Topic 2 Evidence in computers and networks Part 2

Learning Outcome

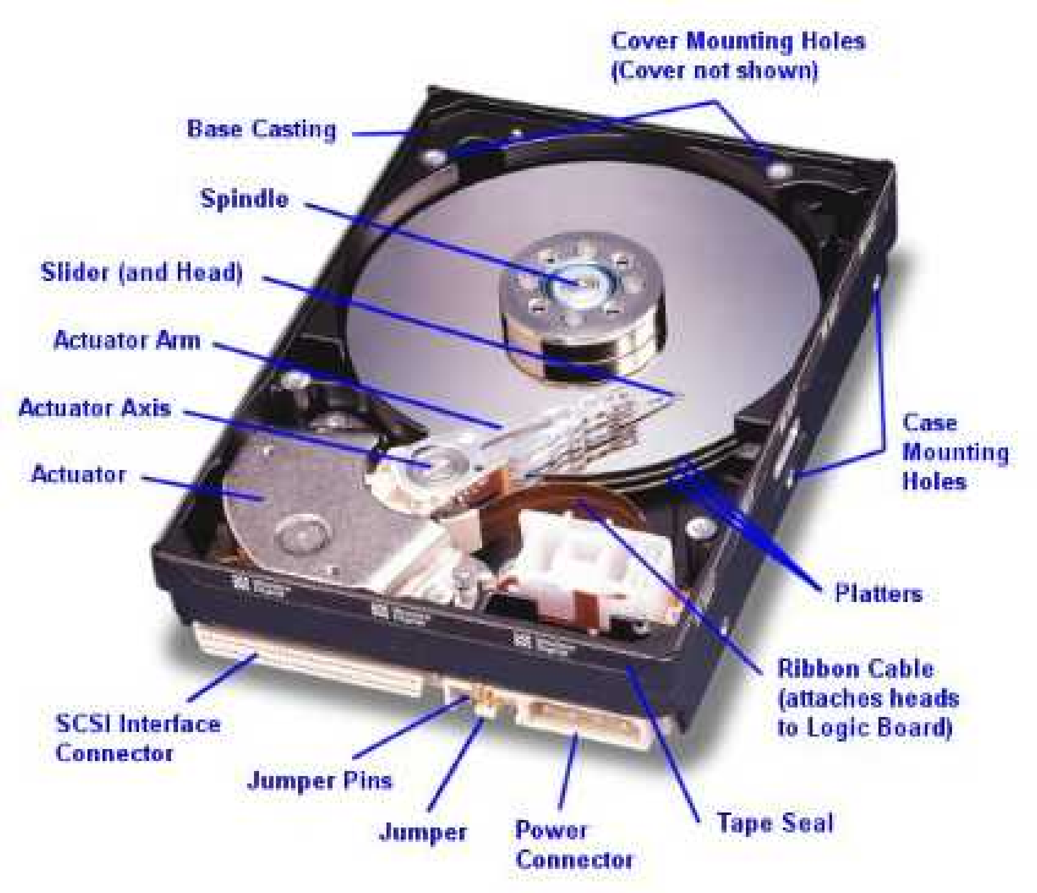
- After successfully completing this lecture, you will be able to
  - Describe MBR Partition Table information
  - Describe Windows file systems FAT and NTFS
  - Describe slack and un-allocated space in hard disk drive

Road Map

- FAT File Systems
- NTFS File Systems
- Slack and Un-allocated space

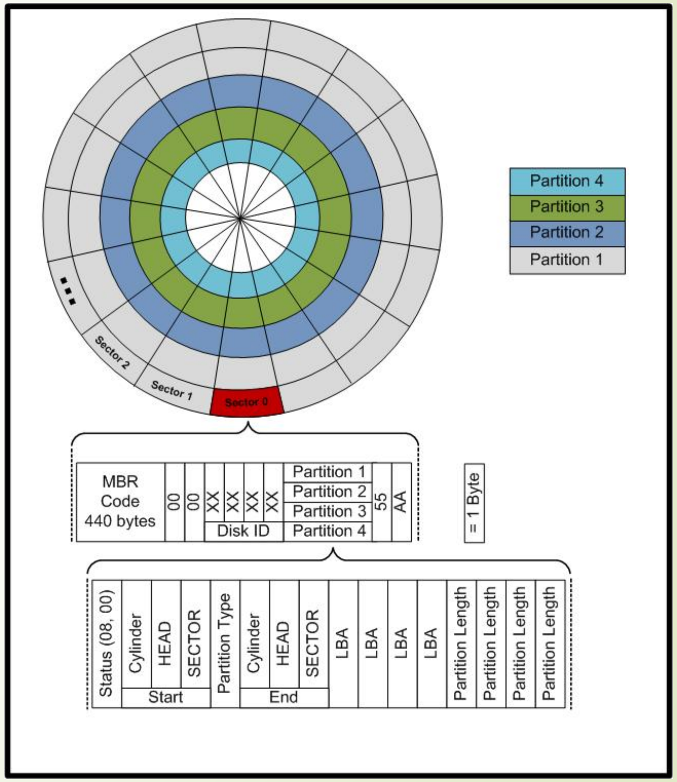
MBR Partition Table

Where are the file systems stored in a hard disk drive?

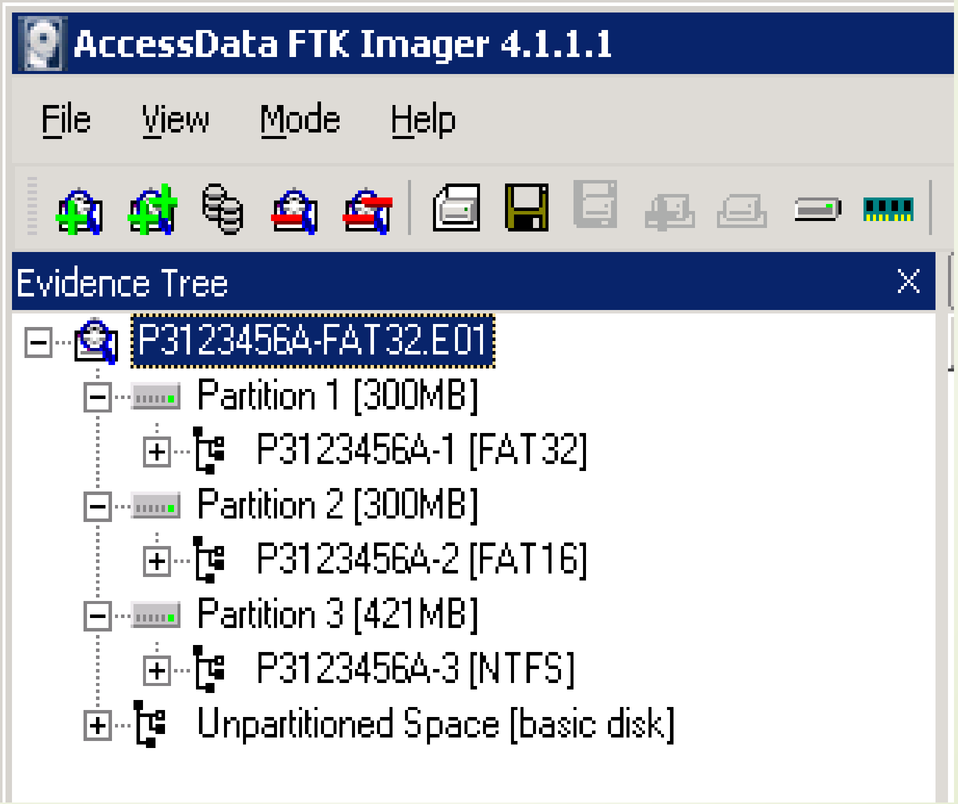


Disk Drive Sectors and Partitions

- Bits are restored as changes of magnetic fields on a disk drive surface
- 8 bits grouped into a byte
- 512 bytes grouped into a sector
- A number of sectors grouped into a partition
- A partition used to store a file system, such as FAT32, NTFS or Ext3



A File System is stored in one (or more) partitions in a drive


AccessData FTK Imager 4.1.1.1

P3123456A-FAT32.E01
- Partition 1 [300MB]
  - P3123456A-1 [FAT32]
- Partition 2 [300MB]
  - P3123456A-2 [FAT16]
- Partition 3 [421MB]
  - P3123456A-3 [NTFS]
- Unpartitioned Space [basic disk]

How can an operating system, such as Windows knows …

- Starting sector of a partition in a drive?
- Number of sectors in a partition?
- The type of file system in a partition?
- Which is the bootable partition with operating system programs inside?

MBR in sector 0 of a physical drive has details of partitions

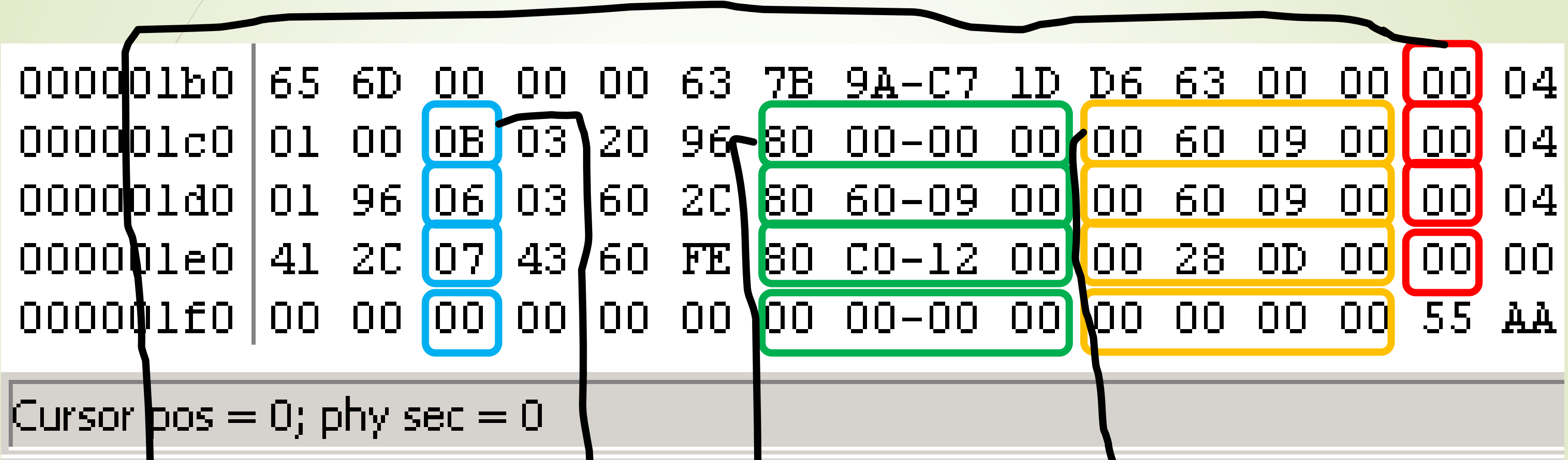
Offset	Description	Size in bytes
0x000	Bootstrap Code Area	446
0x1BE	Partition entry #1	16
0x1CE	Partition entry #2	16
0x1DE	Partition entry #3	16
0x1EE	Partition entry #4	16
0x1FE	0x55	1
0x1A	0xAA	1

A Sample Partition Entry

Offset	Description	Size in bytes
0x0	0x80 Active or 0x00 Inactive	1
0x1	CHS address of the 1st sector in partition	3
0x4	Partition Type e.g. 0x04 means it is a FAT16 partition	1
0x5	CHS address of last sector in the partition	3
0x8	LBA of the 1st sector in the partition	4
0xC	Number of sectors in the partition	4

Here we know the value of n is the LBA of the 1st sector

Partition table starts from byte offset 1be in sector 0



```
000001b0 | 65 6D 00 00 00 63 7B 9A-C7 1D D6 63 00 00 00 04
000001c0 | 01 00 0B 03 20 96 80 00-00 00 00 60 09 00 00 04
000001d0 | 01 96 06 03 60 2C 80 60-09 00 00 60 09 00 00 04
000001e0 | 41 2C 07 43 60 FE 80 C0-12 00 00 28 0D 00 00 00
000001f0 | 00 00 00 00 00 00 00 00-00 00 00 00 00 00 55 AA
Cursor pos = 0; phy sec = 0
```

Entry	Active or inactive	Partition Type (e.g. FAT32)	Starting sector of the partition (in decimal)	Size of the partition in sectors (in decimal)
1	inactive	0B (FAT32)	80 00 00 00(128)	00 60 09 00 (614400)
2	inactive	06 (FAT16B)	80 60 09 00 (614528)	00 60 09 00 (614400)
3	inactive	07 (NTFS)	80 C0 12 00 (1228928)	00 28 0D 00 (862208)
4	00	00	00 00 00 00	00 00 00 00

How to read the Partition table?

Please read the partition table details at the following web pages

- https://en.wikipedia.org/wiki/Master_boot_record
- https://en.wikipedia.org/wiki/Partition_type
- Partition starting sector number and size of a partitions in sectors are read in little Endian byte order
- Partition type 00 means empty partition entry

Windows File Systems

Windows File Systems

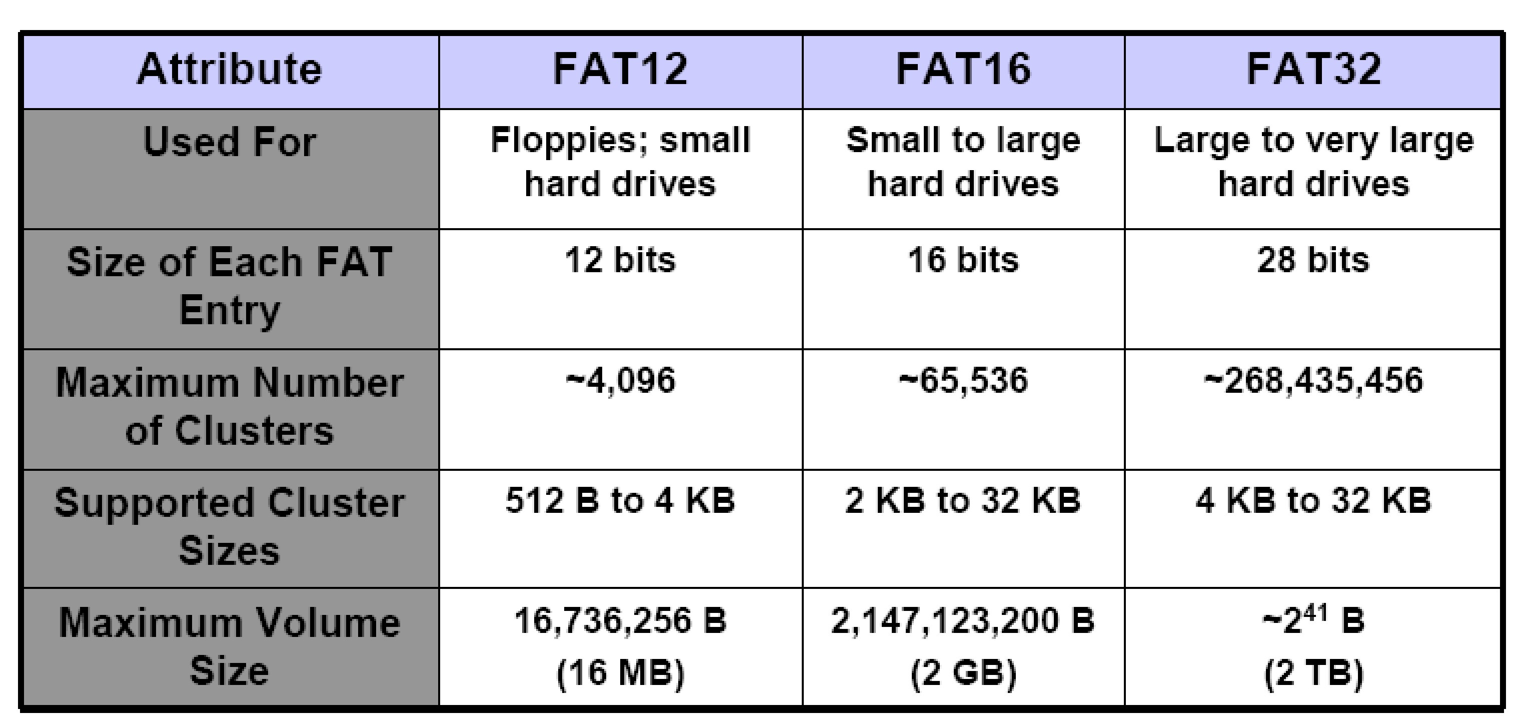
- File system format
  - Organizes and stores data of different files in different designated clusters of sectors
  - Provide index to the logical location (cluster and sectors number) to individual file on the medium
  - Provide date/time information on file creation, modification and access
- Windows File Systems
  - FAT (File Allocation Table)
  - NTFS (New Technology File System)
  - exFAT (Extended FAT)
  - ReFS (Resilient File System)

8.3 Filename Limit

- For backward compatibility with MSDOS, an 8.3 filename is automatically generated for every long filenames
  - TextFile1.txt => TEXTFI~1.TXT
- To show
  - dir /x – shows the short names (if any), and the long names
  - dir /-n – shows only the short names

FAT
(File Allocation Table)

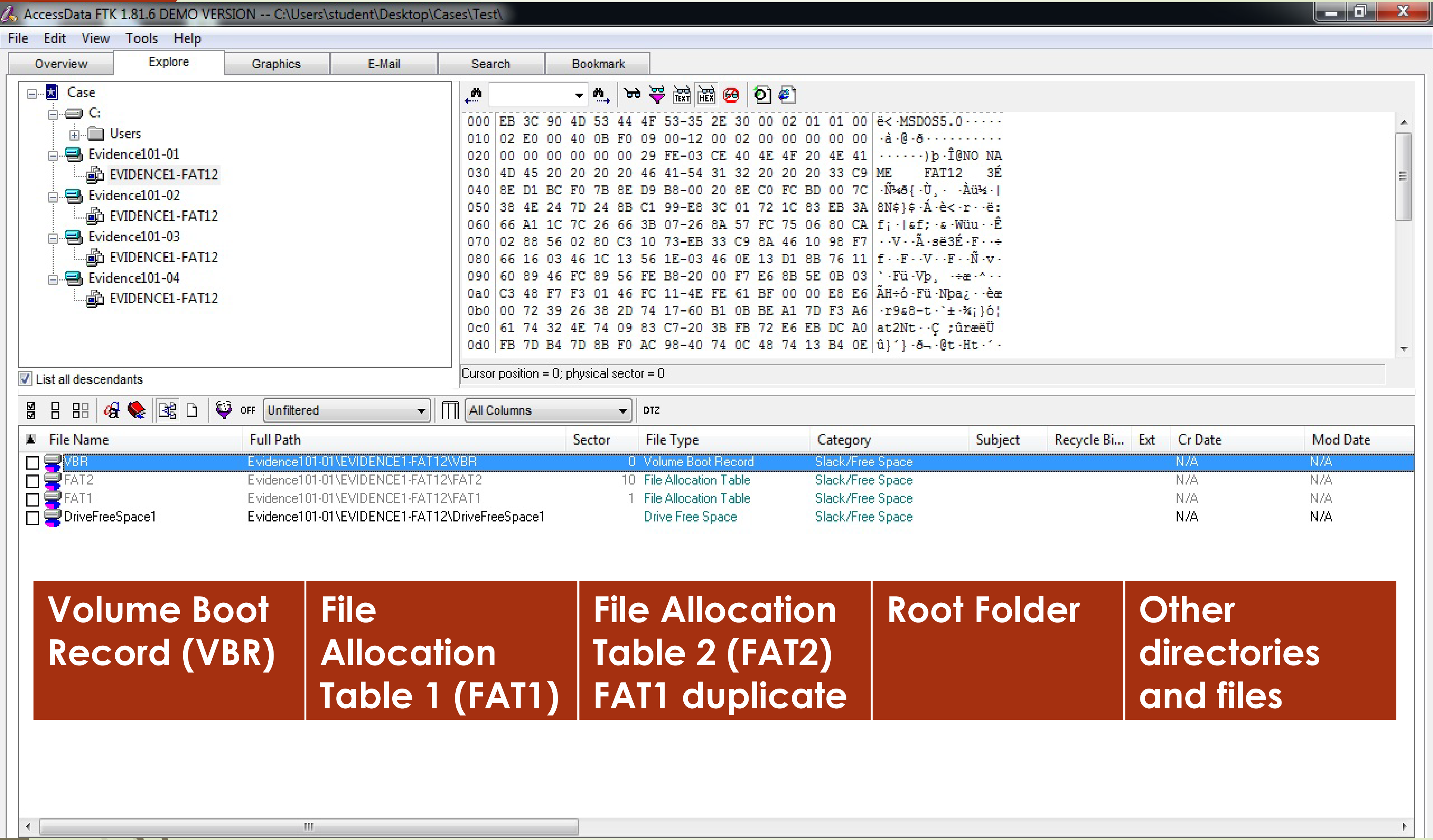
FAT12, FAT16 and FAT32 Comparison



Attribute	FAT12	FAT16	FAT32
Used For	Floppies; small hard drives	Small to large hard drives	Large to very large hard drives
Size of Each FAT Entry	12 bits	16 bits	28 bits
Maximum Number of Clusters	~4,096	~65,536	~268,435,456
Supported Cluster Sizes	512 B to 4 KB	2 KB to 32 KB	4 KB to 32 KB
Maximum Volume Size	16,736,256 B (16 MB)	2,147,123,200 B (2 GB)	~2^{41} B (2 TB)

Source: http://www.c-jump.com/CIS24/Slides/FAT/lecture.html#F01_0200_fats_compared

Sample FAT12 File System



```
000 EB 3C 90 4D 53 44 4F 53-35 2E 30 00 02 01 01 00 ë<·MSDOS5.0·····
010 02 E0 00 40 0B F0 09 00-12 00 02 00 00 00 00 00 ·à·@·ð··········
020 00 00 00 00 00 00 29 FE-03 CE 40 4E 4F 20 4E 41 ······)þ·Î@NO NA
030 4D 45 20 20 20 20 46 41-54 31 32 20 20 20 33 C9 ME    FAT12   3É
040 8E D1 BC F0 7B 8E D9 B8-00 20 8E C0 FC BD 00 7C ·Ñ¼ð{·Ù¸· ·Àü½·|
050 38 4E 24 7D 24 8B C1 99-E8 3C 01 72 1C 83 EB 3A 8N$}$·Á·è<·r··ë:
060 66 A1 1C 7C 26 66 3B 07-26 8A 57 FC 75 06 80 CA f¡·|&f;·&·Wüu··Ê
070 02 88 56 02 80 C3 10 73-EB 33 C9 8A 46 10 98 F7 ··V··Ã·së3É·F··÷
080 66 16 03 46 1C 13 56 1E-03 46 0E 13 D1 8B 76 11 f··F··V··F··Ñ·v·
090 60 89 46 FC 89 56 FE B8-20 00 F7 E6 8B 5E 0B 03 `·Fü·Vþ¸ ·÷æ·^··
0a0 C3 48 F7 F3 01 46 FC 11-4E FE 61 BF 00 00 E8 E6 ÃH÷ó·Fü·Nþa¿··èæ
0b0 00 72 39 26 38 2D 74 17-60 B1 0B BE A1 7D F3 A6 ·r9&8-t·`±·¾¡}ó¦
0c0 61 74 32 4E 74 09 83 C7-20 3B FB 72 E6 EB DC A0 at2Nt··Ç ;ûræëÜ 
0d0 FB 7D B4 7D 8B F0 AC 98-40 74 0C 48 74 13 B4 0E û}´}·ð¬·@t·Ht·´·
```

Cursor position = 0; physical sector = 0

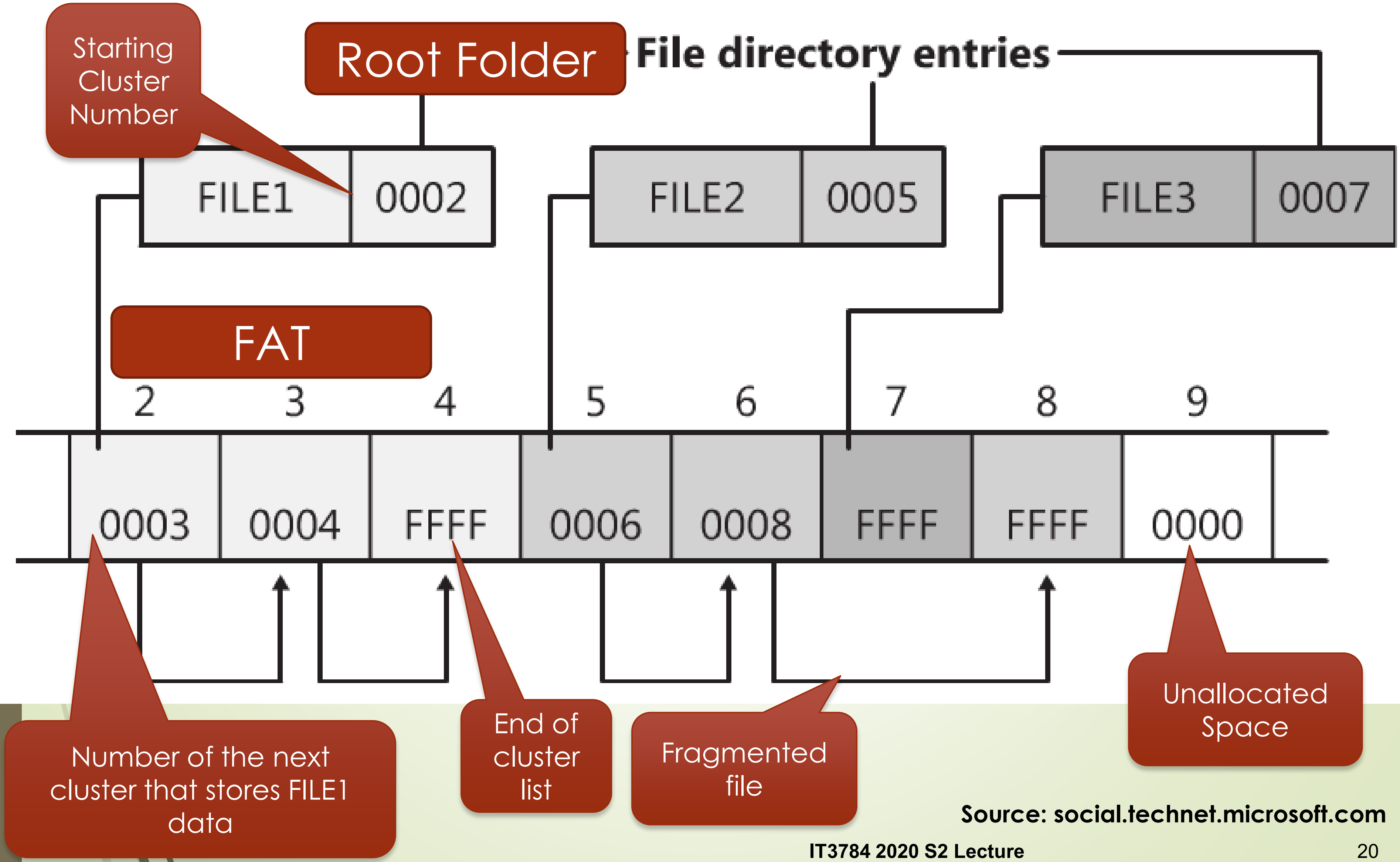
	File Name	Full Path	Sector	File Type	Category	Subject	Recycle Bi...	Ext	Cr Date	Mod Date
□	VBR	Evidence101-01\EVIDENCE1-FAT12\VBR	0	Volume Boot Record	Slack/Free Space				N/A	N/A
□	FAT2	Evidence101-01\EVIDENCE1-FAT12\FAT2	10	File Allocation Table	Slack/Free Space				N/A	N/A
□	FAT1	Evidence101-01\EVIDENCE1-FAT12\FAT1	1	File Allocation Table	Slack/Free Space				N/A	N/A
□	DriveFreeSpace1	Evidence101-01\EVIDENCE1-FAT12\DriveFreeSpace1		Drive Free Space	Slack/Free Space				N/A	N/A

Volume Boot Record (VBR)	File Allocation Table 1 (FAT1)	File Allocation Table 2 (FAT2) FAT1 duplicate	Root Folder	Other directories and files

What are the different areas in a FAT file system?

- **Volume Boot Record (VBR)**
  - Store FAT information that includes
    - number of bytes per sector,
    - number of sectors per cluster,
    - number of sectors per FAT and number of
- **File Allocation Table (FAT)**
  - Stores Addresses of cluster used by individual file
  - Special data patterns represent different status of the cluster
    - Unallocated (0x0000)
    - Bad cluster (0xFFF7)
    - Last cluster in a file (0xFFF8 - 0xFFFF)
- **Root Folder/Directory**
  - Filenames, Directory names
  - Attributes of individual file
    - Date and timestamp, the starting cluster number and status (archived, hidden, system and read-only).

FAT stores linked lists of clusters



Starting Cluster Number

Root Folder

File directory entries

FILE1	0002
FILE2	0005
FILE3	0007

FAT

2	3	4	5	6	7	8	9
0003	0004	FFFF	0006	0008	FFFF	FFFF	0000

Number of the next cluster that stores FILE1 data

End of cluster list

Fragmented file

Unallocated Space

Source: social.technet.microsoft.com

From the Root Folder and FAT, an operating system knows

- From Root Folder
  - FILE1 starting cluster number is 0002
  - FILE2 starting cluster number is 0005
  - FILE3 starting cluster number is 0007
- From FAT
  - FILE1 stored in the clusters 2, 3 and 4
  - FILE2 stored in the clusters 5, 6 and 8
  - FILE3 stored in the cluster 7

Watch the Youtube video “File Allocation Table” from Udacity to learn the how clusters used by a file are linked in FAT

NTFS
(New Technology File System)

NTFS
(New Technology File System)

NTFS provides

- File owner information
- Access Control List in each file/folder header
- System time zone information
- Alternate Data Stream (ADS)
- File storage quota tracking and control
- Encryption File System
- File compression
- Volume shadow copy

What are the different areas in NTFS?

- Partition Boot Sector
  - Similar to VBR in FAT
  - Occupies the first 16 sectors
- Master File Table (MFT)
  - Similar to directory entry in FAT
  - Entry for every file and directory including itself ($MFT)
  - Contains file metadata
  - The starting location of MFT is given in the boot sector
- $bitmap
  - Similar to the file allocation table
  - Represents cluster allocation

Watch the Youtube video "NTFS Forensics and the Master File Table" to learn the NTFS file system how files are stored in NTFS

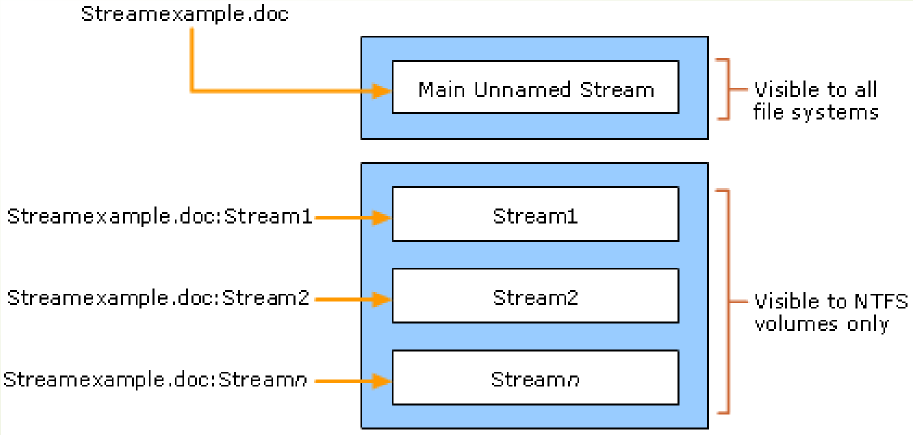
NTFS File System Metadata Files

File Name	Description
$MFT	Entry of MFT itself
$MFTMirr	Backup of the first entries in the MFT
$LogFile	Journal that records the metadata transactions
$Volume	Volume information, such as the label and version
$AttrDef	Attribute information such as identifier values, name and size
.	Root directory of the file system
$Bitmap	Allocation status of each cluster in the file system
$Boot	Boot sector and boot code for the file system
$BadClus	Clusters that have bad sectors
$Secure	Information about the security and access control
$Upcase	Uppercase version of every Unicode character
$Extend	A directory that contains files for optional extension

26

Alternate Data Streams (ADS)

- NTFS ADS were introduced from Windows NT 3.1 onwards
  - For compatibility with the Mac HFS
    - HFS stores icon and other information in an alternative scream.
- ADS are used for other purposes in Windows 2000 and XP
  - Applications can create additional named streams and access these streams by referring to their names, which allows related data to be managed as a single unit.
    - Thumbnails
    - Internet explorer add zone identifier into files downloaded from Internet
- Can be used to hide executable content
  - Perl scripts
  - Windows Scripting Host files
  - Malware


Streamexample.doc
Main Unnamed Stream
Visible to all file systems
Streamexample.doc:Stream1
Stream1
Streamexample.doc:Stream2
Stream2
Streamexample.doc:Streamn
Streamn
Visible to NTFS volumes only

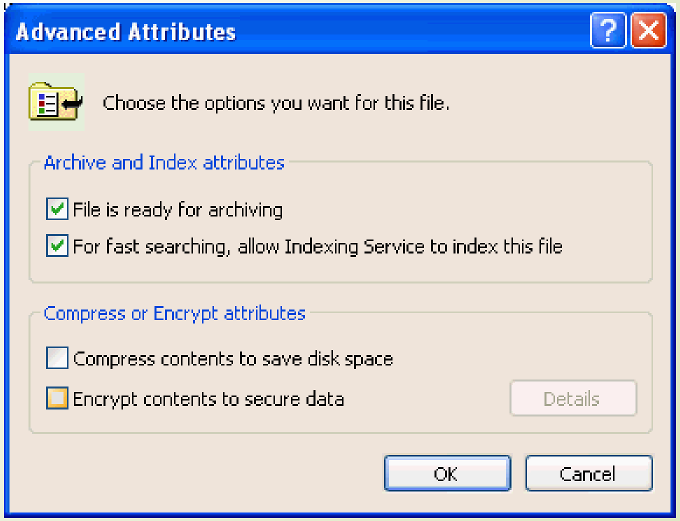
Alternate Data Streams (ADS)

- To create an ADS file
  - echo “this is an ADS” > myfile.txt:ads.txt
  - myfile.txt will also be created but is zero bytes in size

- To identify an ADS file
  - Viewing of NTFS ADS is available for Windows Vista and above
    - Use “dir /r” command
    - myfile.txt:ads.txt:$DATA
  - Many commercial forensic applications will display ADS files in red within the GUI

Encrypting File System

- Allows users to encrypt individual files or entire folders
- Built into Windows 2000 and XP Professional and later
- Encrypted files are only viewable by the user who encrypted them or by designated recovery agents
  - Decryption is automatic without the need to enter password
- Can invoke feature by selecting checkbox in Advanced Attributes property of files


Advanced Attributes

Choose the options you want for this file.

Archive and Index attributes

☑ File is ready for archiving

☑ For fast searching, allow Indexing Service to index this file

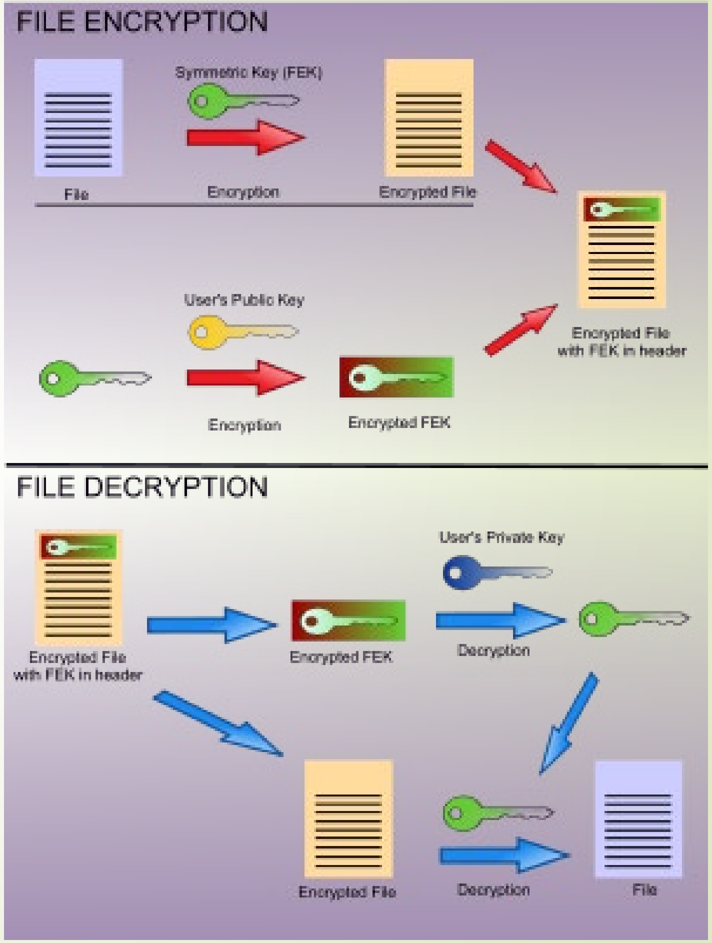
Compress or Encrypt attributes

☐ Compress contents to save disk space

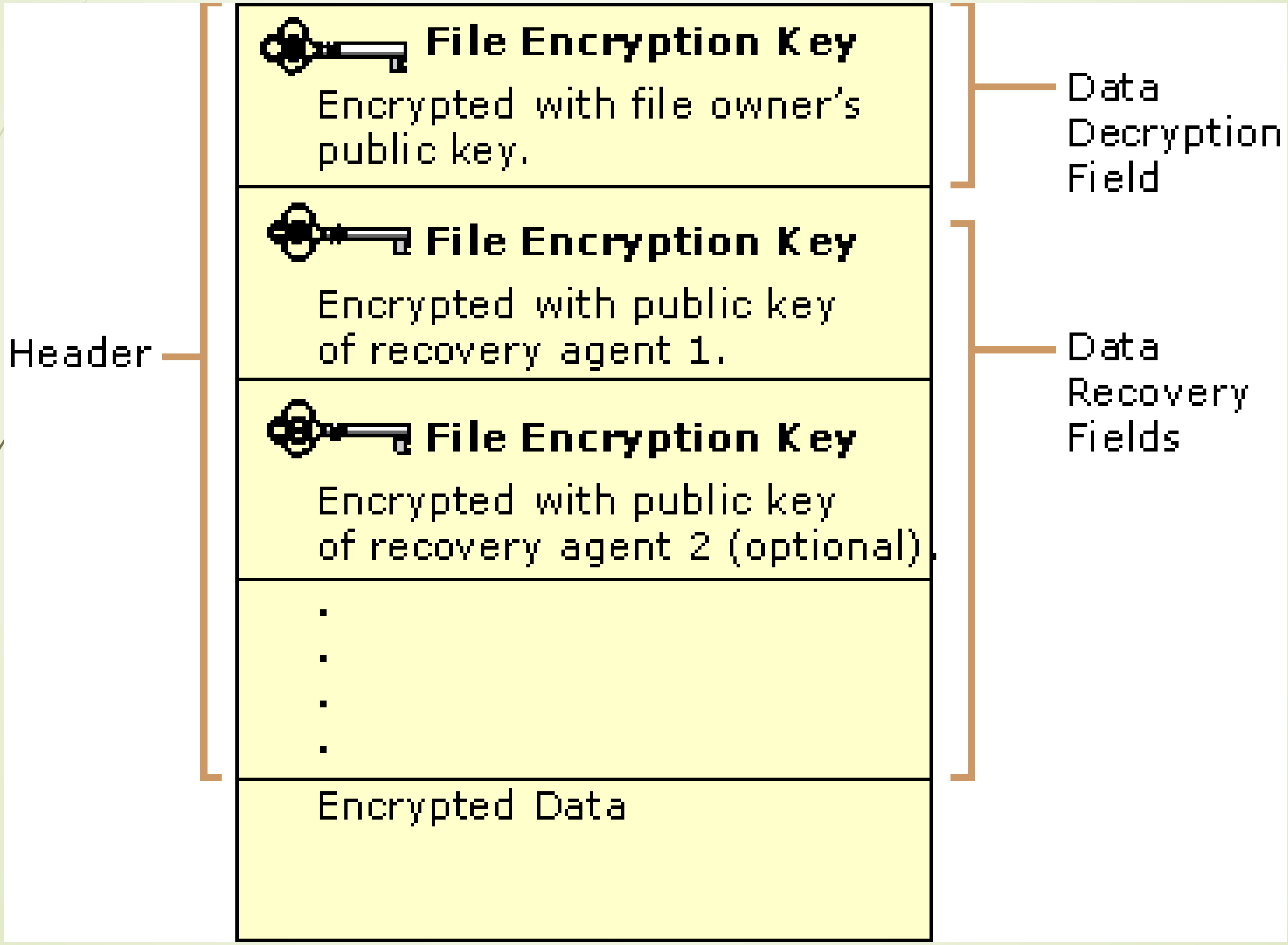
☐ Encrypt contents to secure data

Encrypting File System

- When EFS is activated
  1. User logon password => Passkey
  2. Passkey + User's protected information => Master Key
  3. A pair of private and public is created
     - Unique for each user
  4. Master Key encrypts the private key



Encrypting File System



File Encryption Key

Encrypted with file owner's public key.

File Encryption Key

Encrypted with public key of recovery agent 1.

File Encryption Key

Encrypted with public key of recovery agent 2 (optional).

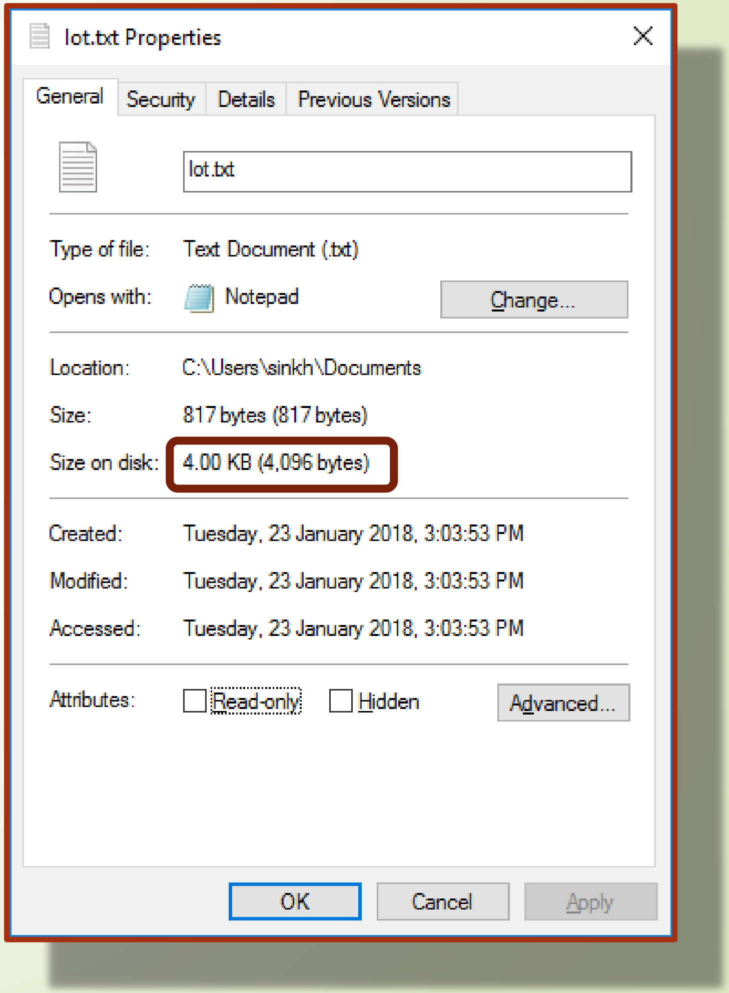
.
.
.
.

Encrypted Data

Slack and Unallocated Space in a disk drive

File Slack in a disk drive

- Lot.txt is a text file that has a size of **817 bytes**
- Why the size of the file on disk is **4,096 bytes** and it is not 817 bytes???



lot.txt Properties

lot.txt

Type of file: Text Document (.txt)

Opens with: Notepad

Location: C:\Users\sinkh\Documents

Size: 817 bytes (817 bytes)

Size on disk: 4.00 KB (4,096 bytes)

Created: Tuesday, 23 January 2018, 3:03:53 PM

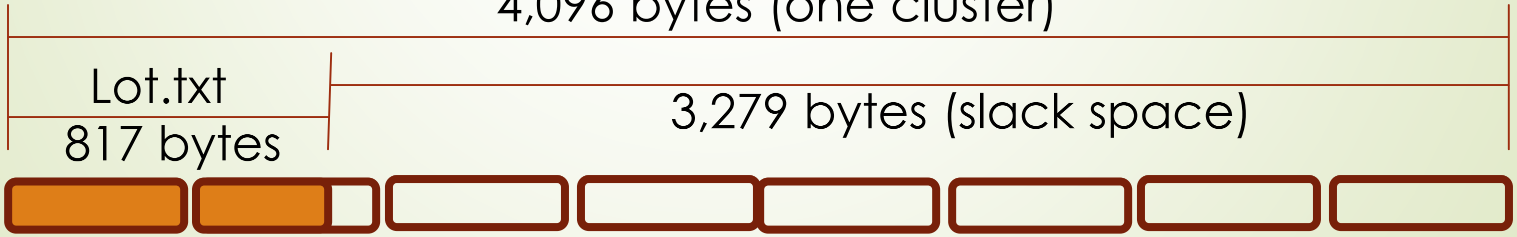
Modified: Tuesday, 23 January 2018, 3:03:53 PM

Accessed: Tuesday, 23 January 2018, 3:03:53 PM

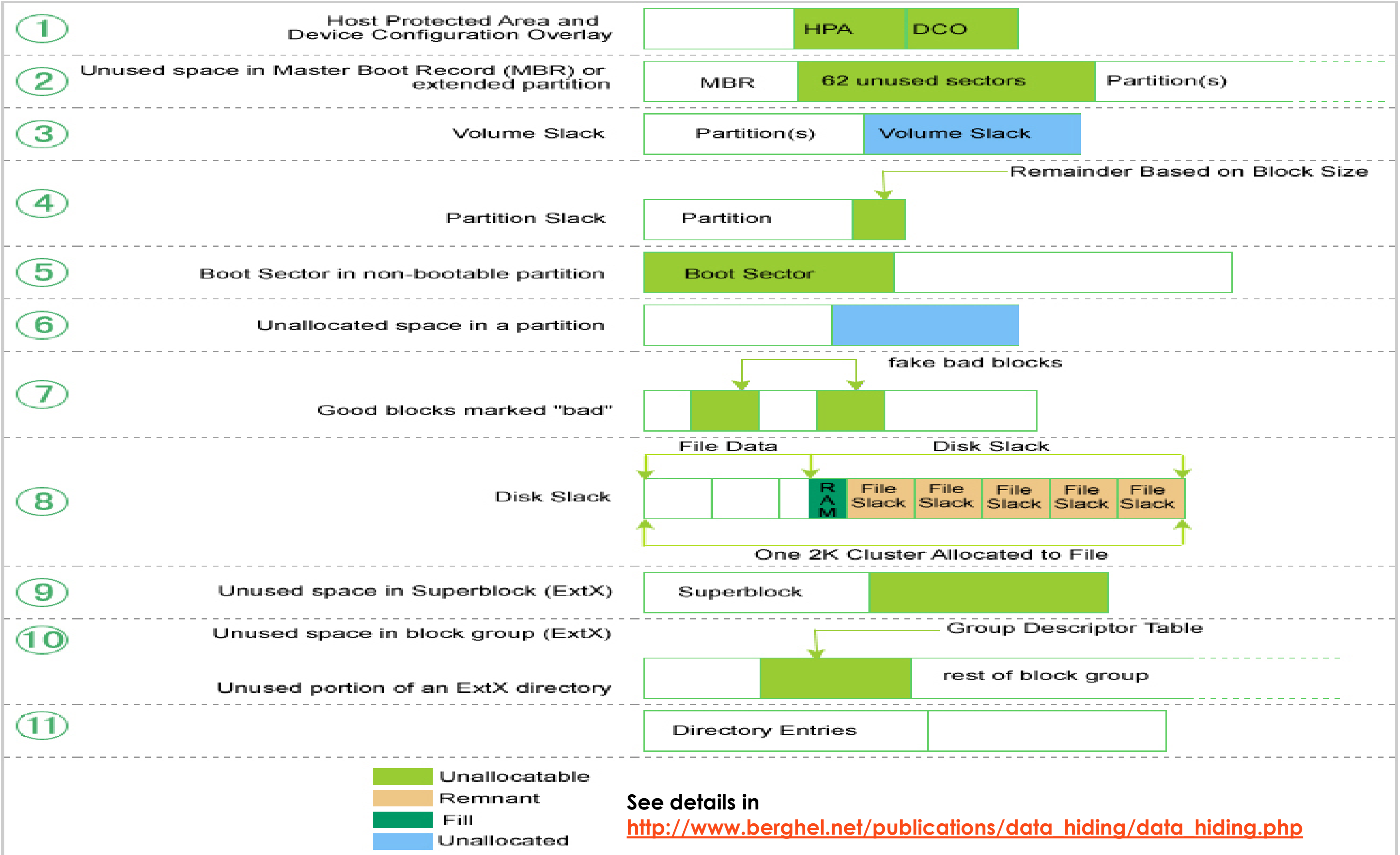
Attributes: □ Read-only □ Hidden

File Slack in a disk drive

- Lot.txt is a text file that has a size of 817 bytes
- Why the size of the file on disk is 4,096 bytes not 817 bytes???
- It is because NTFS allocates minimum one cluster with 4,096 bytes of storage (one cluster has 8 sectors and each sector contains 512 bytes)
- Lot.txt occupies 817 bytes and 3,279 bytes become slack space which may be used by hackers to store stolen information



Hidden Evidence in Hard Disks



1 Host Protected Area and Device Configuration Overlay — HPA, DCO

2 Unused space in Master Boot Record (MBR) or extended partition — MBR, 62 unused sectors, Partition(s)

3 Volume Slack — Partition(s), Volume Slack

4 Partition Slack — Remainder Based on Block Size, Partition

5 Boot Sector in non-bootable partition — Boot Sector

6 Unallocated space in a partition

7 Good blocks marked "bad" — fake bad blocks

8 Disk Slack — File Data, Disk Slack, RAM, File Slack, File Slack, File Slack, File Slack, File Slack, One 2K Cluster Allocated to File

9 Unused space in Superblock (ExtX) — Superblock

10 Unused space in block group (ExtX) — Group Descriptor Table, rest of block group

Unused portion of an ExtX directory

11 Directory Entries

🟩 Unallocatable
🟧 Remnant
🟩 Fill
🟦 Unallocated

See details in
http://www.berghel.net/publications/data_hiding/data_hiding.php

Unallocated Space (Free Space) in a disk drive

- Any space in a partition not currently allocated (i.e., unallocated space) , to a particular file cannot be accessed by the operating system. Until that space has been allocated to a file, **it could contain hidden data.**

Examples :

- unallocated sectors after MBR and before the first partitions
- Unallocated sectors after the last partitions
- Unallocated clusters of sectors within a partition that are not allocated to store files

36

Further Reading

- Read Section 4.1 “File Basics” in

Guide to Integrating Forensic Techniques into Incident Response SP 800-86, NIST

Summary

- There are different types of Windows file systems
- A disk drive has a master boot record that stores information of partitions
- Boot sector in a volume stores the information of the file system
- FAT contains linked lists of cluster numbers used by respective files
- NTFS uses $MFT to stores file information and starting record number of each file in the file system
- Slack and unallocated space in a drive and file system may store stolen information or malware