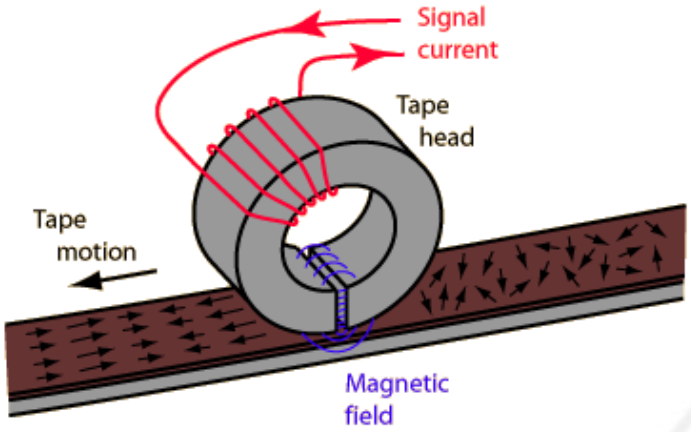
Troopers13
March 11-15, 2013
Heidelberg, Germany

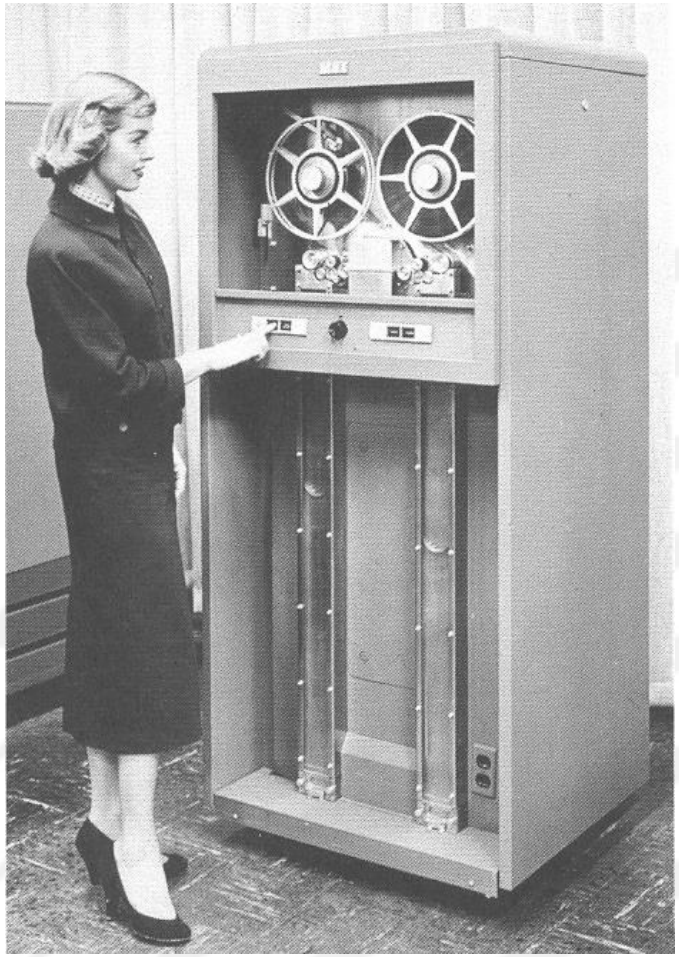
Flash Storage Forensics

Dmitry Sklyarov
Lead Analyst @ Positive Research
http://www.ptsecurity.com/

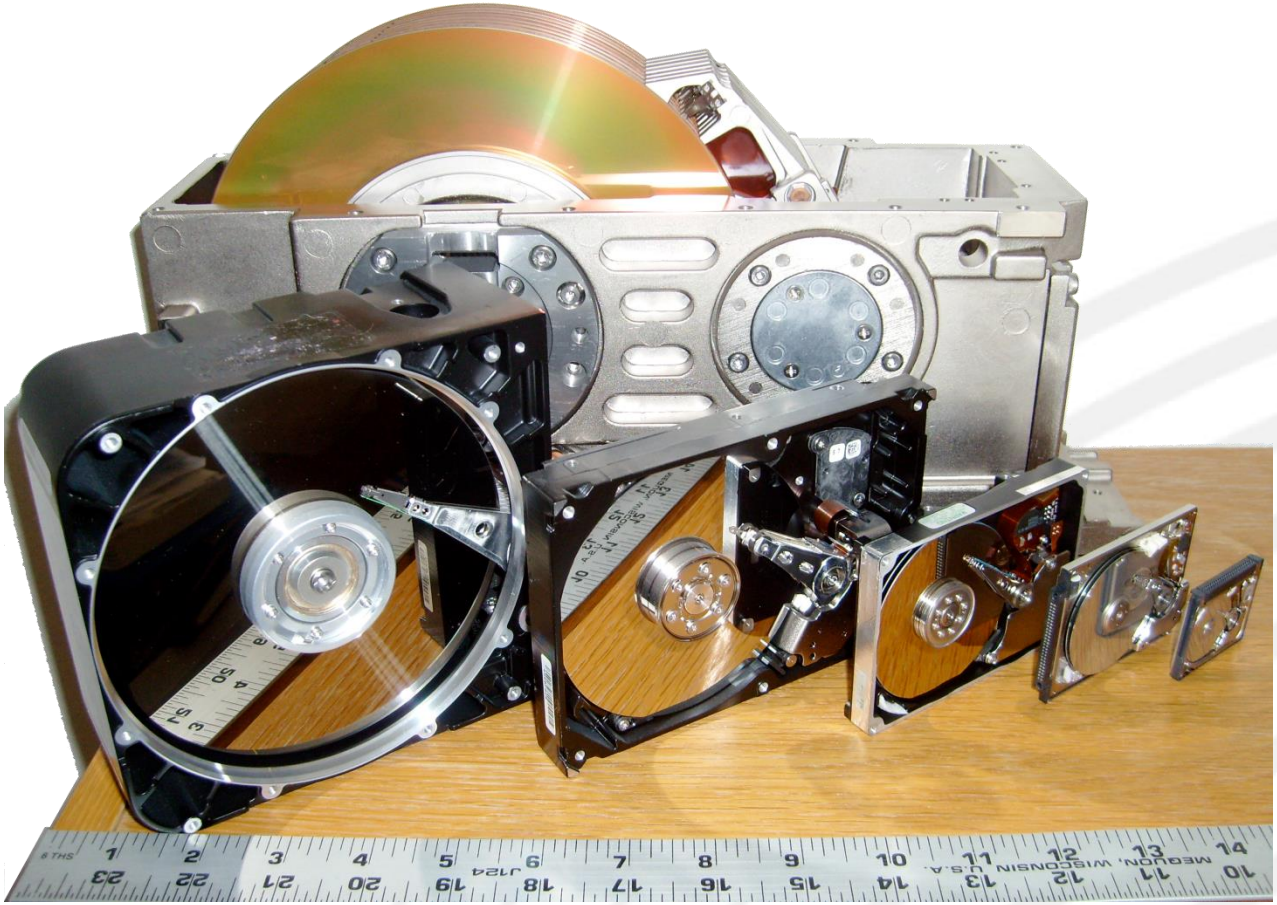
Magnetic Recording

- Invented in 1898
- Media moves near magnetic head

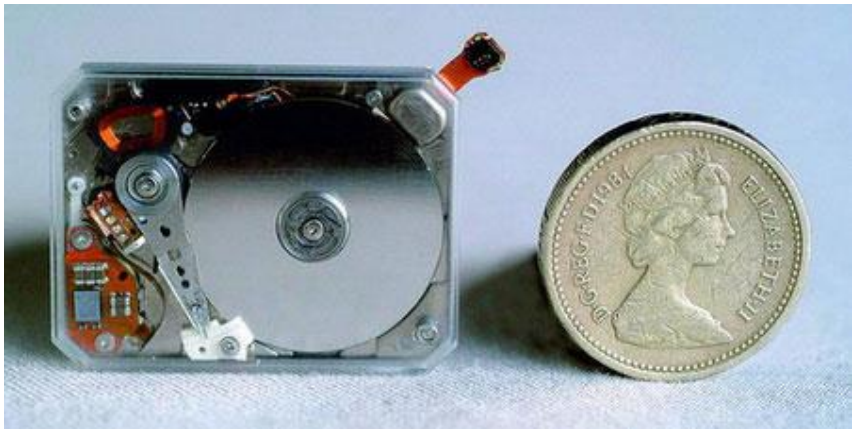




Magnetic drives becomes smaller …



… and smaller

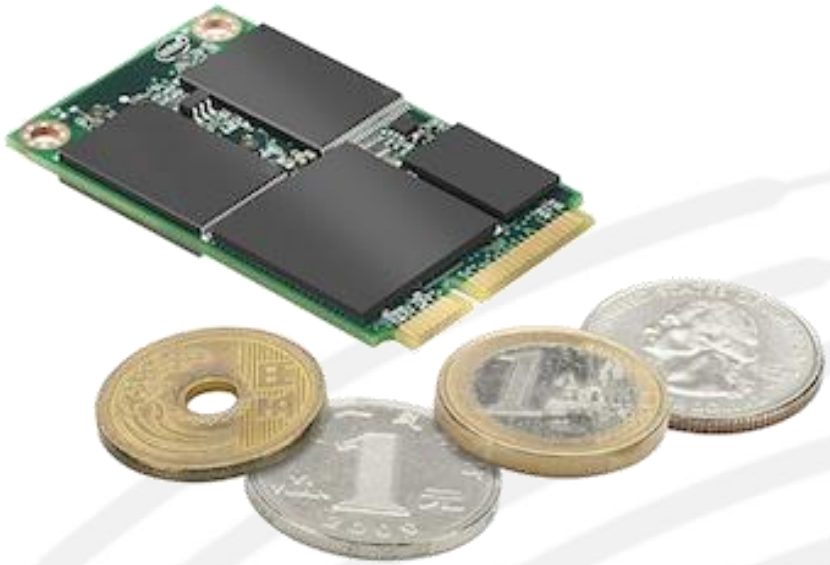


Toshiba's 0.85” 4GB HDD (2005)

- General principles still the same
- Any piece of data could be modified independently
- Erasing performed via overwriting
- Data erasure standards exists

Flash Memory

- Invented in 1984
- Two major types:
  - NOR (1988, Intel)
  - NAND (1989, Toshiba)
- Stores electrical charge into a floating gate of transistor
- Able to retain data for 10-100 years

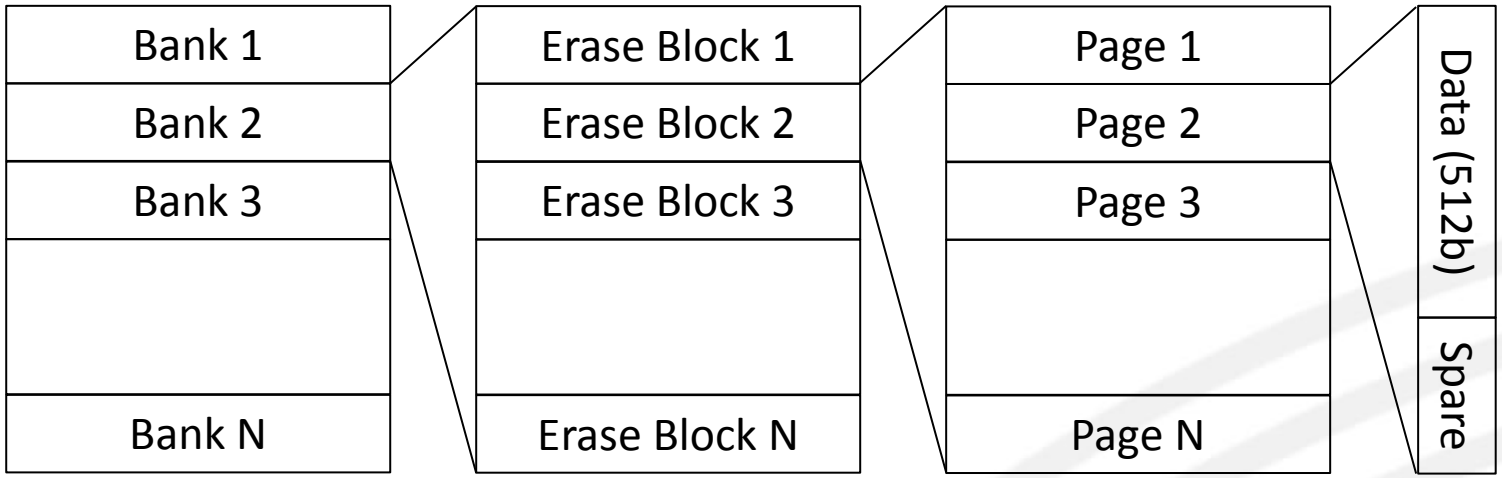


Intel's m-SATA 80G SSD (2010)

Flash Memory Characteristics

- Any byte could be written independently
- Need erase (make all bits=1) before re-writing
- Erasing with precision of block (e.g. 64K) only
- Limited number of guaranteed erase cycles
  - Usually between 10’000 and 1’000’000
  - Inerasable block should be marker as “bad”
- Some blocks could be inerasable when leaving factory

Flash Memory Layout



Spare area could be used for:

- Marking bad pages/blocks
- Storing ECC data
- Holding Physical-to-Logical mapping information

Wear Leveling

Goal: evenly spread the erasing of blocks over the full range of physical blocks

- Dynamic process that rearranges pages/blocks in order to extend flash lifetime
- Algorithms developed by memory device manufacturers
- Implementation details usually keeps secret

Logical Characteristics

USB Flash Drive, SSD or Flash Memory Card

- Simulates behavior of common HDD
- Logical Block Addressing
- Logical Address translates to Physical Address by Flash Memory Controller
- TRIM command for SSD

Logical Characteristics

Embedded Device (e.g. Smartphone)

- Flash-aware firmware
- Tight integration between OS and Flash Memory
- Logical-to-Physical translation often performed on CPU

Flash Translation Layer (FTL)

- Responsible for finding Physical Page that represents actual data for specific Logical Page Number (LPN) of Block device
- State of mapping tables is stored in Flash and cached in RAM
- Unused (TRIM-ed) LPNs are not mapped at all

Altering data in Flash Storage

- Any modification of data changes the mapping
- New data is written to new (free) page
- Previous version of page data (and content of TRIM-ed pages) still resides somewhere in Flash until block erased due to wear leveling or garbage collection

FTL in iOS devices

Implemented in software (runs on CPU)

Spare area of Data pages contains:

- LPN
  - allows to find all Physical pages that were used to store data of some Logical page
- USN (Update Sequence Number)
  - allows to build the ordered “history” of page copies

Accessing raw Flash on iOS devices

- `IOFlashControllerUserClient` kernel service is available
- `externalMethod` functions allows perform “raw” reading of Physical pages
- `ReadPage` request support removed in iOS 5
  - RAMdisk based on iOS 4 could help
  - It is possible to patch the kernel in memory and restore ability to read pages

Which devices could be examined?

- Anything prior to iPhone4S/iPad2/iPod5
  - by loading custom RAMdisk/Kernel
- Jailbroken device
  - by patching kernel in memory
- Any iOS device
  - if you know how to obtain digital signature for your RAMdisk from Apple

How “Forensic” is it?

Not too much…

- Booting the iDevice causes some alteration of Flash content
- Obtaining Flash dump twice would not produce identical results

Is there secure way to erase data?

- Deleting file produces good result at logical level (due to TRIM) – better that HDD
- Neither deleting nor overwriting are actually removes the data at physical level – much worse than HDD
- Probability of successful data recovery depends on amount of unused space on Flash Storage (more space – more chances)

Thanks!

Questions?

POSITIVE TECHNOLOGIES

Dmitry Sklyarov
Lead Analyst @ Positive Research
http://www.ptsecurity.com/