
ORACLE®

Feature rich BTRFS is getting richer with Encryption

An Overview

Anand Jain <anand.jain@oracle.com | anajain.sg@gmail.com>

Oracle

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Agenda

1. **BTRFS Features – Quick recap**
2. Encrypting Data at rest
3. BTRFS Encryption

BTRFS Features – RAS

- **Reliability -** Data and metadata are **checksum**med, scrub checks and fixes silent data corruptions or when read
- **Availability -** Redundancy – **RAID1/10/5/6** for data, metadata or both, (hot spare, auto replace), **COW** for FS to be consistency always (fsck should be dormant at some point)
- **Serviceability -** btrfs-image, btrfs-progs, logs, (sysfs), more to come like.. analytics

BTRFS Features – Datasets/Subvolumes

- **Subvolumes** and **(RW/RO) snapshots** - provides ability to create cheap independent FS with its **own properties**
- **Rollback** OS upgrades at the FS level
- **BTRFS send** and **receive** – provides **incremental backup** (or **remote replication** for sandbox testing), shorten recovery time and the recovery point

BTRFS Features – Transparent encoders

- **Transparent data encoders** – **smart compression**, choose either LZO (for speed, block compression algo) or ZLIB (good compression for wider variety of data). Get better IO performance and more disk space
- **Deduplication** – the space savers
- Upcoming **Encryption**

BTRFS Features – Security

- ACL
- SELinux
- Upcoming Encryption

BTRFS Features – additionally...

- Scalable 64 bit addressing up to **16 exabytes of volume or file size**
- **SSD optimization** - trim when blocks are no longer in use
- **Seed devices** - start with RO device add writable device to it
- **Separate worker** threads for CRC and Data encoders
- Per subvolume **Quota**
- **Easy management with btrfs(8) cli**

1. BTRFS Features
2. **Encrypting Data at rest**
3. BTRFS Encryption

Encrypting data at rest - Why

- - Makes it **harder** for the intruder to **use stolen data**
- - FS can be **maintained** and **serviced** by a **3rd party** data center without worrying about confidential information being leaked

Encrypting Data at rest – **Siri Bhoovalaya**, a **1000** years old manuscript ciphered in numbers

Proceedings of the 7th National Conference; INDIACom-2013
Computing For Nation Development, 7th – 8th March, 2013
Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi

An Inimitable Cryptographic Creation: Siri Bhoovalaya

Anil Kumar Jain
SMIEEE, FIETE, FIE(I),SMCSI
ak.jain@ieee.org

ABSTRACT
Muni Kumudendu, a great genius, created a unique treasure house of knowledge known as 'Siri Bhoovalaya' about a thousand years ago in Karnataka, India. The unique feature of this literature is that it is scripted in numerals only and enciphered in such manner that deciphering could be done in different fashions which result in plain-texts in different languages. One page of this epic corresponds to one 'Chakra'. A scheme for deciphering of Chakra is termed as 'Bandha'. Only 1270 of Chakras available this day. A Chakra is a 27x27

brilliance but also dexterity to put in great literature and enormous knowledge in it.

KEYWORDS
Chakra, Bandha, ChitraKavya, Substitution Cipher, Transposition Cipher, Steganography..

1. INTRODUCTION
Siri Bhoovalaya is an amazing piece of literature which is yet to be well discovered, assimilated and its content made

Encrypting Data at rest – the choices are

Level	Encryption Method	AlgoKEY	Pros (Cons)
APP / DB	MySQL AES_ENCRYPT() Oracle Transparent Data Encryption	AES/128bits AES/192bits	Highly granular (app overhead)
Kernel / FS	ecryptFS, ENCFS, [VFS EXT4]		
Kernel / Volume	LOOP_AES, DM_CRYPT, TRUECRYPT	AES128..256 AES256 AES256, Blowfish448, CAST5, Serpent, 3DES, Twofish	Easy setup (not granular, less flexible)
HW	T10 SCSI CBD	SP_IN [ ] and SP_OUT [ ]	
	LTO Tape Drive	AES-GCM 256	Less overhead (Spl HW)
	Self Encrypting Drive / SSD	AES-CBC 128/256	Performance (Spl HW, less flexible)

Encrypting Data at rest – by FS

ecryptFS	AES/128bits	Layered FS
EncFS	Blowfish/AES variable key size	FUSE
EXT4	AES256-XTS	(VFS)

1. BTRFS Features
2. Encrypting Data at rest
3. **BTRFS Encryption**

BTRFS Encryption - the aim

Provides the ability to encrypt sensitive application data on storage media completely transparent to the application

BTRFS Encryption – **Current** use case

- 2nd level of protection from data theft, (and first level is still the access control)

BTRFS Encryption – the setup

- Encryption **policy** is applied to **per subvolume**
- Policy includes – **Cipher algorithm**, **master-key keyring id**
- Stored in **extended attribute** with name '`btrfs.encrypt`', able to get or set using '**`setfattr(1)/getfattr(1)`**', (should migrate to use VFS encryption attributes)
- Default cipher algorithm is **CTR(AES) 128bits key** (also should provision for 256bits)

Create an encryption enabled btrfs subvolume :

```
btrfs subvolume create -e '<>' /btrfs/subvol1
```

BTRFS Encryption - Cipher algorithm

- Advanced Encryption Standard (**AES**)
- A symmetric cipher algorithm defined in the Federal Information Processing (**FIPS**) standard no. 197. AES provides 3 approved key lengths: 256, 192, and 128 bits.
- AES mode **CTR** (Counter) is a block cipher, but works **like a stream cipher**
- No crypto metadata is stored along with the data so **ciphertext** and **plaintext** sizes are of **same size**
- (XTS and ECC are being considered)

BTRFS Encryption - KEY management

- Uses **Keyring**, of type '**user**'
- Manage it using **btrfs**(8) or **keyctl**(1)

Check the **key status**:

```
btrfs subvolume show /btrfs/subvol1
```

Key in / Key out:

```
btrfs subvolume encrypt -k <in|out> /btrfs/subvol1
```

BTRFS Encryption – the operational

- Encrypts **extent pages** on the write path to the disk. Needs COW and paged IO
- **Master-key** is **requested** from the keyring; encrypts all file-data under an encrypt-enabled subvolume using master key and creates per file (will be random) iv; (which will be encrypted and added as an extended attribute to the inode)
- Encrypts **data only**, as of now
- Uses **asynchronous block cipher** KPIs

BTRFS Encryption – Per subvolume properties

- Extended attribute framework is enhanced to support **encryption specific attributes**
- Property **inheritance** by child is now depends on the type of child object

BTRFS Encryption – Encoder framework

- BTRFS **encoder framework** which is **enhanced** to support **crypto** framework
- More crypto tfm can be added
- Internally encryption it's a **type of encoder** along with compression, and so they are mutually exclusive (as of now)
- (Redundent page cleanups and optimizations are for later)

BTRFS Encryption - btrfs-progs

- 'btrfs subvolume create -**e**' is added
- 'btrfs subvolume show' shows the **status of the key**
- 'btrfs subvolume encrypt -k <**in|out**>' help to key in/out a subvolume
- (Code is in place to set/get 'btrfs property' sub cli, however 'btrfs property' cli needs a revamp, so setting encryption property through the properties-subcommand is disabled as of now)

BTRFS – Current focus

- Encryption
  - Stability
  - VFS layer integration
  - Other methods of key management like key on a usb drive
  - Other Cipher AES modes (CTS(CBC(AES))
- BTRFS Volume
  - A complete volume manager features

BTRFS Encryption – More to come, Targeted use cases

- **IaaS**; Third party system maintenance – provide **access to cipher-text –** a lot of challenges
- Backup or remote replication **without decipher**
- **Data in transit** could **leverage** encrypted data at rest
- **Asymmetric encryption,** private key on the subvolume
- **Data integrity checker**, public key on the subvolume

Thank You !

Thanks Oracle. Thanks LinuxCon JP

Q & A

anand.jain@oracle.com |
anand.sg@gmail.com

github.com/asj/BTRFS-Encryption

Hardware and Software

ORACLE

Engineered to Work Together


ORACLE®