Aalto University

Encrypting stored data

Tuomas Aura

CS-C3130 Information security

Aalto University, autumn 2020

Outline

1. Scenarios for data encryption
2. File encryption
3. Encrypting file system EFS
4. Full disk encryption: BitLocker
5. Data recovery

This lecture uses Windows as a case study.

Simple application of cryptography — and a good example of how difficult it is to build a *secure system*

Acknowledgement:
These slides are partly based on Microsoft material

Data encryption

- Scenarios:
  - lost and stolen notebook computers
  - stolen workstations and servers
  - decommissioning hard drives

  → Risk of disclosure of confidential data
- The obvious solution: encrypt data on drive

However, security often conflicts with usability, reliability and deployability. Also, system design and implementation may have weaknesses.

Scenarios for data encryption

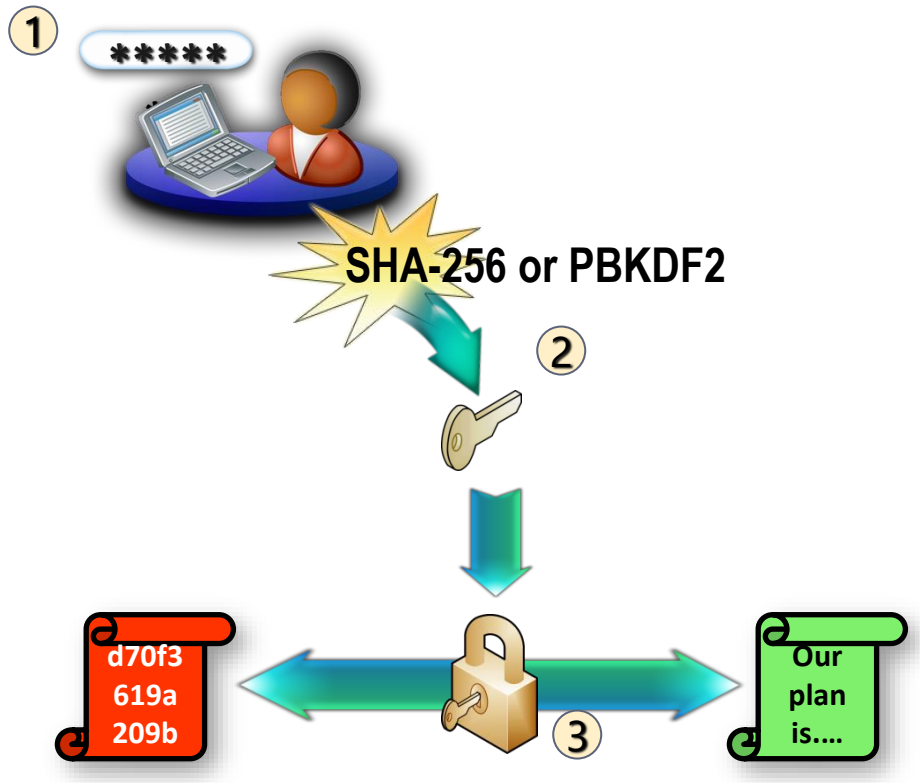
- Lost and stolen laptops
  - Contain confidential data and access credentials
- Physically compromised servers
  - Contain business secrets, customer data and PII
  - Unauthorized insiders have physical access
- Decommissioned hard drives
  - Secure decommissioning is expensive
  - Hardware recycling is typically done in the cheapest and fastest way: no time for secure disk wipe
  - Old PCs may be sent abroad for recycling

FILE ENCRYPTION

Simple file encryption

1. User enters passphrase
2. Passphrase hashed to produce a key
3. File encrypted with the key
   - Symmetric encryption, e.g. AES in CBC mode
   - Integrity check with HMAC or AES-GCM

- Examples: crypt(1), gpg, openssl




```
% gpg --output ciphertext.gpg --symmetric plaintext.doc
Enter passphrase:
```

Limitations of file encryption

- User action needed, and users are lazy
  - Automation (scripting) is difficult. How to store passphrase?
- Passphrase can be brute-forced
- After encryption, what happens to the old plaintext file?
- Software creates temporary files and backup copies
  - Unencrypted file versions and data fragments may be left on disk
- Incompatibility with advanced services:
  - Background processing like search indexing
  - Cloud storage and sharing

Difficulty of deleting files

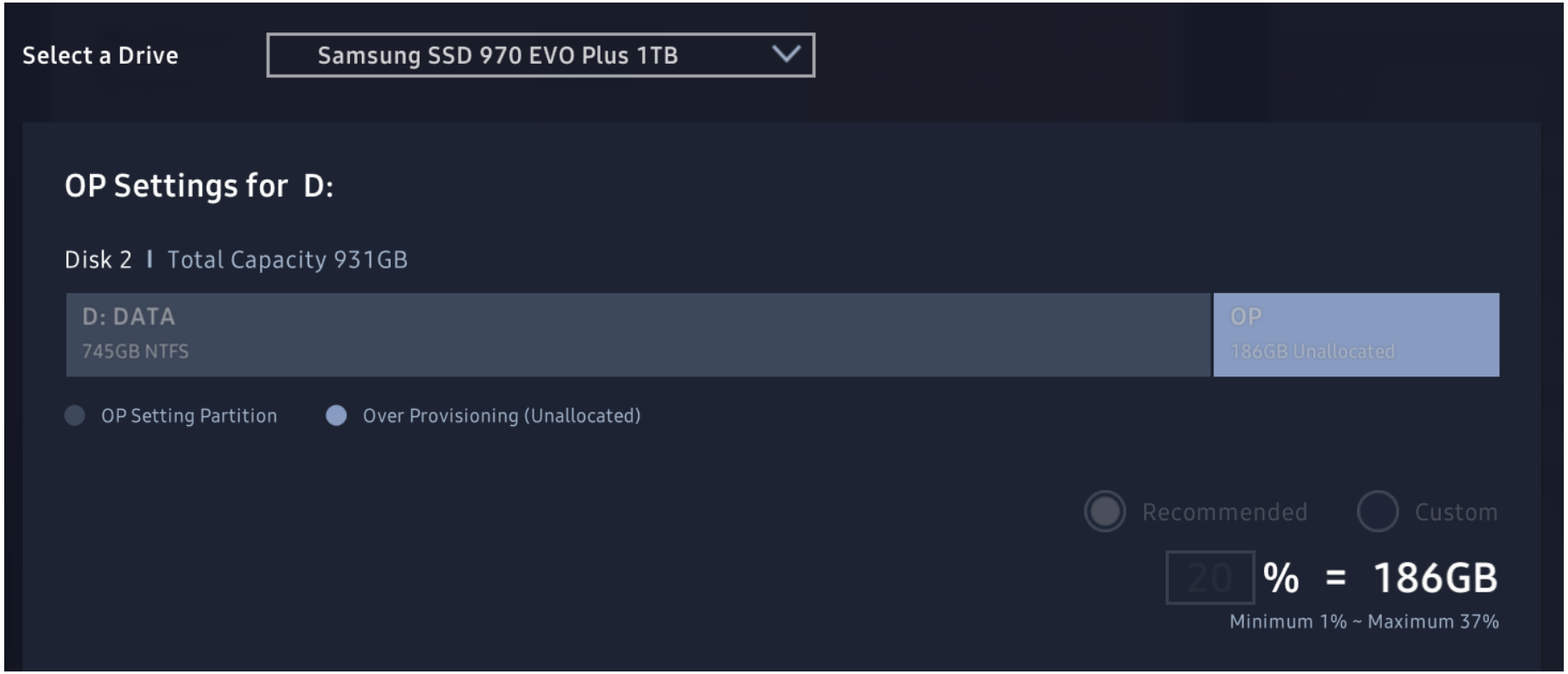
- Deleting a file marks the space free but does not erase data
- Overwriting a file does not always erase the old contents
  - Unpredictable file system behavior: backups, version history, RAID, copy on write, journal, bad blocks etc.
  - Solid-state disks (SSD) avoid rewriting the same physical blocks
- How to reliably delete a file?

Wiping files

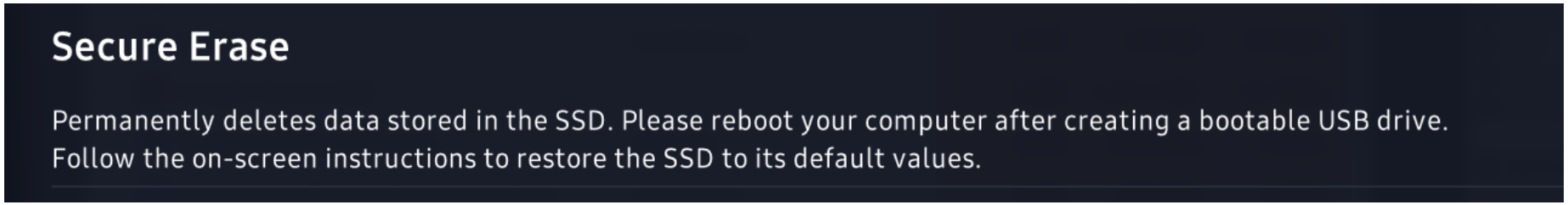
- Overwriting all free disk space
  - Must write pseudorandom data, not just zeros or ones
  - Should read and verify after write
  - Erases most data, but no 100% guarantee (e.g. SSD overprovisioning can hide up to 20% capacity)
  - Magnetic data remanence on magnetic media
- Physical destruction
  - Heating magnetic medium above its Curie temperature
  - Grinding SSD or disks to fine powder

Windows:
`cipher.exe /W D:\`

SSD overprovisioning example



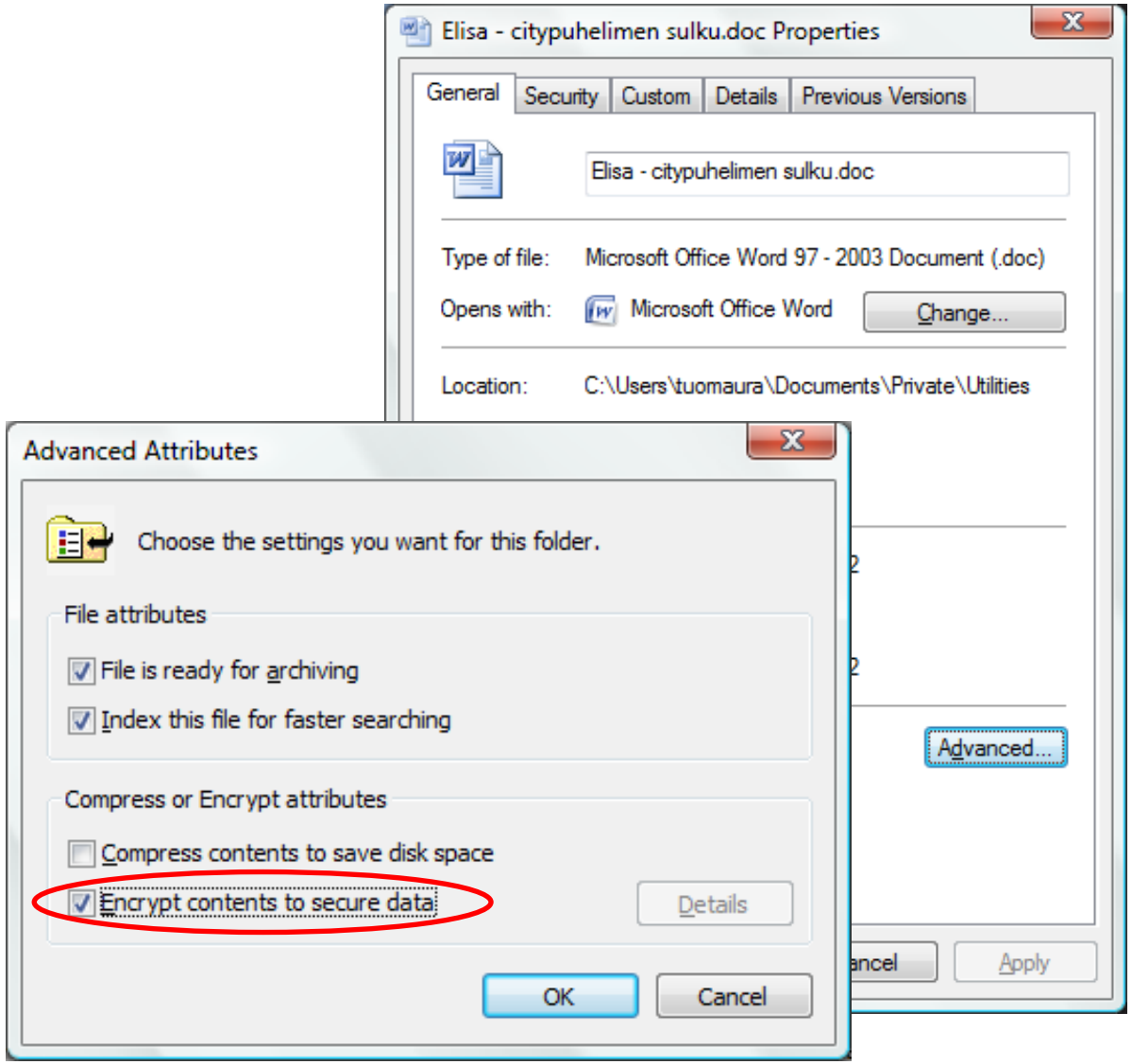
- Filling the disk does not overwrite all blocks
- Must use built-in secure erase function



ENCRYPTING FILE SYSTEM

Windows encrypting file system (EFS)

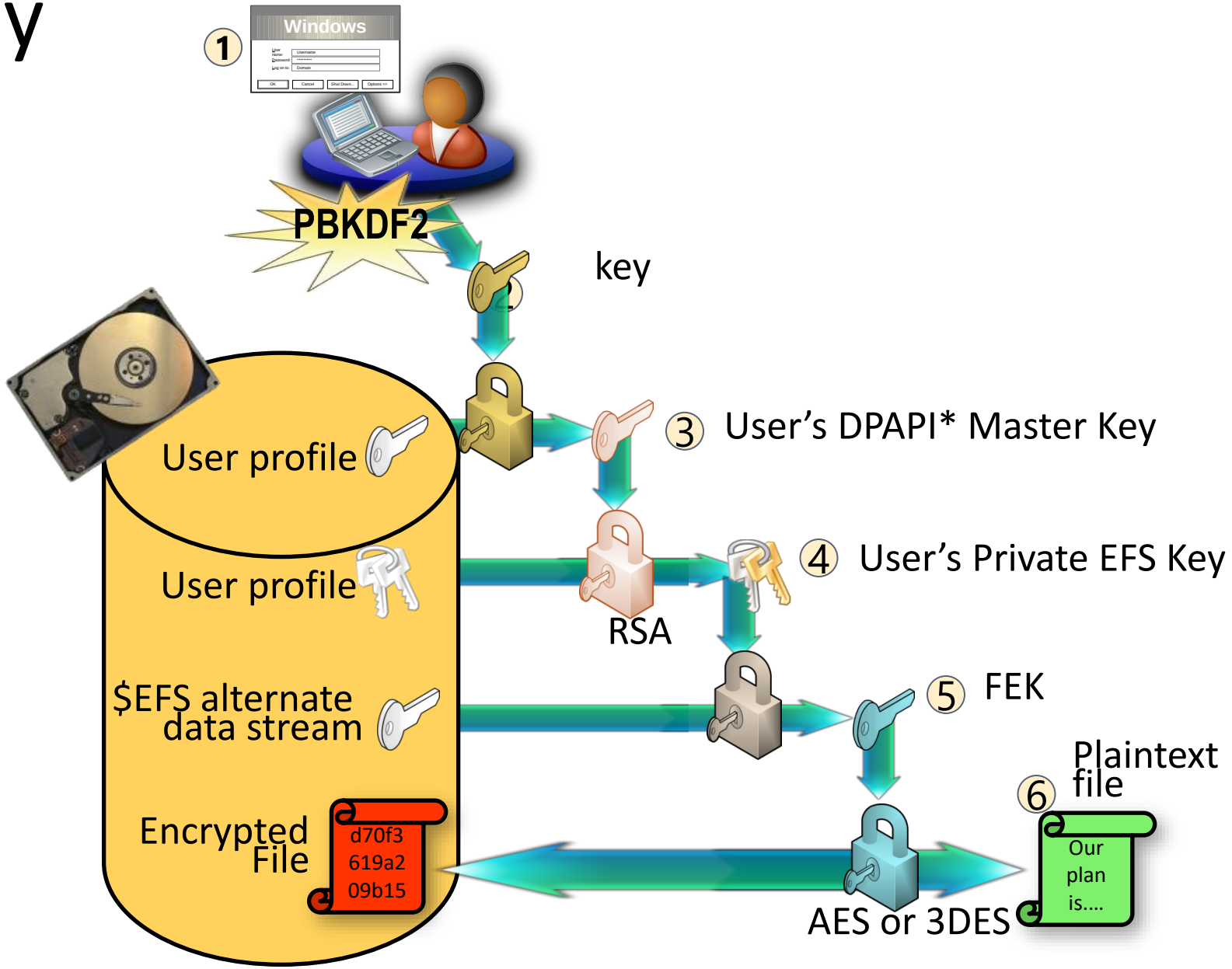
- Encryption is a file attribute in NTFS
- Enable encryption for a folder → new files encrypted from start
- Files can be read when the user is logged in
- Encryption and decryption are transparent to applications



EFS key hierarchy

*) DPAPI = Data Protection API

1. User logs in with password
2. Hashed to produce key
3. Used to decrypt User's Master Key
4. Used to decrypt User's Private EFS Key
5. Used to decrypt File Encryption Key (FEK)
6. Used to encrypt on write and decrypt on read



EFS ANALYSIS

EFS limitations: partial protection

- Encrypts contents of specific files and folders
- Some data is not encrypted:
  - Folder and file names
  - EFS-unaware software may create backup copies, temp files
  - Registry, system files and logs cannot be encrypted
  - Page file can be encrypted but requires policy configuration
  - Search index, if you has chosen to index the encrypted files
- When encrypting plaintext files, the original file is not wiped: always create files in an encrypted folder
- Hibernation file may store decryption keys: disable hibernation
- EFS only on NTFS and FAT; files copied elsewhere are decrypted

EFS limitations: attacks

- Requires strong passwords; password cracking is a danger
- Key logger or hidden camera can capture user password
- Malware can steal EFS data when the user is logged in
- Attacker with physical access to the computer or its hard disk can compromise the OS and install a root kit or key logger. Easiest to configure a malicious data recovery agent (DRA)

Which attacks apply to lost or stolen laptops?

EFS and password cracking

- EFS security depends on the secrecy of the user password
- Password are vulnerable to brute-force cracking
  - Password hashes are stored in a database (SAM) on the disk
  - Windows NT and LM hashes are fast to compute and use no salt
  - Attacker can extract the password hashes
- Why is it not sufficient to reset user or admin password?
- EFS supports smart cards as a login method
- Q: Do EFS and Windows Hello work together?

EFS limitations: usability

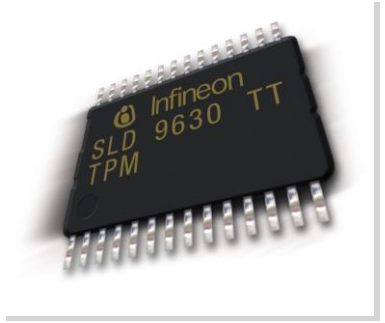
- User must understand what data is encrypted and when
- Non-domain users must backup EFS keys or risk data loss
- User password needed for decryption
  - OS cannot index files when user logs out
  - Backups contain encrypted files (do keep your old EFS keys backup!)
  - Password reset by admin or special tools may lead to loss of data
- Transparent decryption can cause surprises
  - Files copied to online storage and other media are decrypted
  - Files copied to USB stick (FAT) are converted to PFILE, cannot be opened in another computer

FULL DISK ENCRYPTION: BITLOCKER

Full disk encryption

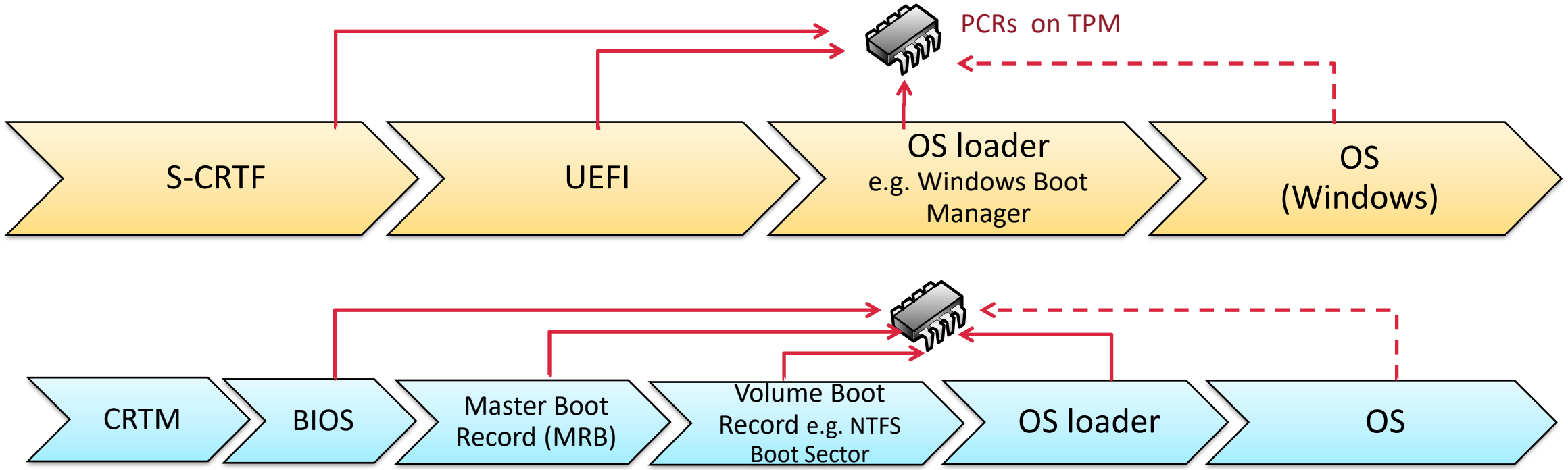
- Entire disk is encrypted: protects all information
- Usually, user must provide password, key, or physical token at boot time or to mount the disk; thereafter transparent encryption
- Software vs hardware based solutions:
  - Security of entirely software-based products depends on the strength of the password or key
  - Tamper-proof hardware prevents brute-force cracking and allows the use of a short PIN
- The main innovation in BitLocker was to use tamper-proof hardware for unsupervised boot

Trusted platform module



- Trusted platform module (TPM) is a smart-card-like tamperproof module on the motherboard or integrated in the CPU
- Stores cryptographic keys
  - Protects keys against both software and hardware attacks
  - Key cannot be stolen, but can it be misused?
- Accumulates platform measurements in platform configuration registers (PCR) for checking software integrity during boot

Measured boot with TPM



- Each component in the boot process measures next component, i.e. computes its hash, before launching it
- OS is usually signed and not measured

Software authentication with TPM

- Measuring platform configuration:
  - Each software module computes hash of the next module and extends the hash into a platform configuration register (PCR) in TPM
  - Then transfers control to the next module
- PCRs contain a cumulative fingerprint (hash) of all software loaded during the boot process
- Originally designed as a DRM feature: decrypt music only for untampered OS and media player

Sealing data with TPM

- TPM operations:
  - Sealing (TMP2_Create): encrypt data in any platform configuration
  - Unsealing (TPM2_Unseal): decrypt the data, but only if the platform configuration has not changed from the time of sealing
- Before unsealing, the TPM compares current PCR values to ones saved when sealing

Windows BitLocker

- Principle: Encrypt drive with symmetric encryption. Seal the symmetric encryption key with TPM and store the sealed key on the disk. Unseal the key when booting
  - TPM detects tampering of system software and prevents booting by refusing to unseal the key

- Design goals:
  - Laptop thief has to attack the TPM hardware
  - Computer and hard drive decommissioning by resetting the TPM
  - Windows user login and access control cannot be bypassed by moving disk to another computer or booting from USB
  - Dual boot allowed, but Linux cannot access the data

Which PCR values does BitLocker use for sealing?

Extra material

*PCR 00: CRTM, BIOS and Platform Extensions
(PCR 01: Platform and Motherboard Configuration and Data)
*PCR 02: Option ROM Code
(PCR 03: Option ROM Configuration and Data)
*PCR 04: Master Boot Record (MBR) Code
(PCR 05: Master Boot Record (MBR) Partition Table)
(PCR 06: State Transitions and Wake Events)
(PCR 07: Computer-Manufacturer Specific)
*PCR 08: NTFS Boot Sector
*PCR 09: NTFS Boot Block
*PCR 10: Boot Manager
*PCR 11: BitLocker Critical Components

- If any of the *-marked values has changed, the decryption key will not be unlocked and a recovery password is needed
- BitLocker keys will be unlocked before OS upgrade

(Different PCR list in UEFI)

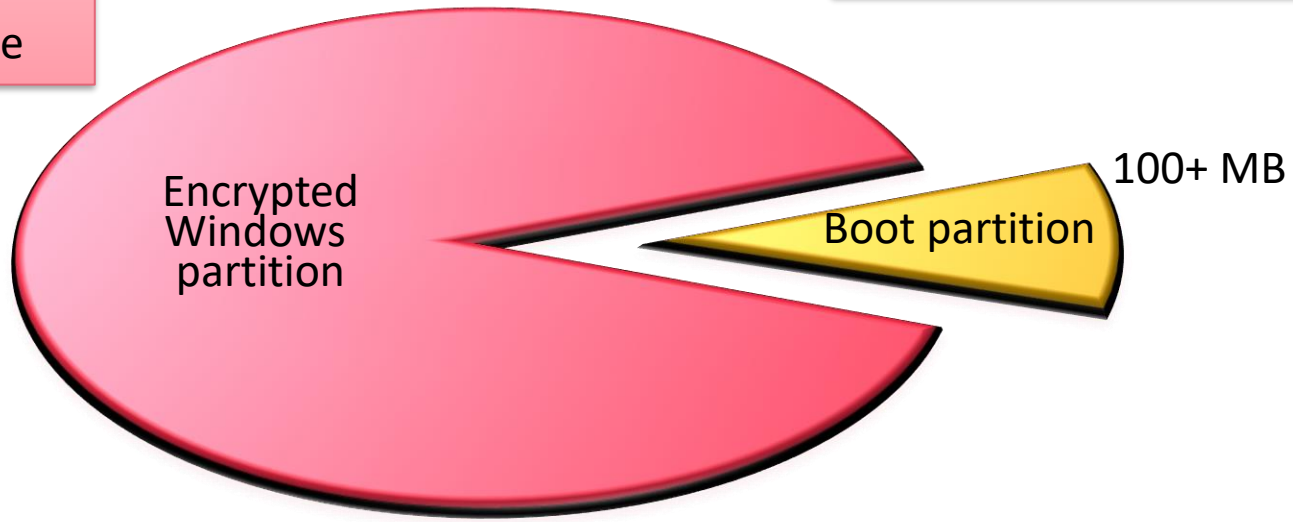
BitLocker partitions

Windows partition contains:

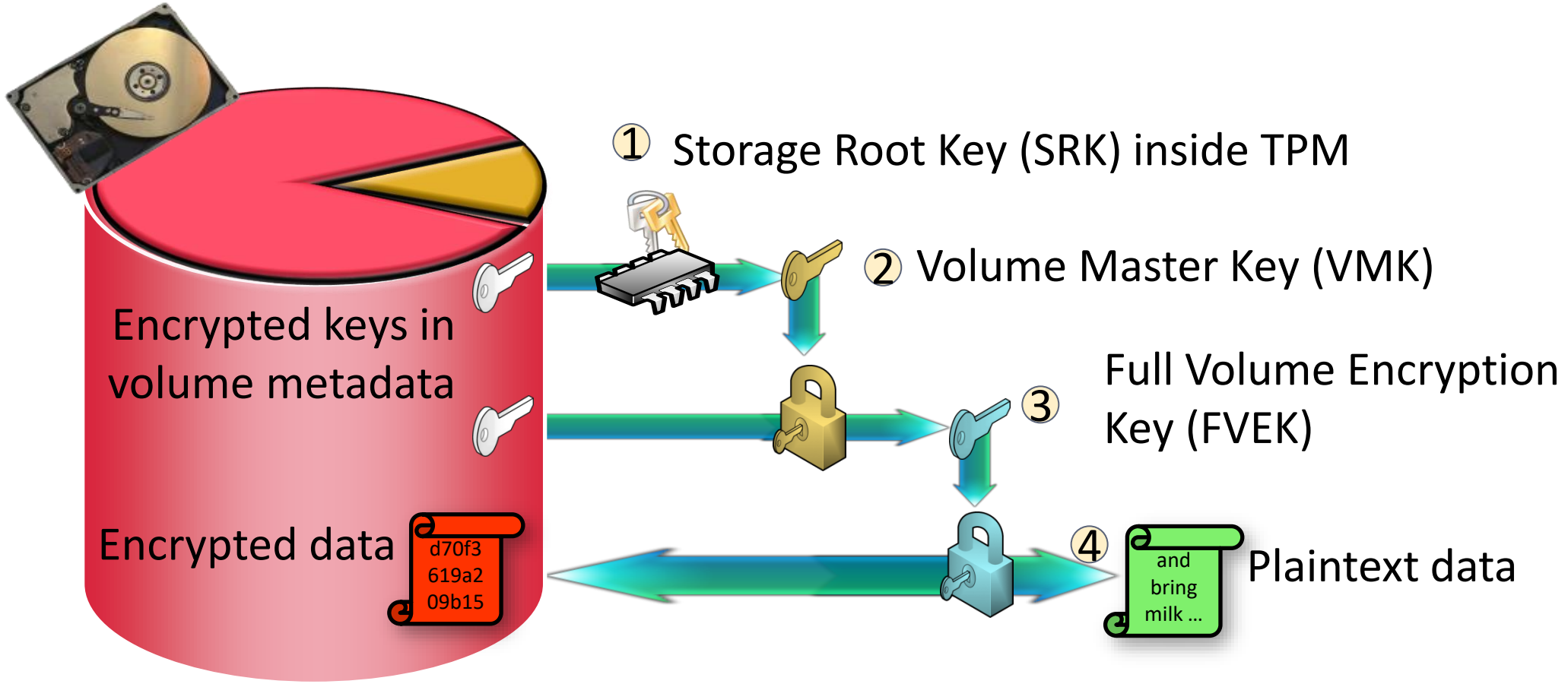
- Volume metadata with MAC
- Encrypted OS
- Encrypted page file
- Encrypted temp files
- Encrypted data
- Encrypted hibernation file

Boot (or Recovery) partition contains:

- Master boot record (MBR)
- OS loader
- Boot utilities



BitLocker key hierarchy



Separate VMK/FVEK adds flexibility — how?

Individual disk blocks are encrypted on write and decrypted on read.

Algorithms and key sizes

Extra material

- Storage root key (SRK) is a 2048-bit RSA key
- Volume master key (VMK) is a 256-bit symmetric key
- Full volume encryption key (FVEK) is a 128 or 256-bit symmetric key
- Disk sectors encrypted with AES in CBC or XTS mode
  - Initialization vector (IV) derived from sector number (no space for storing a random IV in each disk block)
- No integrity check on data: a MAC would increase the data size

BitLocker modes

- TPM only = transparent mode
  - Unsupervised boot: VMK unsealed if PCR values correct
  - Attacker can boot stolen laptop but not log in

Very convenient but dangerous - see the following slides!

- TPM and PIN: TPM requires a PIN at boot time
  - TMP will lock up after a small number of incorrect PINs

Best practice for security

- Network unlock
  - Unsupervised reboot for servers in the same network with AD
- Others: USB stick with key file, USB stick + PIN, password only

[Photo: Microsoft]

BITLOCKER ANALYSIS

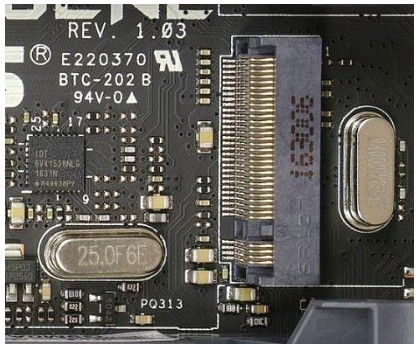
Secure path issues

- Hacked hardware can intercept PIN input to TPM
  - Hardware key logger, modified keyboard, modified BIOS
  - Attacker must have access to the computer twice: first to install the hardware, then to recover the captured PIN/key

    → Lost and stolen computers are safe
- Malware could fake the reboot process and ask for the PIN
  - Could also steal BitLocker keys from a USB stick
- Hidden camera can also capture the PIN

DMA attack

- Direct memory access (DMA) enables auxiliary devices to access main memory directly
  - FireWire, PCIe (also Thunderbolt, M.2, ExpressCard) busses
  - Hot-pluggable helps attacker



- DMA memory dump attack: Attacker connects malicious device to a DMA port, reads the memory, and recovers the key (FVEK)
- Windows 10 disables hot-pluggable PCIe ports when screen locked

[Photos: Wikimedia, notebookreview.com]

Hardware security issues

- Communication between the TPM and CPU is vulnerable
  - If TPM is on discrete chip, attacker can tap the bus to it. Breaks security of the TPM-only mode by sniffing the VMK
  - Some TPMs are on a separate PCB, making the attack easy [link]

- The TPM itself must be secure
  - Infineon TPMs had a key-generation bug in 2017 that required firmware update and generation of a new storage root key [link]
  - Sleep mode implementation flaws allow overwriting of PRC values, which allows attacker to unseal VMK in TPM-only mode [link, link]

Cold boot attack

- Data remanence in main memory:
  - After power loss, DRAM memory contents decay in seconds

(a) 0 sec / 100% **(b)** 2 sec / 99.2% **(c)** 3 sec / 93.4% **(d)** 4 sec / 93.1% **(e)** 5 sec / 61.4% **(f)** 6 sec / 51.9%

Measurements with GDDR2

[Michael Gruhn, Tilo Müller: On the Practicability of Cold Boot Attacks, ARES 2013]

- Cold boot attack (reset attack)
  - Reboot into a minimal hacker OS from USB stick
  - Reboot cuts the power only for a fraction of a second
  - Secret keys can be recovered from memory (possibly with some bit errors)

Cold boot attack: consequences

- Attacker with physical access to a running computer
  - can break software-based full-disk encryption
  - can bypass most OS access controls
- Sleeping laptop = running laptop
  - Lost and stolen laptops are vulnerable if in sleep mode
- BitLocker in TPM-only mode is vulnerable even if powered down or hibernating
  - Attacker can boot it up and then do the cold-boot attack

Mitigation of cold-boot attacks (1)

Best protection: never given attacker access to a running computer

- Disable sleep on the computer, shut down or hibernate instead
- Do not leave computer running unsupervised
- Set BitLocker to a supervised boot mode (TPM+PIN, TPM+USB)

Weaker protection:

- Password-protect BIOS or UEFI and disable USB boot
- BitLocker network unlock for non-mobile domain computers
- UEFI secure boot: only allow certified OS loaders

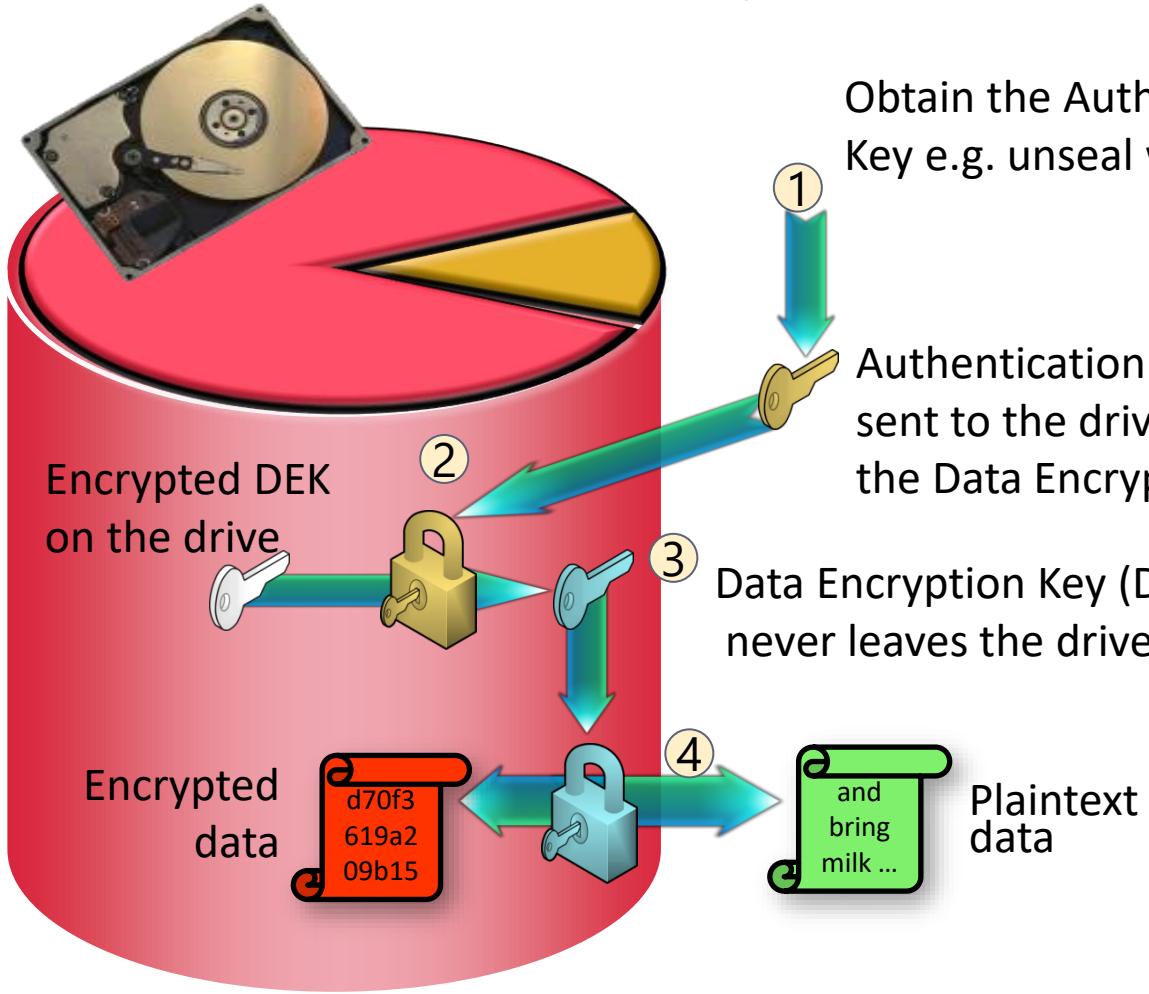
Mitigation of cold-boot attacks (2)

OS and hardware solutions:

- Overwrite memory before reboot (easy to circumvent)
- Memory Overwrite Request Control (MOR): UEFI cleans the memory after reboot
- Encrypted main memory in Xbox
- Self-encrypting drive: encryption on the disk drive itself; keys are only briefly in main memory

Self-encrypting drive

Encryption built into disk hardware/firmware



Performance: offload the data encryption and decryption (AES) to hardware on the drive

Security: mitigates cold-boot and DMA vulnerabilities because keys are only briefly on the main memory during boot

Attacker could capture keys on the PCIe/SATA/ SCSI bus during transfer

Microsoft Encrypted Hard Drive = eDrive = TCG Opal

BitLocker and unsupervised OS update

- During software update that requires reboot, BitLocker is suspended: decryption keys are stored on disk
  - Any update script can suspend BitLocker:

```
> Suspend-BitLocker -MountPoint "C:" -RebootCount 1
```

- Consequence: stolen computer is vulnerable if sleeping or allows unsupervised boot (TMP only mode)
  - Attacker can wait for a software update that reboots the computer, interrupt the boot process, and recover VMK from the disk
  - Attacker may even be able to go to the recovery console
- Solution would be to only allow supervised reboot

So, is BitLocker at all secure?

Current situation seems still this:

- If the computer powered down (or hibernating) and BitLocker requires supervised boot (TPM+PIN or TPM+USB key), then it is safe
  - Possible to boot up unsupervised, protect data drive D: with TPM+PIN
- Running computer, or one that allows unsupervised boot (TPM only), is vulnerable if the attacker has physical access to it
- Firmware and software updates sometimes needed to fix implementation flaws

DATA RECOVERY

Need for data recovery: EFS

If the decryption key is lost, encrypted files will be lost

- EFS files cannot be read after password reset
  - Reset by admin and hacking tools have the same effect
  - Password change by the user, of course, is ok
- EFS files in backup media unreadable if the user's EFS private keys are lost
  - Can happen when rebuilding or cleaning user profile or replacing the computer
- Removable FAT drives may contain encrypted files that depend on the computer that wrote them

→Excellent idea to backup new and old EFS certificates and private keys!

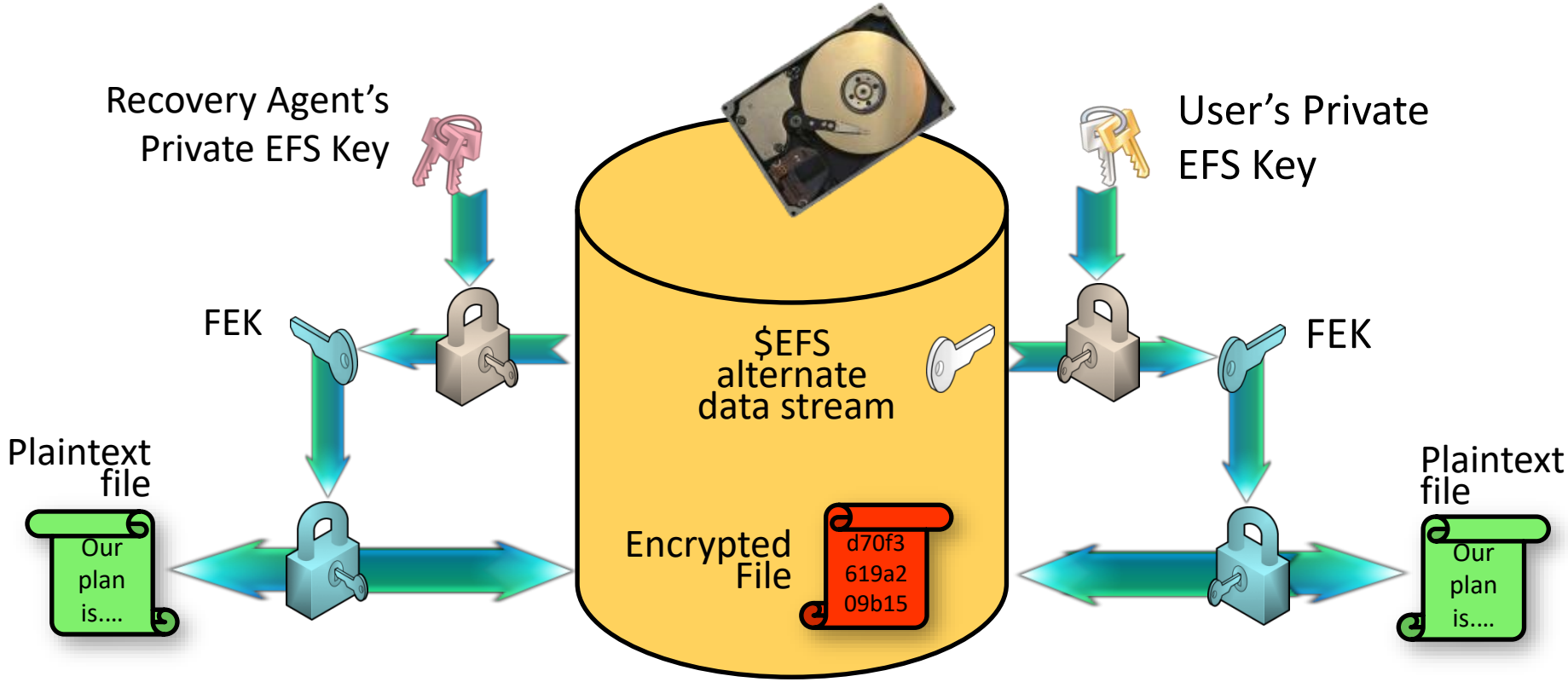
Data recovery in EFS: domain

- Windows domain has a data recovery agent (DRA)
  - FEK is encrypted also with the DRA public key
  - Domain Administarator is the default DRA
  - Other DRAs can be defined in a Group Policy in the domain
- If uncertain, check on any EFS-encrypted file:

```
> cipher /c .\file.txt
```

Data recovery in EFS

- File encryption key (FEK) is encrypted with one or more recovery agents' public keys
- The same mechanism for sharing encrypted files between users



Data recovery in EFS: non-domain

- Standalone machine has no default DRA
  - Possible to set up your own DRA (cipher.exe)

Not for the average user

- User may backup their EFS certificate and private key to a USB or cloud drive
  - Open certmgr.msc, under Personal, export *all* Encrypting File System certificates with private key

Not for the average user

- Latest Password Reset Disk recovers EFS private key
  - Did you make one?

Ok for all, but requires user action

Need for data recovery: BitLocker

If the TPM fails to unseal keys, encrypted data will be lost

- TPM cannot unseal the key when
  - Installing or updating Linux boot loader (dual boot)
  - Motherboard with TPM replaced
  - Processor with embedded TPM replaced
  - TPM has been reset
  - Hard drive moved to another computer
  - TPM boot PIN forgotten
  - TPM boot PIN mistyped many times

→ Everyone will eventually need recovery

Data recovery in BitLocker

- Recovery password:
  - User can print a 48-digit recovery password or store it on a USB stick or remote disk; it is actually a 128-bit key
  - BitLocker encrypts the VMK with the recovery password and stores it with the volume metadata (in the same way as the TMP-sealed VMK)
- Domain computers save recovery key to Active Directory, others to Microsoft account or employer/university Azure AD

Where are your BitLocker recovery keys for each computer and drive? Always check data recovery methods before the accident happens

SUMMARY

Summary

- It is a very good idea to encrypt stored data, and convenient technical solutions exist

However:

- Backups the keys to paper or to an online service
- Check that you can find and download the recovery keys
- Otherwise, expect your data to be lost accidentally
  – there will be no way to decrypt it!

List of key concepts

- File encryption, wiping data, data remanence
- *EFS*, transparent encryption , key hierarchy, hibernation file
- Full-drive encryption, *BitLocker*, TPM, sealing and unsealing, trusted boot, PCR
- Secure path, DMA attack , unsupervised boot, cold-boot attack, network unlock, self-encrypting drive
- Data recovery agent, recovery password, key backup

Exercises

- What secure methods are there for erasing magnetic hard drives and tapes, USB stick or solid-state drives (SSD), and paper documents?
- How to delete a specific file from a computer securely without erasing the whole disk?
- What security properties does GPG file encryption or EFS provide that full-disk encryption does not?
- How vulnerable is EFS to password guessing?
- Why do EFS and BitLocker have a key hierarchy of so many levels? Are all the keys necessary?
- Compare the security of fully software-based full-disk encryption and the TPM approach against brute-force password guessing.
- Explain the differences between running, sleeping, hibernating, and powered-down laptop computer under the cold boot attack.
- In EFS, local Admin cannot read the users' encrypted files without the user passwords; can the Admin get around this?
- Windows 2000 had the local Administrator account as the default EFS DRA for non-domain machines. Why was this not a good idea?
- Transparent operation (happens without the user or application even knowing) improves usability of data encryption, but are there risks associated with the transparency?
- How would you design the encryption of files in cloud storage? Can the encryption support file sharing between users? What would be the policy and mechanism for revoking access to shared data?

Related reading

- Online:
  - Halderman et al., *Lest We Remember: Cold Boot Attacks on Encryption Keys*. http://citp.princeton.edu/memory/
- Stallings and Brown: Computer security, principles and practice, 3rd ed., chapter 2.6