Dr. Enbody
Synopsis

- Forensics view
  - Delete actually deletes
  - Snapshots can differ even if no writes

- Security view
  - Delete doesn't delete

- Huh???
The Key: when do deletes happen?
Hard drive delete marks for “distant” future \textit{reuse}.

SSD delete marks for ASAP future \textit{erase}.
Delete depends on write
which is a function of technology
so we take a look at that.
Organization

1. SSD Technology
   why delete works differently; what does delete mean?
2. Forensics view
   differences that matter
3. Security view
   state of digital remnants
SSD

- Fast
- Low power
- No moving parts to fail
- All data available at same rate
- 1TB for $600 (2.5 inch by crucial)
A quick HDD, 4k random reads, 7ms seek latency

An SSD, 4k random reads, <1 ms seek latency

One millisecond
Actual read event

Ars Technica
Technology

NAND Flash
like in a USB stick
but on steroids
Floating-gate Transistors

- A floating gate transistor creates a tiny cage (the floating gate).
- Electrons are driven into or out of that cage (using a quantum tunneling effect).
- The charge those electrons represent is permanently trapped inside the cage (0 if charged, 1 if not).
NAND Flash
Read

To read an individual bit within the grid, the bit line select and source select transistors are both turned on.

Then, all the word lines that you're not reading have $V_{\text{read}}$ applied to them, forcing the transistors to conduct regardless of whether they have a charge in their floating gates.

Only the word line of the bit that you're interested in has the lower voltage applied to it.

If that bit has a stored charge (and hence is a logical 0), the transistor will remain open, and no current will flow through the chain of transistors.

If it has no stored charge (and hence is a logical 1), the transistor will close, and a current will flow.

The presence or absence of a current on the bit line is detected and treated as a 1 or 0, respectively.

All the bit lines will deliver a signal simultaneously, in parallel.
That is, whether or not the float gate contains a charge changes how much voltage must be applied to the transistor's gate in order for it to conduct or not conduct.

We can read the contents of the cell by applying a low voltage to the gates, and seeing if a current flows.
Cells grounded

Page to be read

Cells energized to greater than the voltage of a “1”

Cells energized to greater than the voltage of a “1”

Cells energized to greater than the voltage of a “1”

The cell does not conduct - it is charged so it is a 0

The cell conducts - it has no charge, so it is a 1

The cell conducts - it has no charge, so it is a 1
Organization

- A typical flash memory grid has
  - 32 to 256 columns
    (32 to 256 bit lines)
  - 4,096 to 65,536 rows
    (4,096 to 65,536 word lines).
- The total grid is called a *block*
- Each row is called a *page*
NAND flash

- NAND can be organized to use 60% less space than NOR—mostly wiring.
- But that space-saving organization means NAND flash can read/write data one page (row) at a time.
- That is, page is the smallest addressable unit.

In a modern SSD with 25 or 20 nm elements, the page size is 8192 bytes.
Erase

- SSDs can read and write to individual pages, but only erase in blocks.
- A freshly erased, blank page of NAND flash has no charges stored in any of its floating gates; it stores all 1s. 1s can be turned into 0s at the page level, but it's a one-way process (turning 0s back into 1s is a potentially dangerous operation because it uses high voltages).
- Consequently, in order to avoid corruption and damage, SSDs are only erased in increments of entire blocks.

block is 256 pages at 8192 bytes/page
Write procedure

1. Erase:
   whack *whole* block hard to force to all 1s
2. Write:
   flip individual bits to 0

(remember block is 256 pages at 8192 bytes/page )
Extra erase step takes time.

In a current-generation SSD with 8192-byte pages, a block can be made up of as many as 256 separate pages, meaning that to write a tiny 8KB file, the SSD must actually first copy two whole megabytes of data into cache, then erase the whole block, then re-write most or all of the entire 2 MB.

Note the digital remnants.
Here, a block starts with two free pages and one change that must be made to a page. The change goes to a new page, and the old page is marked stale. When a new file comes in and there aren't enough free pages, the whole block is read into cache or RAM and reordered, while the controller erases the entire block, then writes the entire block back out. This is why under certain circumstances, full or old SSDs can seem slow.
SLC, MLC, ...
- SLC = “Single-Level Cell” stores 1 bit
- MLC = “Multi-Level Cell” stores 2 bits
- TLC = “Triple-Level Cell” stores 3 bits

Note: 2 bits means 4 levels of charge to be managed and detected.
SLC is faster (only one level of charge to manage and read) and can handle more write cycles.

Used in enterprises. Very expensive.
Limited Writes

- Each time the cells go through a program/erase cycle, some charge is trapped in the dielectric layer of material that makes up the floating gates; this trapped charge changes the resistance of gates.

- As the resistance changes, the amount of current required to change a gate's state increases and the gate takes longer to flip.

- Over time, this change in resistance becomes significant enough that the amount of voltage required to write a 0 into the cell becomes so high, and the amount of time it takes for the cell to change becomes so long, that the cell becomes worthless.

- Cells become read-only.
MLC SSDs are much more susceptible to degradation than SLC SSDs, because each cell in an MLC drive has four possible states and stores two bits, and so each is more sensitive to changes in residual charge or difficulties adding new charge.

Additionally, flash cells are continually decreasing in size as we push further and further down the semiconductor process size roadmap.
Chip RAID

Ars Technica
Cache: few consumer SSDs contain some amount of DDR2 or DDR3 SDRAM, usually between 128 and 512 MB.
Over-provisioning

Over-provisioning provides some breathing room so that there can potentially be free pages available to receive writes even if the drive is nearing capacity.

Additionally, in the event that some cells wear out prematurely or go bad, over-provisioning means that those bad cells can be permanently marked as unusable without a visible decrease in the drive's capacity.
Garbage Collection

As the contents of pages are modified the SSD keeps track of which pages contain good data and which contain stale data.

Those stale pages aren't doing anyone any good just sitting there, but they also cannot be individually erased.

When the SSD controller has an opportune moment, it will garbage-collect those pages:

- it will take an entire block that contains both good and stale pages, copy the good pages to a different block, and then erase the entire first block. or not—leaving digital remnants
TRIM

OS-initiated garbage collection

OS knows what pages are stale so it can instruct the SSD to consolidate pages and erase blocks.
Why?

TRIM and garbage collection speed up writes by increasing the number of available erased pages while reducing fragmentation.

But at the cost of redundant writes. … and possible digital remnants
Write Amplification

TRIM and garbage collection increase the number of (redundant) writes so they increase wear!
Wear leveling

If cells wear out, it is best to spread the writing around. Controller will swap static pages with active pages to level wearing. The result is more redundant writes.
DELETE
Delete actually deletes
- TRIM and garbage collection will eventually erase data.

Snapshots can differ
- With TRIM, garbage collection and wear leveling working in the background, a system with no writes is erasing data.
Sanitizing

Reliably erasing data from storage media is critical for secure data management.
Delete doesn’t mean “delete right now.”
Whack and copy

Remember, writing a page involved whacking a block and copying pages to the new location.
The original data remains until TRIM or garbage collection erases it.
Wei et al. in 2011 tested a pile of SSDs for secure erase by ripping them apart to test individual chips for data, i.e. bypassing all controls.
ATA Secure Erase

The ATA standard secure erase should erase all cells.

Wei et al. showed that most but not all implemented it correctly. Furthermore, verification of correct erase isn’t feasible.
Overwriting

Overwriting a hard drive erases data. On SSDs it works some \textit{but not all} the time, often leaving more than 2/3 of data. Digital remnants from whack-and-copy can be hidden due to over-provisioning and wear leveling. Also, worn-out cells become read only.
Digital Remnants

Wei et al. found

- over-provisioning at 6% to 25%.
- 16 stale copies of some files.
Cell deletion

- MLC get deleted with one overwrite
- SLC may take a second overwrite

Therefore, the problem with ineffective overwriting on SSD is *not* cell technology. Rather, the cells are skipped.
Degaussing

No effect on SSD.
Encryption

Some SSDs encrypt all data—managing encryption in hardware, some also compress all data, at least one also doesn’t store duplicate pages.

Erase is sometimes done by only erasing the key (e.g. Apple Filevault2).
Verification

Verification of deletion of a fully encrypted disk is impossible (original data is random).
Full Disk Sanitation

Do you trust the vendor’s implementation?

especially since you cannot verify erasure.
Individual File Sanitation

Wei et al. tested 13 published protocols (including government standards) for overwriting files on hard drives and none worked for SSD leaving between 4% and 75% of the data.
ACS-2 Proposed Standard

The ACS-2 proposed standard includes “BLOCK ERASE” for partial disk sanitation.

No drives implement it (because it is still only a proposal).
Wei et al. also tried overwriting free space and defragmenting a drive before overwriting, but neither improved results.
Secure Erase

Securely erasing a disk is *not supported* on all drives.

Securely erasing a file is *not supported* on any drives.

Overwriting and degaussing are *not effective*. 
Notes

Wei et al.’s results with respect to digital remnants didn’t allow for TRIM and garbage collection so don’t count on those results for forensics.
Hard drive delete marks for “distant” future reuse.

SSD delete marks for ASAP future erase.
References

- Ars Technica Primer

Thank you!

Questions?