Analysis of FileVault2: Apple's full disk encryption

Omar Choudary
University of Cambridge

Work in collaboration with
Felix Grobert and Joachim Metz

Security & Privacy

Show All

General | FileVault | Firewall | Privacy

FileVault secures the data on your disk by encrypting its contents. It automatically encrypts and decrypts your files while you're using them.

WARNING: You will need your login password or a recovery key to access your data. A recovery key is automatically generated as part of this setup. If you forget both your password and recovery key, the data will be lost.

FileVault is turned off for the disk "Macintosh HD".

Turn On FileVault...

Click the lock to prevent further changes.

?

Security & Privacy

Show All

The recovery key is a “safety net" which can be used to unlock the disk if you forget your password.

Make a copy and store it in a safe place. If you forget your password and lose the recovery key, all the data on your disk will be lost.

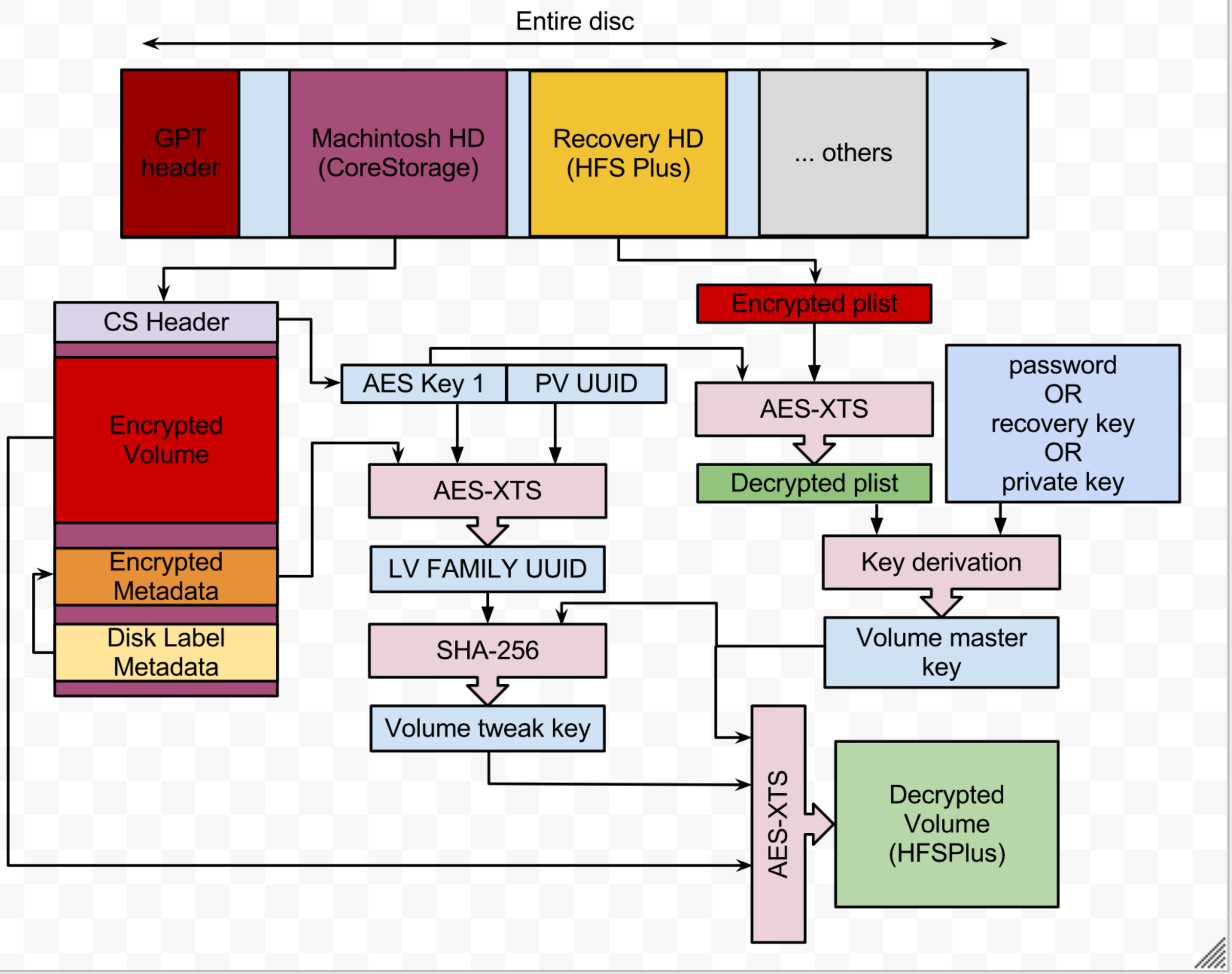
PBDV-QM67-YKPC-M5JB-OVVJ-BFO7

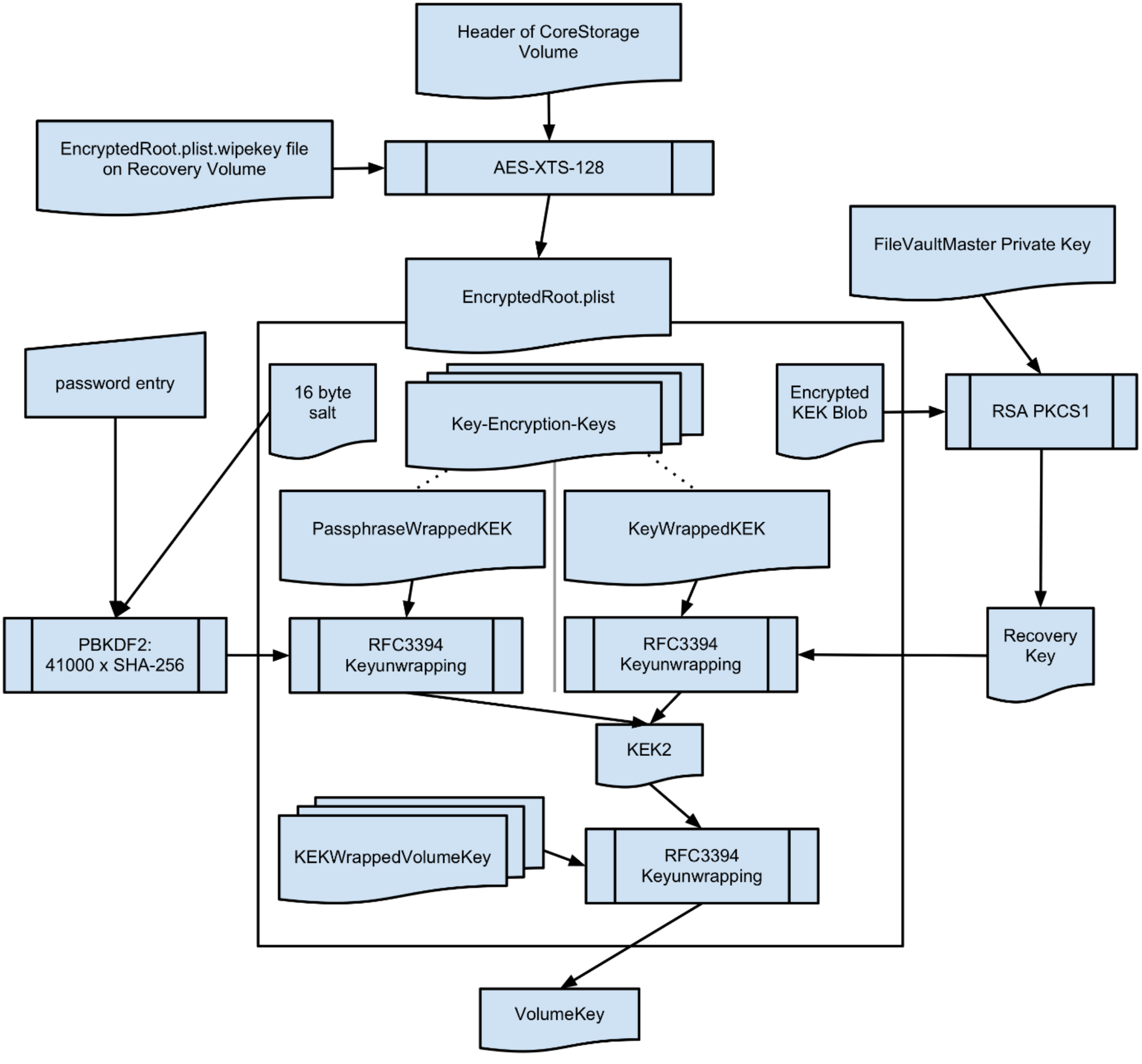
Cancel

Back

Continue

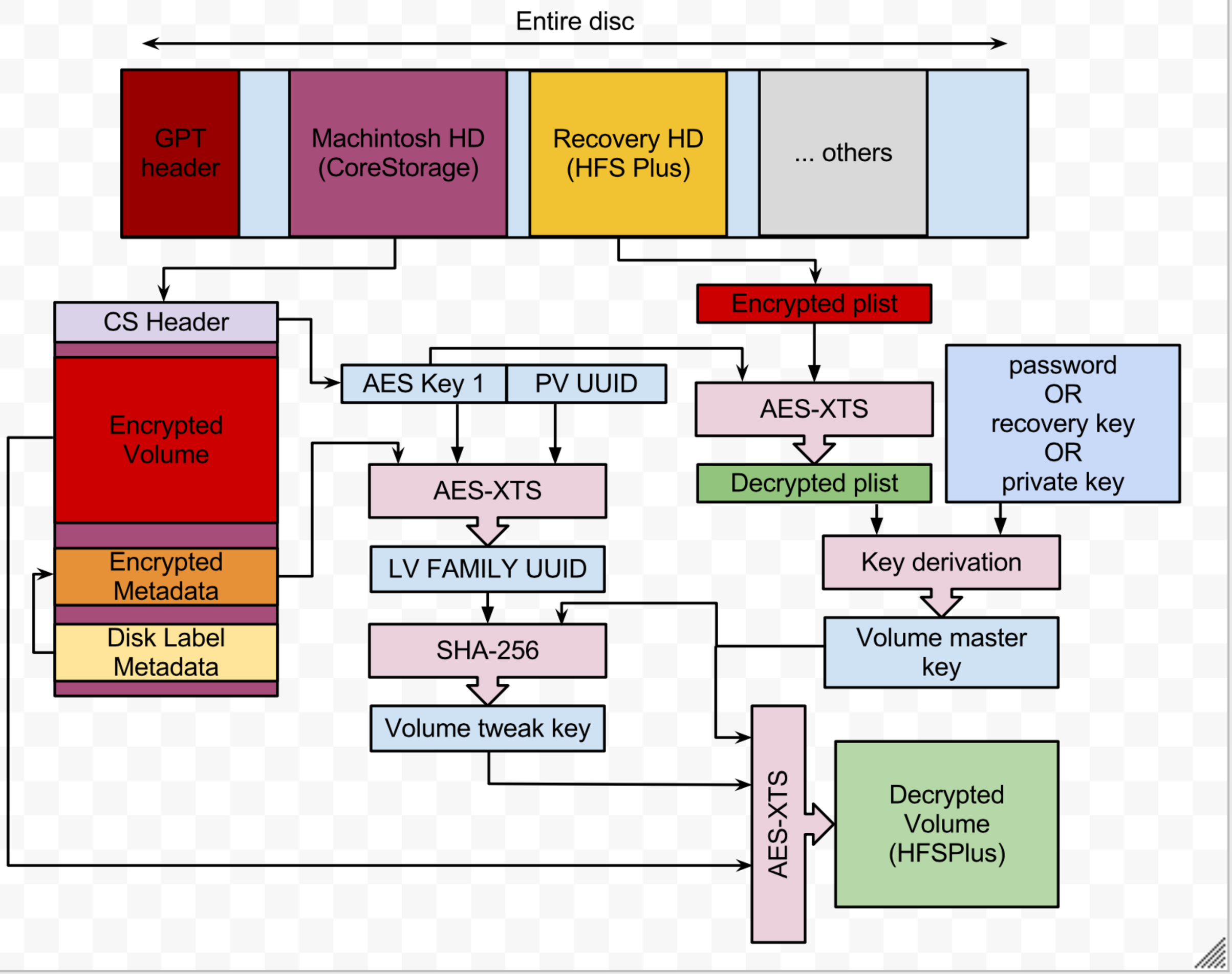
Click the lock to prevent further changes.

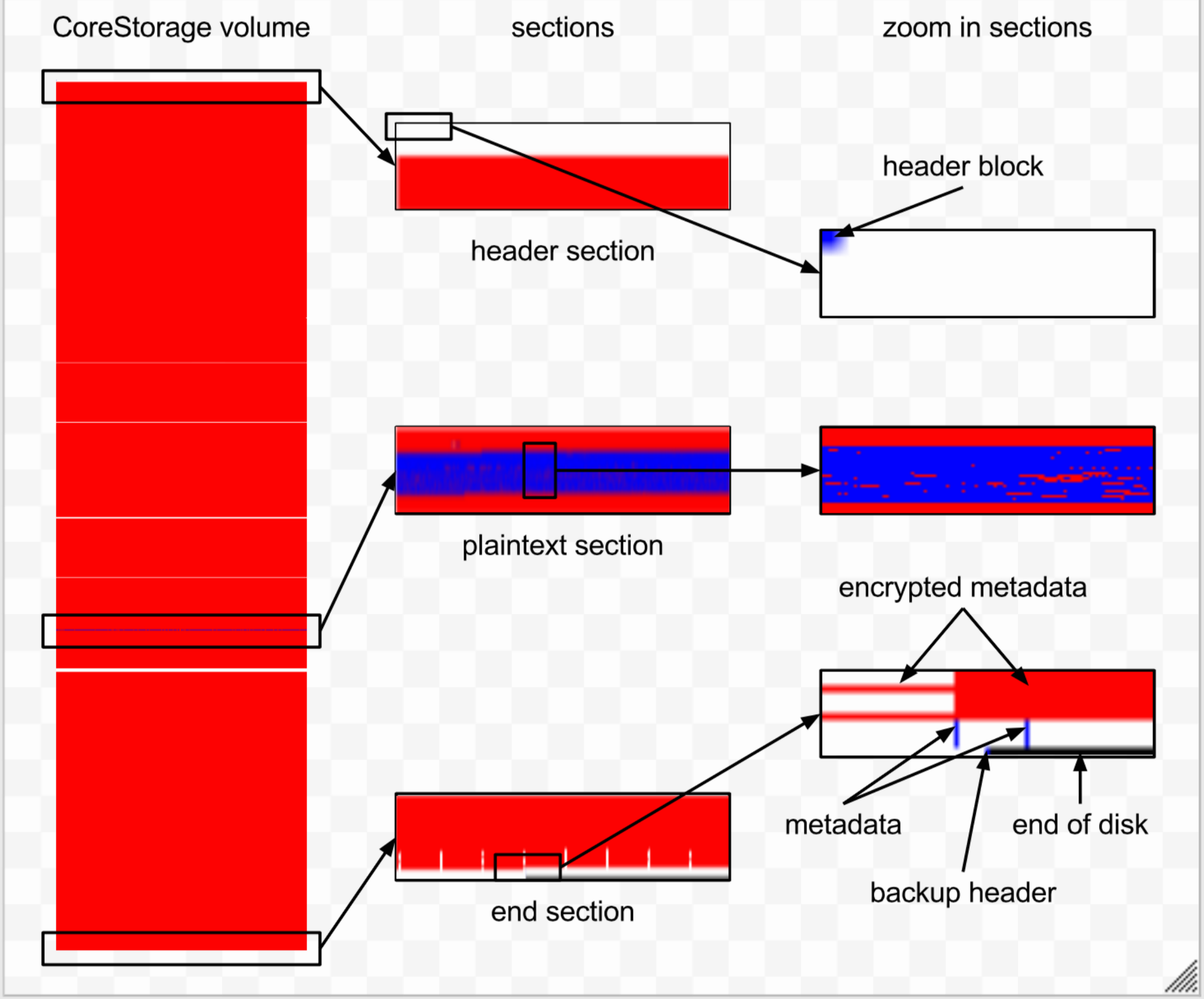

Entire disc
GPT header
Machintosh HD (CoreStorage)
Recovery HD (HFS Plus)
... others
CS Header
Encrypted Volume
Encrypted Metadata
Disk Label Metadata
AES Key 1
PV UUID
AES-XTS
LV FAMILY UUID
SHA-256
Volume tweak key
Encrypted plist
AES-XTS
Decrypted plist
password OR recovery key OR private key
Key derivation
Volume master key
AES-XTS
Decrypted Volume (HFSPlus)

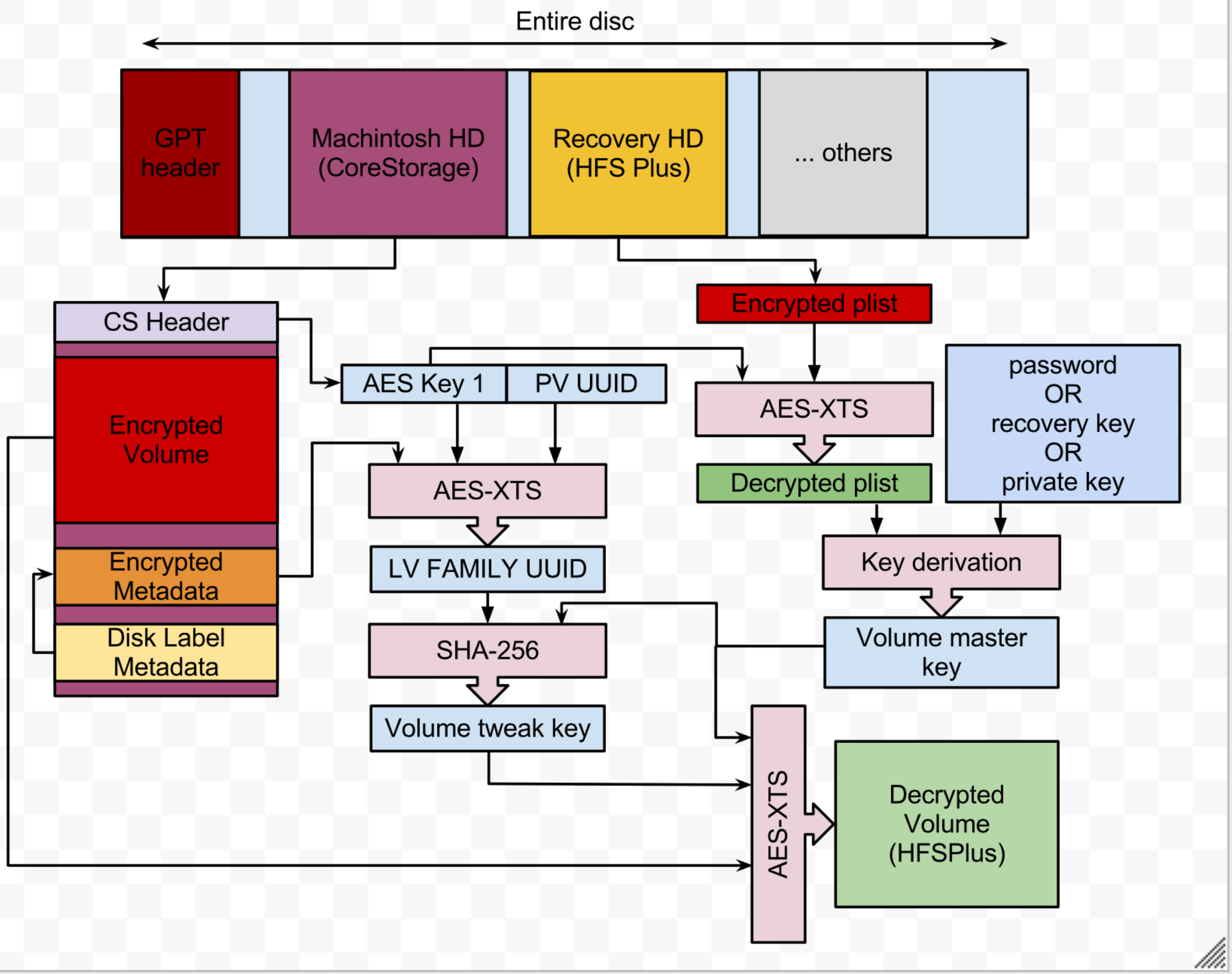

Header of CoreStorage Volume
EncryptedRoot.plist.wipekey file on Recovery Volume
AES-XTS-128
EncryptedRoot.plist
FileVaultMaster Private Key
password entry
16 byte salt
Key-Encryption-Keys
Encrypted KEK Blob
RSA PKCS1
PassphraseWrappedKEK
KeyWrappedKEK
PBKDF2: 41000 x SHA-256
RFC3394 Keyunwrapping
RFC3394 Keyunwrapping
Recovery Key
KEK2
KEKWrappedVolumeKey
RFC3394 Keyunwrapping
VolumeKey

The key derivation process can be summarised as:

```
p = get_user_password()
salt = get_salt_from_PassphraseWrappedKEK()
iterations = 41000
pk = pbkdf2(p, salt, iterations, HMAC-SHA256)
kek_wrapped = get_kek_from_PassphraseWrappedKEK()
kek = aes_unwrap(kek_wrapped, pk)
vmk_wrapped = get_vmk_from_KEKWrappedVolumeKey()
vmk = aes_unwrap(vmk_wrapped, kek)
```


Entire disc
GPT header
Machintosh HD (CoreStorage)
Recovery HD (HFS Plus)
... others
CS Header
Encrypted Volume
Encrypted Metadata
Disk Label Metadata
AES Key 1
PV UUID
AES-XTS
LV FAMILY UUID
SHA-256
Volume tweak key
Encrypted plist
AES-XTS
Decrypted plist
password OR recovery key OR private key
Key derivation
Volume master key
AES-XTS
Decrypted Volume (HFSPlus)


CoreStorage volume
sections
zoom in sections
header block
header section
plaintext section
encrypted metadata
metadata
end of disk
backup header
end section


Entire disc
GPT header
Machintosh HD (CoreStorage)
Recovery HD (HFS Plus)
... others
CS Header
Encrypted Volume
Encrypted Metadata
Disk Label Metadata
AES Key 1
PV UUID
AES-XTS
LV FAMILY UUID
SHA-256
Volume tweak key
Encrypted plist
AES-XTS
Decrypted plist
password OR recovery key OR private key
Key derivation
Volume master key
AES-XTS
Decrypted Volume (HFSPlus)

More details

- Check our paper: "Infiltrate the vault: security analysis and decryption of Lion full disk encryption"

- Available on ePrint: http://eprint.iacr.org/2012/374.pdf

- See also the open source software to mount encrypted volumes: http://code.google.com/p/libfvde/