

Sécurité des systèmes d'exploitation

Sécurité du stockage

Plan

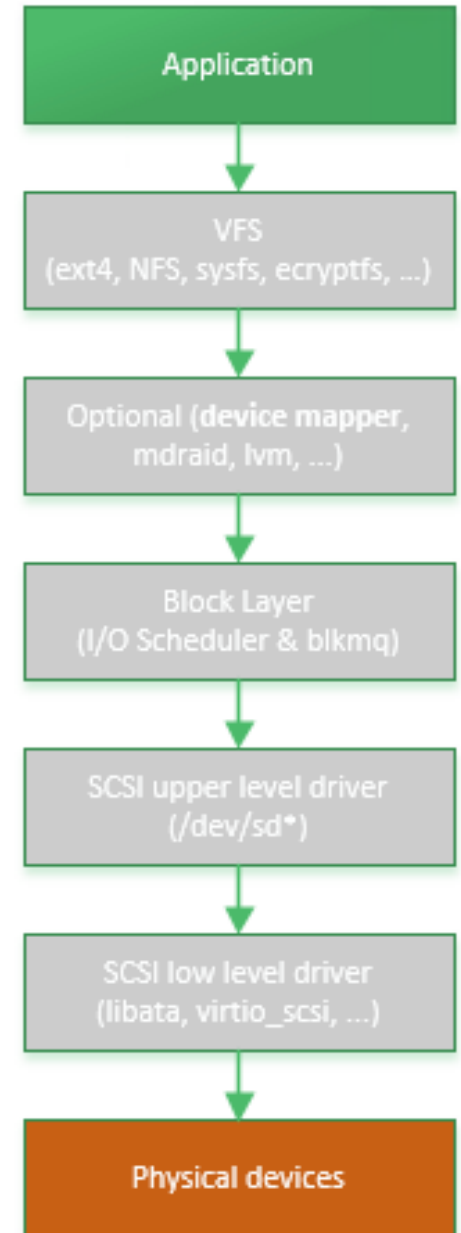
- Introduction
- Chiffrement
 - EFS
 - dm-crypt
 - Bitlocker
- Contrôle d'intégrité
- Disponibilité

Introduction

- Le contrôle d'accès contribue à la sécurité des données stockées
- Modèle de contrôle d'accès remis en cause si le stockage est accédé « à froid » (*data at rest*)
 - Limite de l'utilisation d'UID ou de SID
 - Duplication ou usurpation possible
 - Utilisation d'un programme ne tenant pas compte des informations de contrôle d'accès

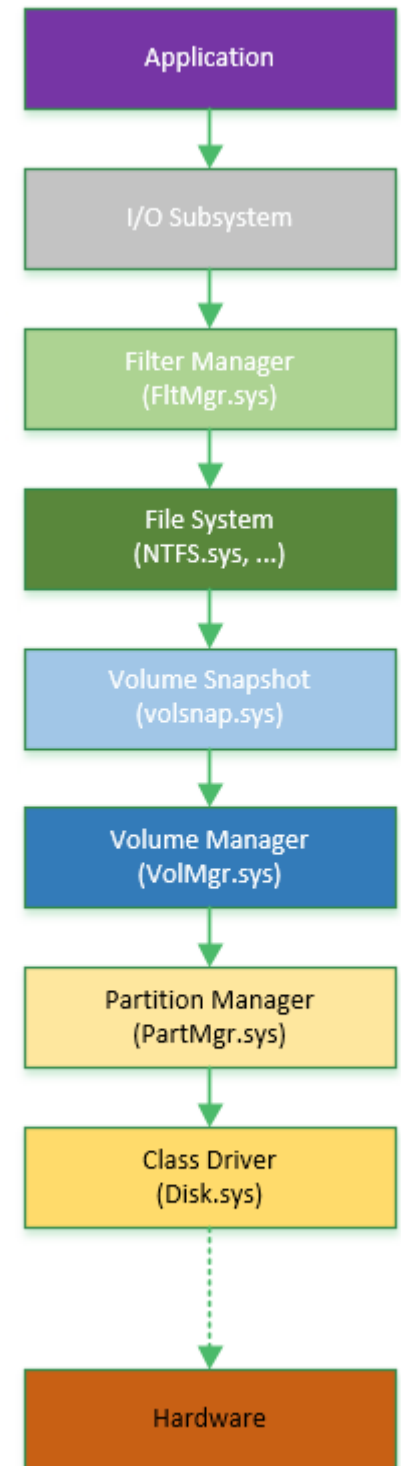
Introduction

- Le stockage sous Linux (simplifié)
 - Version complète
- Device Mapper
 - Pilote noyau qui fournit un *framework* pour la gestion de volume
 - Méthode générique pour créer des *mapped devices*, qui peuvent être utilisés comme des volumes logiques



Introduction

- Le stockage sous Windows (simplifié)
 - Explication détaillée
 - Les requêtes sont encapsulées dans des IRP (*I/O request packets*) avant d'être envoyées aux différents pilotes de la pile de stockage
 - Le *Filter Manager* permet de charger des pilotes spéciaux appelés mini-filtres



Chiffrement

- Assurer la confidentialité des données
- Plusieurs niveaux :
 - Chiffrement de fichier
 - Gestion des clés par l'utilisateur
 - Chaque chiffrement nécessite une action de l'utilisateur
 - Adapté au chiffrement de documents (envoi de pièces jointes)
 - Chiffrement au niveau du système de fichiers
 - Utilisation de méta-données pour stocker des éléments cryptographiques
 - Partage fin entre utilisateurs
 - Gestion des clés et des primitives cryptographiques « transparente »

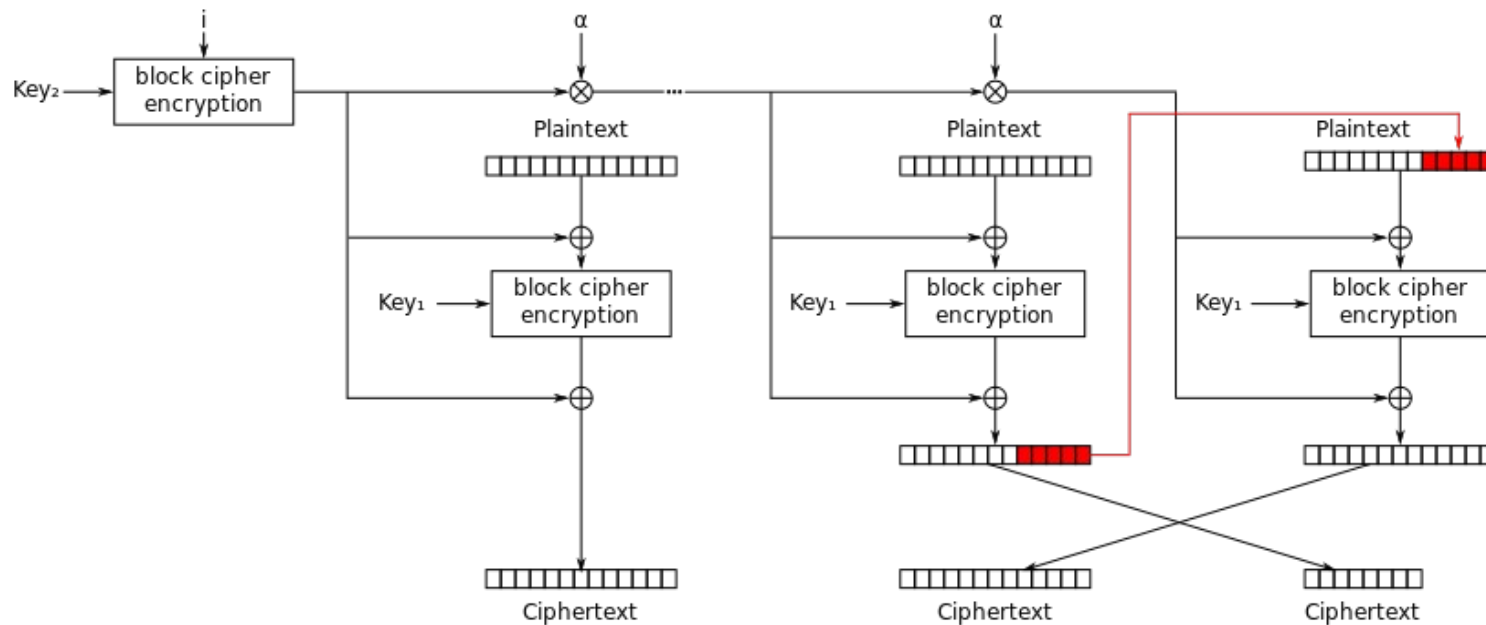
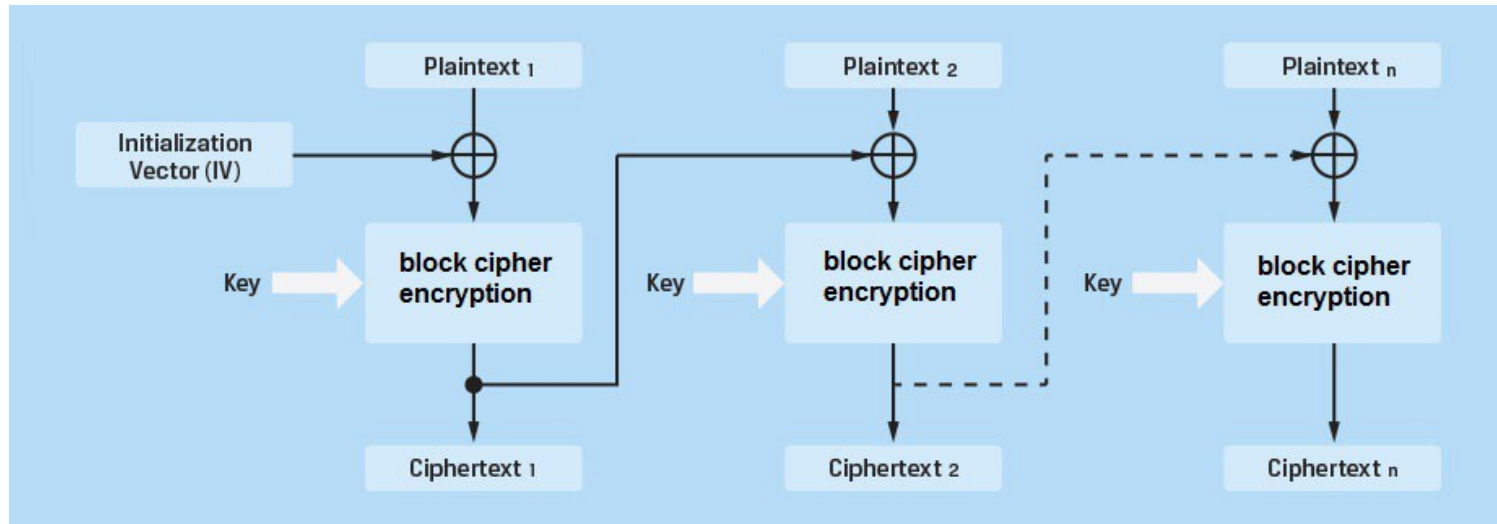
Chiffrement

- Plusieurs niveaux (suite) :
 - Chiffrement de volume/partition
 - Chiffrement transparent et complet de la partition
 - Protection en tout ou rien
 - Travaille au niveau bloc : pas de compréhension des données stockés
 - Efficace uniquement quand la machine est éteinte
 - Gestion des utilisateurs limitée ou complexe
 - Amène à réserver une machine à un seul utilisateur
 - Chiffrement matériel de périphérique
 - Dépend du contrôleur (par ex. support des normes IEEE 1667-2009 ou TCG OPAL)
 - Peu de garanties sur la gestion de clé
 - Produits parfois peu sécurisés : ex. disques [interne](#) et [externe](#)

Chiffrement

- Utilisation d'algorithmes de chiffrement symétrique pour le chiffrement des données
 - Critère performance
- Modes (chiffrement de volume/partition)
 - CBC avec un IV (*Initialization vector*) généralement dérivé du n° de secteur
 - XTS (XEX with tweak and ciphertext stealing)
 - Normalisé par IEEE (P1619/D16) en 2007, repris par le NIST (SP 800-38E) en 2010
 - Utilise 2 clés
 - par ex., pour utiliser la primitive AES-128, il faut fournir 256 bits

Chiffrement



XEX with tweak and ciphertext stealing (XTS) mode encryption

Source : Wikipedia

Chiffrement

- Chiffrement de système de fichiers : EFS (Encrypted File System)
- Introduit dans Windows 2000, intégré à NTFS
 - Pris en compte pour des partitions FAT depuis Windows 10 1709
- Chiffrement des fichiers avec l'algorithme AES depuis Windows XP SP1 (3DES et DESX auparavant)
- Gestion des accès utilisateurs
- Gestion des recouvrements

Chiffrement

- Le système génère une clé symétrique, appelée FEK (*File Encryption Key*), différente pour chaque fichier
 - Pour accéder au contenu d'un fichier, EFS déchiffre la FEK à l'aide de la clé privée de l'utilisateur puis utilise la FEK pour déchiffrer le fichier
 - Les FEK sont protégées par une paire de clé asymétrique propre à chaque utilisateur
- Un fichier chiffré se compose d'un en-tête (métadonnées EFS) et des données chiffrées
 - Les FEK chiffrées sont stockées dans l'en-tête de fichier

Chiffrement

- L'en-tête contient 2 types d'entrée

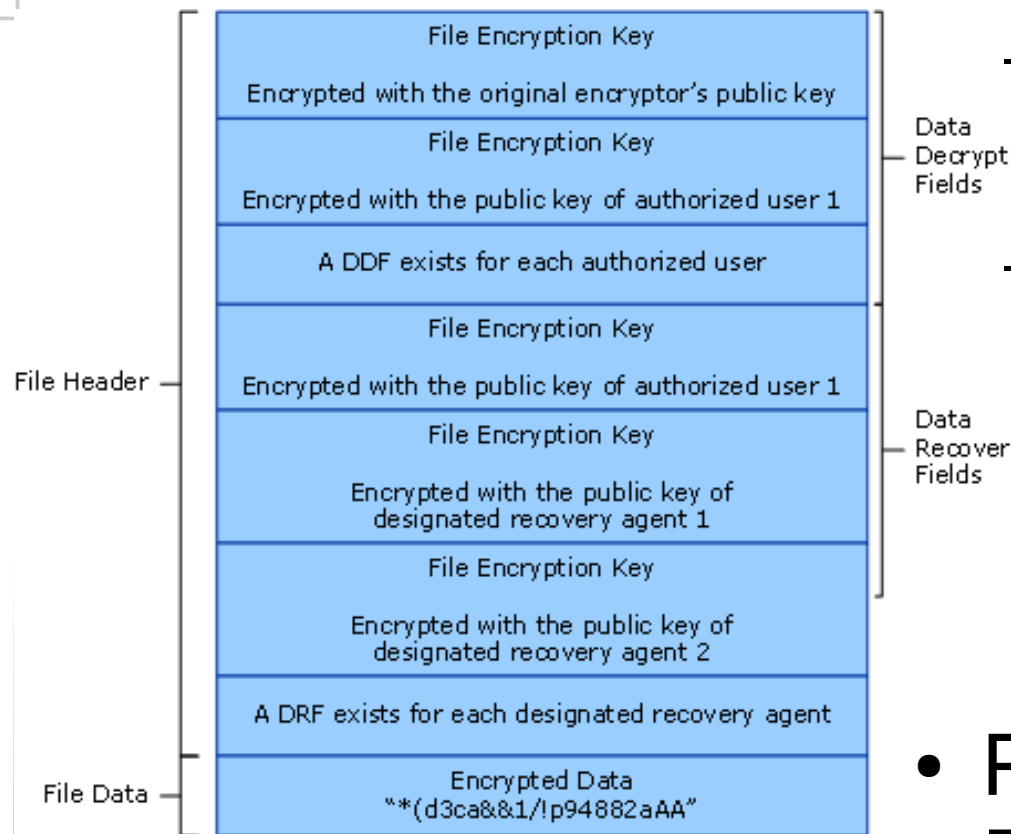
- Chiffrement/déchiffrement

- Une entrée par utilisateur

- Recouvrement

- Implique la définition d'un ou plusieurs agents de recouvrement dans la stratégie de sécurité de la machine

- Pour chaque entrée, la FEK est chiffrée avec la clé publique (utilisateur ou agent de recouvrement)



Source : Microsoft

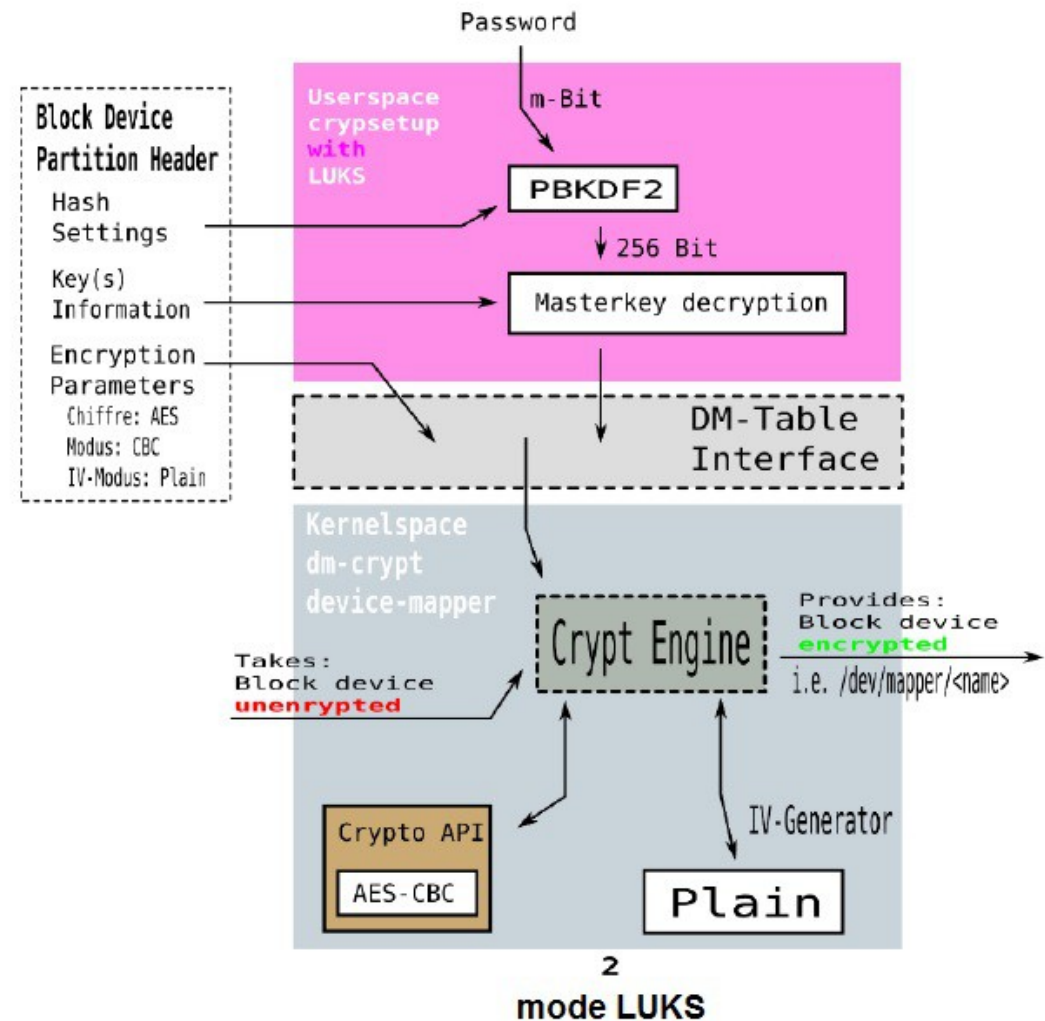
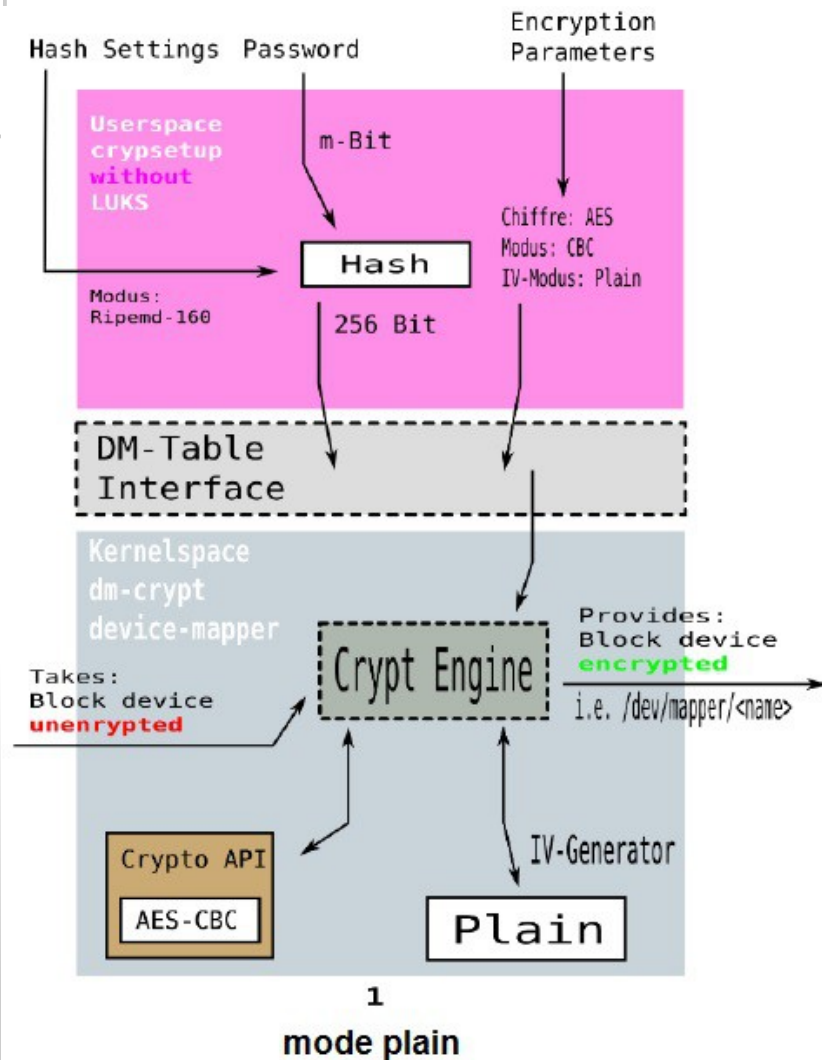
SSE - Sécurité du stockage



Chiffrement

- dm-crypt
 - Permet le chiffrement d'un périphérique en mode bloc géré par le *device mapper*
 - Partition (i.e. /dev/sda2)
 - Fichier contenant un système de fichiers
 - Intégré au noyau Linux depuis la version 2.6
 - `grep DM_CRYPT /boot/config-`uname -r``
 - Utilise l'API Crypto du noyau (voir /proc/crypto)
 - 2 modes de chiffrement
 - Mode « plain » : la clé maître est dérivée du mot de passe (*passphrase*)
 - Mode « LUKS » : jusqu'à 8 mots de passe peuvent déchiffrer la clé maître

Chiffrement



Chiffrement

- **LUKS** (Linux Unified Key Setup) supporte plusieurs algorithmes et modes de chiffrement
- Algorithmes : AES, SERPENT, Twofish, ...
- Modes principaux
 - ECB
 - CBC
 - Gestion des IV
 - Plain ou plain64 : numéro de secteur
 - **ESSIV** (*encrypted sector-salt initialization vector*) :
 - $IV(\text{secteur}) = E_{\text{sel}}(n^{\circ}\text{secteur})$ avec $\text{sel} = H(K)$
 - La fonction de hachage est précisée dans la dénomination du mode (par ex. cbc-essiv:sha256)
 - XTS utilise plain ou plain64 pour ses « IV »

Chiffrement

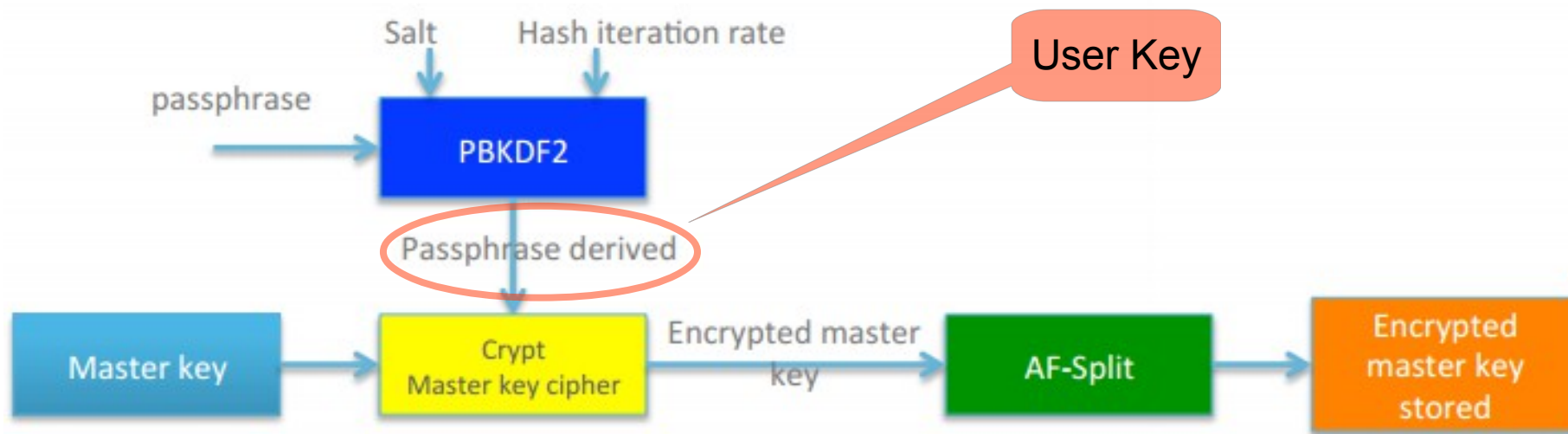
- Un volume chiffré LUKS est constitué de
 - Un en-tête de volume contenant notamment les algorithmes cryptographiques utilisés, le condensat PBKDF2 de clé maître (et les paramètres sel et nombre d'itérations)
 - 8 emplacements de clé (*key slot*)
 - Données chiffrées

Chiffrement

- Plusieurs clés
 - *Master Key* (MK) : utilisée pour chiffrer les données
 - Générée à partir d'une source d'aléa
 - *User Key* (UK) : utilisée pour déchiffrer la SMK correspondante
 - $UK = \text{PBKDF2}(\text{mdp_utilisateur}, \text{sel}, \text{nb_itération}, \text{taille_clé})$
 - Le sel et le nombre d'itération sont propres à chaque UK, ils sont différents de ceux de l'en-tête LUKS
 - *Split Master Key* (SMK) : stockée après l'en-tête LUKS (dans un des 8 emplacements)
 - La MK est obtenu à partir d'une SMK
 - La MK est vérifiée à partir d'un hash PBKDF2 de référence stocké dans l'en-tête LUKS

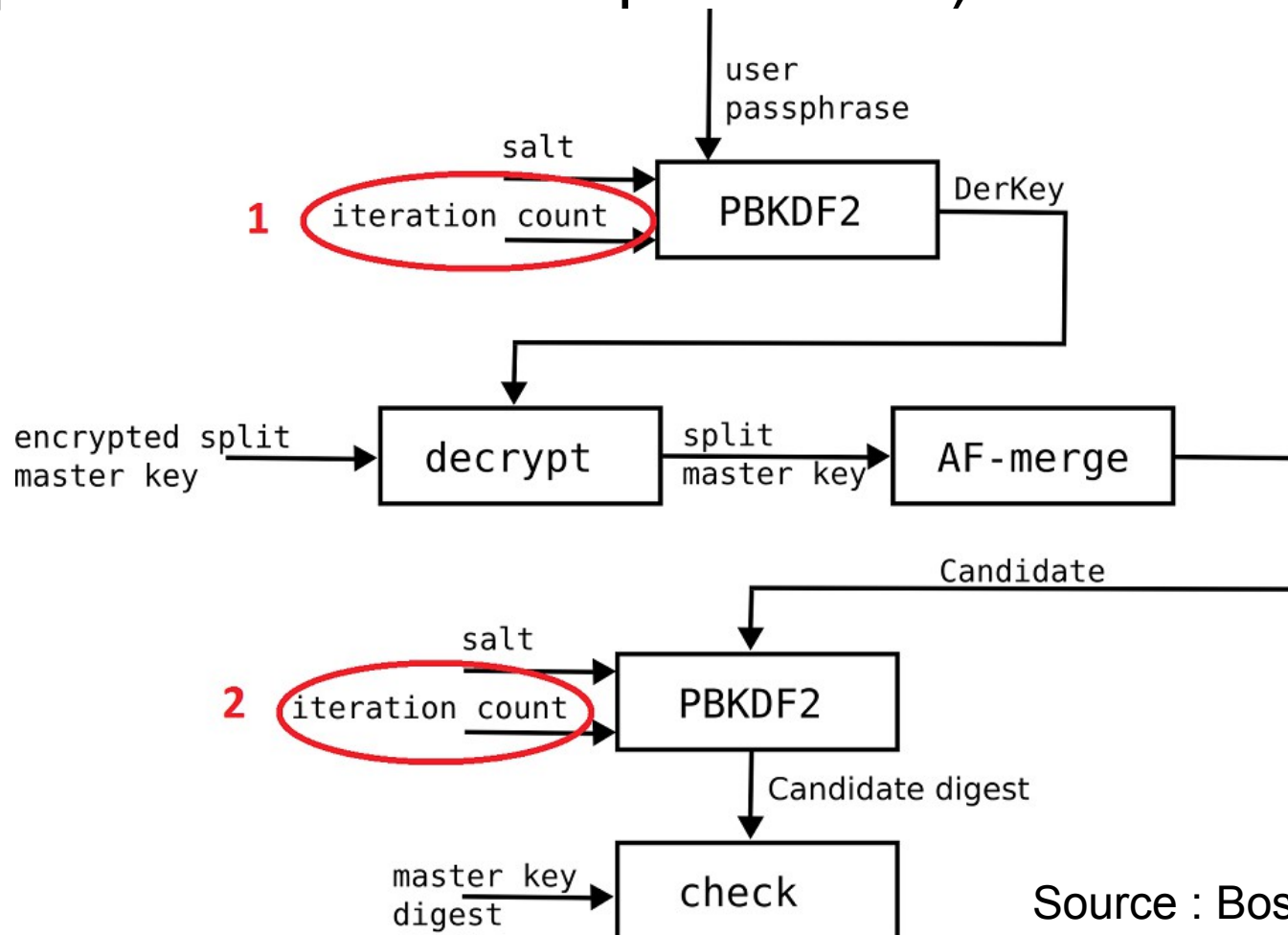
Chiffrement

- Processus de stockage de la MK
 - Pour un emplacement donné



Chiffrement

- Processus d'obtention de la MK
 - Fait emplacement par emplacement (pas de lien explicite utilisateur/emplacement)



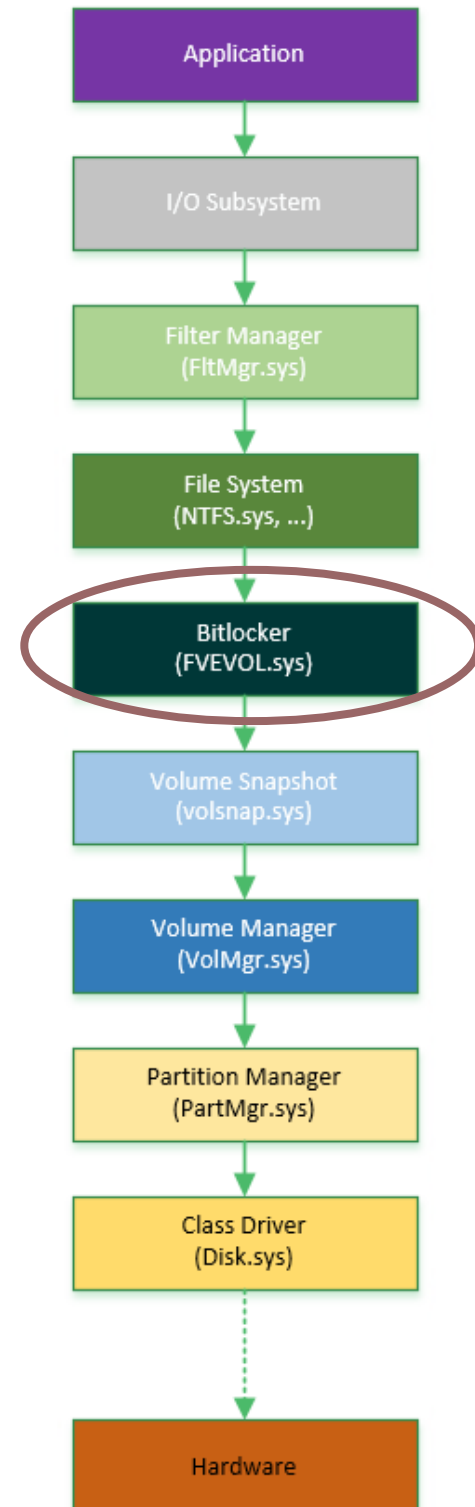
Source : Bossi et al.

Chiffrement

- Chiffrement de volume/partition : Bitlocker
 - Windows Vista & 7 : disponible uniquement dans les versions Enterprise et Ultimate ;
 - depuis Windows 8 : disponible à partir de la version Professional ;
 - Windows Server \geq 2008
- Depuis Windows 7, chiffrement des supports amovibles : « BitLocker-To-Go »
- Chiffrement d'un volume existant sans perte de données
- Mécanisme de recouvrement

Chiffrement

- Se présente sous forme d'un pilote inséré dans la pile de stockage
 - FVE (Full Volume Encryption)
 - en principe indépendant du système de fichiers
 - Support de NTFS (depuis Windows Vista), de FAT, FAT32 et exFAT (depuis Windows 7) et ReFS (depuis Windows Server 2012)
- Chiffrement AES (128 ou 256 bits)
 - Mode CBC, avec ou sans *Elephant diffuser*
 - Mode XTS depuis Windows 10 1511



Chiffrement

- Clés cryptographiques
 - VMK (*Volume Master Key*) : clé de 256 bits servant à protéger (chiffrer) la FVEK
 - Il est possible de la changer
 - FVEK (*Full Volume Encryption Key*) : clé utilisée pour le chiffrement du volume
 - Tailles : 128 ou 256 bits (256 ou 512 bits dans le cas d'utilisation du mode XTS)
 - Clés stockées sous une forme chiffrée sur le volume système
 - L'accès à la VMK se fait à l'aide d'un protecteur

Chiffrement

- Protectors Bitlocker
 - TPM, TPM + code PIN, TPM + code PIN + clé de démarrage, TPM + clé de démarrage, clé de démarrage, carte à puce, mot de passe (pas pour un volume système), clé ou mot de passe de récupération
 - Lors de l'utilisation du TPM, le descellement de la clé SRK se fait sur la base des mesures effectuées

Scenario	VMK blob	Algorithm used to encrypt VMK
Default (TPM-only)	SRK(VMK)	RSA
TPM and PIN	(SRK+SHA256(PIN))(VMK)	RSA
TPM and PIN and USB	XOR((SRK+SHA256(PIN)),SK)(VMK)	AES
TPM and USB (TPM+SK)	XOR(SRK(IK),SK)(VMK)	AES
Startup key (SK)	SK(VMK)	AES
Recovery key (RK)	RK(VMK)	AES
Recovery password ¹	(Chained-hashing>Password),Salt)(VMK)	AES
Data volume password ¹	(Chained-hashing>Password),Salt)(VMK)	AES
Public-key-based	IK(VMK) where IK is RSA or ECC-encrypted with the PK	AES
Clear key (CC)	CC(VMK)	AES
Auto-unlock key (AUK)	OS_VMK(IK(VMK)) or user_PK (form user cert store)	AES

Source : Microsoft

Chiffrement

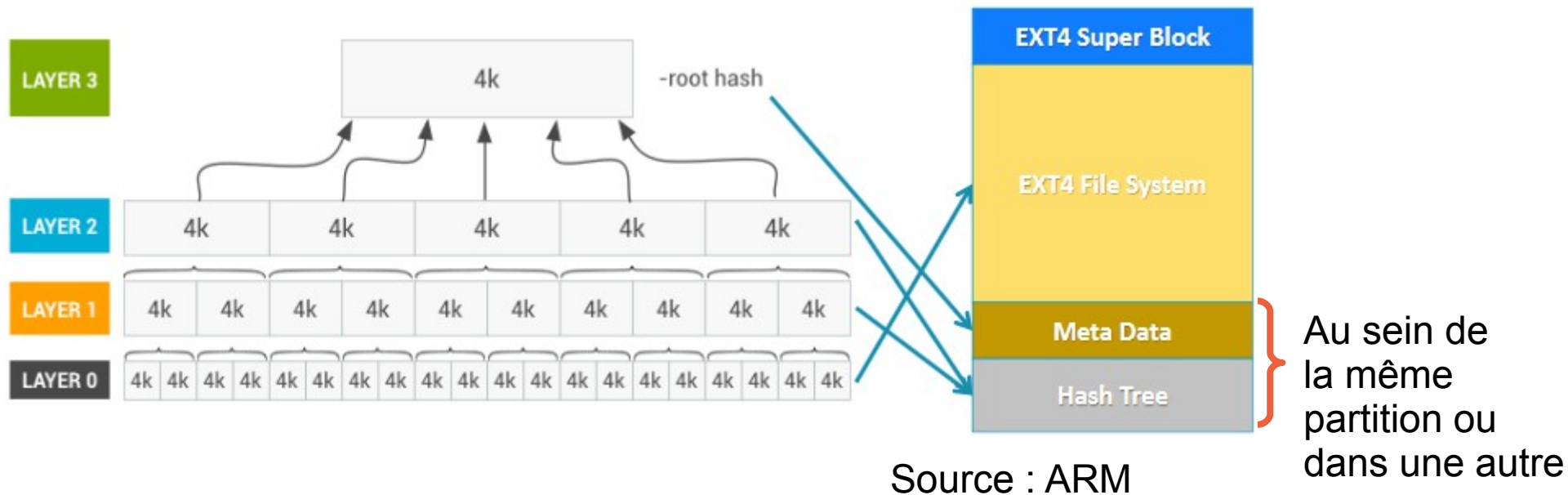
- Attaques sur les solutions de chiffrement de volume
 - Utilisation de bootkit ou de rootkit
 - « Evil Maid » [attack](#)
 - Bruteforce sur le mot de passe ou la clé de chiffrement
 - Attaques DMA
 - Cold boot attack
 - Spécifique Windows : fichier hyberfil.sys

Contrôle d'intégrité

- Assurer la confiance envers les données
- Plusieurs niveaux :
 - Fichier
 - un condensat (cryptographique) est calculé périodiquement pour un fichier donné et est comparé avec une base de référence
 - Approche suivie par des logiciels du type AIDE, OSSEC, Tripwire, ...
 - Système de fichiers
 - Mis en place pour un objectif de contrôle de la corruption via des sommes de contrôles
 - Btrfs
 - Volume/partition

Contrôle d'intégrité

- Dm-verity fournit un contrôle d'intégrité transparent d'un périphérique bloc
- Utilisation d'un arbre de Merkle
 - Un bloc de hash de 4ko de la couche 1 stocke les hash (SHA256) de 128 bloc de données



Contrôle d'intégrité

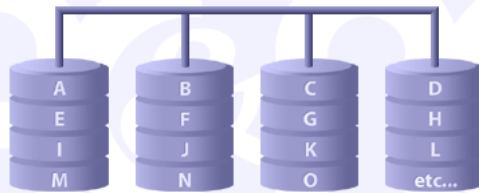
- La sécurité du hash racine est primordiale
- Inconvénient lié à la lecture-seule
 - Nécessite de remonter la partition sous-jacente en RW, faire la modification, calculer le nouvel arbre puis remonter le tout avec dm-verity
 - Noyau ≥ 4.12 : dm-integrity (peut être combiné avec dm-crypt)

Disponibilité

- 2 approches
 - Sauvegarde/archivage
 - Redondance (« RAID » logiciel), développer de la résilience face à une panne matérielle
 - Systèmes à base de noyau Linux
 - LVM supporte les niveaux RAID 1, 4, 5, 6 et 10
 - RAID0 est également supporté mais n'offre aucune redondance
 - *device mapper* a une cible *mirror* (même principe que RAID1)
 - Systèmes Windows
 - Mise en miroir de volume, utilisation de mécanismes de parité, y compris en mode RAID5
 - Fonctions disponibles sur les versions >= Pro et Server
 - *Storage Spaces* ajoute un mode miroir à 3 disques (2 copies)

Disponibilité

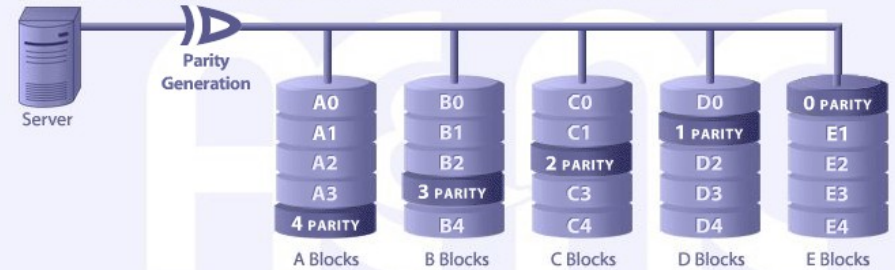
RAID LEVEL 0 : Striped Disk Array without Fault Tolerance



Copyright © 1996 - 2004 Advanced Computer & Network Corporation. All Rights Reserved.

ACNC
www.acnc.com

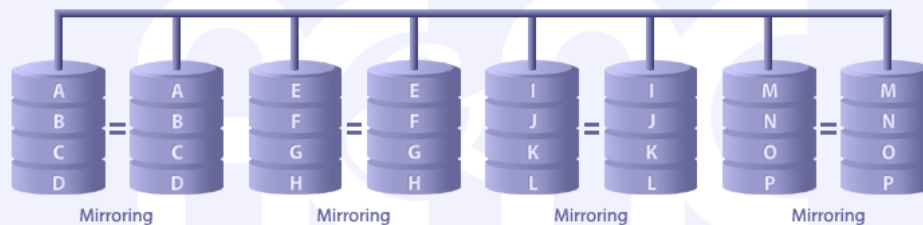
RAID LEVEL 5 : Independent Data Disks with Distributed Parity Blocks



Copyright © 1996 - 2005 Advanced Computer & Network Corporation. All Rights Reserved.

ACNC
www.acnc.com

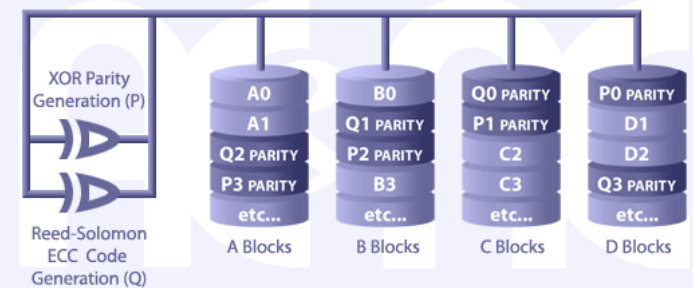
RAID LEVEL 1 : Mirroring & Duplexing



Copyright © 1996 - 2004 Advanced Computer & Network Corporation. All Rights Reserved.

ACNC
www.acnc.com

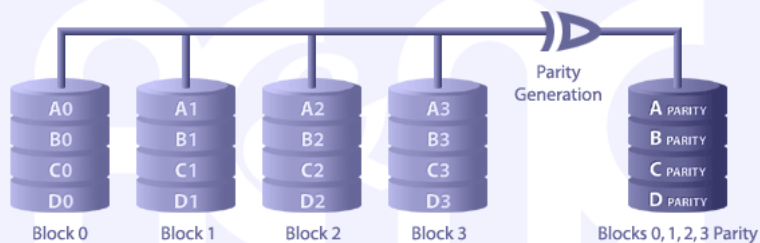
RAID LEVEL 6 : Independent Data Disks with Two Independent Distributed Parity Schemes



Copyright © 1996 - 2005 Advanced Computer & Network Corporation. All Rights Reserved.

ACNC
www.acnc.com

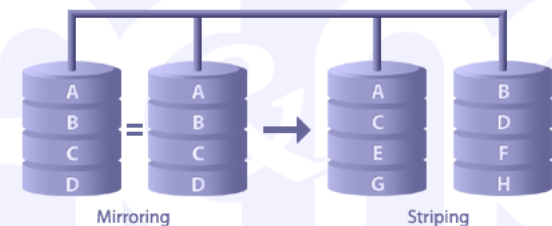
RAID LEVEL 4 : Independent Data Disks with Shared Parity Disk



Copyright © 1996 - 2004 Advanced Computer & Network Corporation. All Rights Reserved.

ACNC
www.acnc.com

RAID LEVEL 10 : Very High Reliability Combined with High Performance



Copyright © 1996 - 2004 Advanced Computer & Network Corporation. All Rights Reserved.

ACNC
www.acnc.com

Pour approfondir

- What users should know about Full Disk Encryption based on LUKS, Simone Bossi and Andrea Visconti, 2016
- Evaluation of Some Blockcipher Modes of Operation, Phillip Rogaway, 2011
- VeraCrypt 1.18 Security Assessment, Quarkslab
- BitLocker, Aurélien Bordes, SSTIC 2011
- An in-depth analysis of the cold boot attack, Carbone et al., 2011

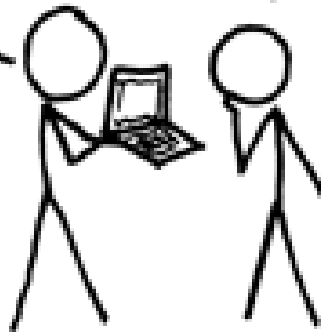
Questions ?

A CRYPTO NERD'S IMAGINATION:

HIS LAPTOP'S ENCRYPTED.
LET'S BUILD A MILLION-DOLLAR
CLUSTER TO CRACK IT.

BLAST! OUR
EVIL PLAN
IS FOILED!

NO GOOD! IT'S
4096-BIT RSA!



WHAT WOULD ACTUALLY HAPPEN:

HIS LAPTOP'S ENCRYPTED.
DRUG HIM AND HIT HIM WITH
THIS \$5 WRENCH UNTIL
HE TELLS US THE PASSWORD.

GOT IT.

