode. Furthermore, we use PKCS7 padding.

For generating initialization vector (IV) we use Microsoft .NET RNGCryptoServiceProvider class. Encryption is performed "on the fly".

Finally, the cloud file has encryption metadata stored as an HTTP header: x- amz-meta-cb-encryptioninfo:

<version>;<sourcesize>;<algo>;<keysize_in_bits>;<Base64encodedIV>;;<Compression>;

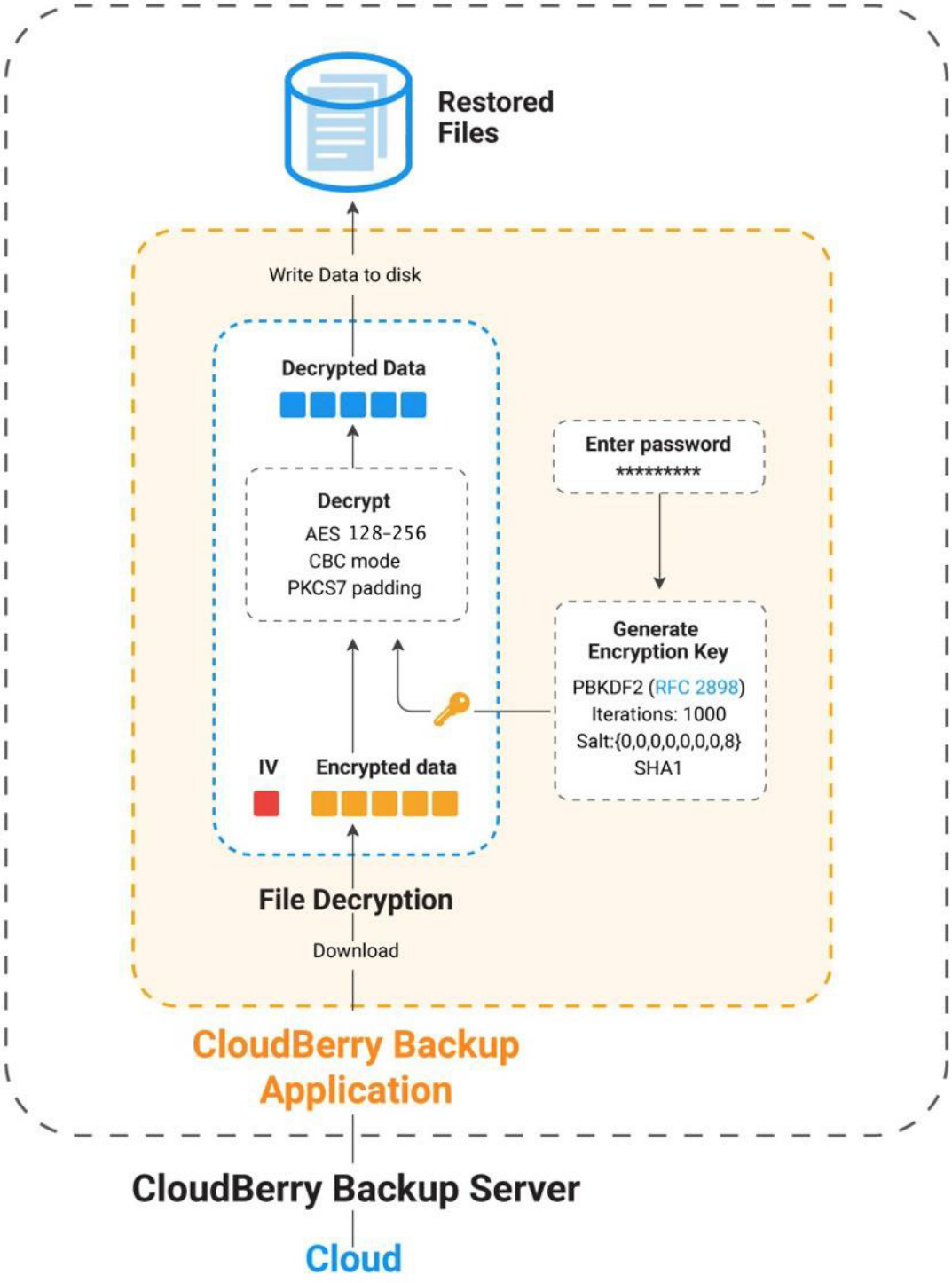
where:

- <version> - currently only version 1 is supported;
- <sourcesize> - an original file size (unencrypted/uncompressed);
- <algo> - one of these values: AES, 3DES, DES or RC2;
- <keysize_in_bits> - a size of the encryption key in bits. For example, for AES 128, the size is 128;
- <Base64encodedIV> - a Base64 encoded IV. It's different for each file. So there is no way to decrypt several files at once by specifying only one IV.
- <Compression> - either blank or GZip if file is compressed.

Data Decryption Procedure

1. When the user enters a password into the GUI dialog to download the encrypted file, the program generates an encryption key using the PBKDF2 function (RFC2898).

2. This key is used for decrypting the data.



3. We start downloading a file. Using the encryption algorithm (taken from the HTTP header x-amz-meta-cb-encryptioninfo) and the key we decrypt every downloaded portion of data.

4. Once the portion of data is decrypted, it is moved to the disk.

5. When the last portion of data has been decrypted, we check a padding. It allows to confirm that the entered password for decryption was correct. If password was incorrect, the disk data is discarded, and you get an error on restore.

Master Password

CloudBerry Lab also provides an additional option to protect from unauthorized access to CloudBerry Backup. You can set a master password in Tools | Options | General that will require entering a password to start working with CloudBerry Backup.