Digital Forensics 1: Introduction and Concepts

FORS 201

Chris Edwards

School of Computing

Semester One 2024

Section 1

Introduction

Digital and Electronic Forensics

...is the application of forensic techniques to computers and other electronic devices.
Many familiar forensic concepts apply:

- Reliance on scientific principles
- Forensic evidence as circumstantial evidence
- Need for expert witnesses
- Processes of discovery and analysis
- Handling ("tagging and bagging") and chain of custody
- Validation of forensic techniques (to be legally admissible)
- Uphold ethical and legal principles (not "nail the perp")

Digital and Electronic Forensics

Digital forensics uses knowledge of ICT (Information and Communication Technology) in application to the law.

Work in digital forensics requires a broad range of technical knowledge. Digital forensics personnel may work primarily in a laboratory setting, processing case data, and may act as expert witnesses in court (for prosecution or defence). Digital forensic scientists may have areas of specialisation, e.g. Android phone/tablet forensics. Others may work primarily on researching new digital forensic techniques, and may overlap with information security (InfoSec) and cybersecurity fields.

A Forensics Ethos

- Search for the truth
- Appreciate limits of certainty
- Conduct work without bias or prejudice
- Can work for either side, but only one at a time
- Be methodical, and document everything
- Be prepared to defend, demonstrate, and duplicate methods

Data vs Information

(Information scientists like to distinguish between the two.)

- Data are raw values, e.g. **9.5**
- Alone, a datum does not mean anything *per se*
- *Information* is contextualised, e.g. "This DVD costs NZ$9.50 incl. GST"
- Information *informs*, and supports decision-making
- Computers are fundamentally *data*-processing machines

Information Quality

Some measures:

- Relevance
- Precision
- Accuracy
- Completeness
- Timeliness
- Conciseness
- Presentation (e.g. sorting, graphing)

Information Quantity

Some measures:

- bit — an information "atom", having two possible values (0/1, True/False)
- byte (8 bits) — 1 text character (from an alphabet of 256)
- 32 bits — 1 colour pixel (RGBA: Red, Green, Blue, Alpha (transparency))
- 64 bits — typical binary integer
- 512 or 4096 bytes — disk sector
- kilobyte (1000 bytes)
- megabyte (1 million bytes)
- gigabyte (10^9 bytes)
- terabyte (10^{12} bytes)
- petabyte (10^{15} bytes)
- exabyte (10^{18} bytes)
- Also binary (2^n) variants, e.g. kibibyte (1024 bytes)

Electronic vs Digital Forensics

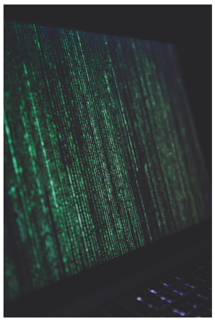
- Not all electronics are digital (some are *analog*)
- Not all digital systems are electronic
- However, there is generally extensive overlap
- Digital electronics pervade our daily life
- Digital devices often record data constantly and automatically

Analog Data

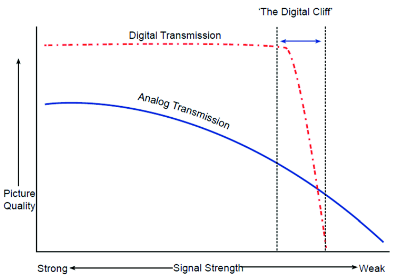
- Analog as in *analogy*, e.g. mechanical, electrical or magnetic analog of a signal
- Suffers from *generation loss*
- Perfect copies are impossible
- Generally degrades gracefully
- Examples: vinyl records, music cassettes, printed books, photographic film

Digital Data

- Data represented numerically, discretely
- Typically binary (zeroes and ones, or logical True/False values)
- Can be copied (*cloned*) without loss or limit
- Commonly encoded physically using electromagnetic signals (voltage, current, magnetism, light/radio)
- Degrades catastrophically (the "digital cliff")

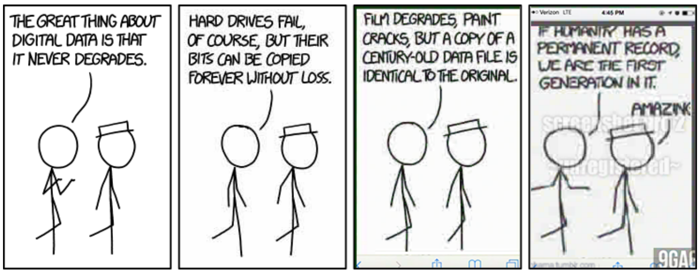


The Digital Cliff



Source: Australian Government (2013): https://www.abcb.gov.au/-/media/Files/Resources/Education-Training/Handbook-Digital-TV-Antenna-Systems-2013.pdf

xkcd: The Failed Promise of Digital Data



Source: https://xkcd.com/1683/

Section 2

Binary Data and a Brief History of Computing

Computers and Binary

In computer systems, information is usually coded as binary data, i.e. as strings of ones and zeroes.

Bits and Bytes

The *bit* is the smallest unit of information (an information atom, if you will). One bit represents one of two possible states (often thought of as 1/0, true/false, on/off).

If you need more states (say for an entire alphabet), assemble multiple bits into larger groups, e.g. into 8-bit bytes.

Binary Codings

To carry meaning, there must be some *interpretation* that can be systematically applied to strings of binary digits. These are known as binary *codings* (or *encodings*).

Without knowing how it is structured and coded, any given binary string is meaningless.

For example, in ASCII text, the *code point* for A is:

$$\texttt{0b1000001} = 65_{10} = \texttt{0x41} = \text{“A”}$$

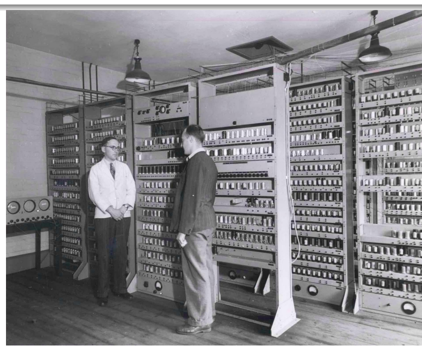
Electronic computers have gone through several significant eras, characterised by the kinds of data/information they were processing...

The Numeric Age (1940s–1950s)

The very earliest computers dealt only with numbers: typical applications were science and engineering: trajectory calculations for ballistics, chemical simulations, etc.

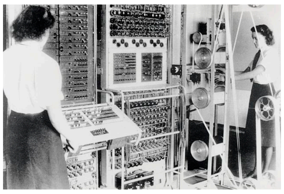
The basic unit of memory was the numerical “word”, often 18–40 bits in length, giving around 6–12 decimal digits of precision. Memory was small: the early EDSAC computer initially had just 512 words of 17 usable bits each (equivalent to about 1000 characters of text, although these machines had no built-in support for something so frivolous!).

EDSAC (1949)



Colossus (1943)

- First electronic programmable computer
- Not a general-purpose, stored-program computer
- Special-purpose cryptographic processor (deciphering the German Lorenz or “Tunny” cipher)
- Boolean logic, implemented electronically (vacuum tubes)
- Remaining Colossi used by GCHQ, scrapped by 1960
- All documentation was deliberately destroyed
- A rebuild project was started in the 1990s



A Colossus in 1943

The Text Age (1960s–1970s)

Soon, computer applications (especially commercial computing) needed to deal with text. The 5-bit Baudot code had been used in the days of telegraph, and provided enough code points for the basic alphabet ($2^5 = 32$ characters); see 5-hole paper tape.

Later computer applications needed mixed-case text, digits, punctuation, special control characters, etc., and coding schemes such as ASCII and EBCDIC were devised in the 1960s. Various non-Latin alphabet encodings were also devised internationally.

In the 1990s, the need for a global text encoding gave rise to the Unicode project.

IBM 360

The Multimedia Age (1980s–)

With the rise of personal computers, data processing expanded to include still images, multi-channel audio, video, 3D polygons and voxels, etc. These too require new binary coding schemes (especially ones providing data compression). However, fundamentally, the computer is still just processing streams of 0s and 1s.

Amiga 1000

Source: b52_rock_lobster, Instagram, https://www.instagram.com/p/BuZLhpEF4Lm/

Binary Number Systems

- Modern digital computers use binary for all internal processing and storage
- Many reasons for this, but mainly comes down to simplicity and efficiency
- Binary works in analogous ways to the familiar base-10 (decimal) system

Let's examine how decimal works systematically...

Decimal Deconstructed

Let's examine the real meaning behind a decimal integer such as 372. This actually stands for an expanded expression involving multiplications and additions:

```
      372
        =
  3 x 100
+ 7 x  10
+ 2 x   1
```


Decimal Deconstructed

Basic scheme:

- 10 digits, 0...9
  (Q:Why are they called *digits*?)
- Each position signifies a different *order of magnitude*
- Moving left one digit: $10\times$
- Moving right one digit: $\frac{1}{10}\times$
- Most significant digit is at left (*big-endian* notation)
- A 1 followed by a string of 0s means a power of 10.
- A string of 9s means one less than a power of 10.

Decimal Deconstructed

If you number the digit columns from right starting at 0, the pattern becomes very clear and systematic:

Column:	...	3	2	1	0
Place Value:	...	10^3	10^2	10^1	10^0
Place Value:	...	1000	100	10	1

You can see why the term *base-10* is used: the place value is an exponential expression with 10 as the base.

Denoting the Base

In mathematics, the base is by convention denoted using a subscript, e.g. 01001011_2. Computer languages often use other notations, such as a leading **0b** for binary, a leading **0x** for hexadecimal, and a leading \ for octal.

Binary Numbers

The binary scheme is essentially analogous to decimal, but with 2 instead of 10 as the base:

- Only 2 digits, 0 and 1
- Each position signifies a doubling or halving
- Moving left one digit: $\times 2$
- Moving right one digit: $\div 2$
- A 1 followed by a string of 0s indicates a power of 2.
- A string of 1s means one less than a power of 2.

Binary Numbers

Column:	...	3	2	1	0
Place Value:	...	2^3	2^2	2^1	2^0
Place Value:	...	8	4	2	1

The same pattern as for decimal, but with 2 as the base.

Binary Example

Take the 8-bit unsigned binary integer **0b00010100**:

Place Value:	128	64	32	16	8	4	2	1
Face Value:	0	0	0	1	0	1	0	0

Total Value: $1 \times 16 + 1 \times 4 = 20$

Octal Numbers (not examinable!)

This is a base-8 system and is occasionally encountered in computing (e.g. Unix filesystem permissions, PostgreSQL binary data entry). Each octal digit (0–7) represents 3 bits of information.

Column:	...	3	2	1	0
Place Value:	...	8^3	8^2	8^1	8^0
Place Value:	...	512	64	8	1

Example: $61_8 = 6 \times 8 + 1 \times 1 = 49$

Hexadecimal (“Hex”)

Since data in computers are often grouped into 8-bit bytes ($2^8 = 256$ values), we could conveniently represent binary data using pairs of base-16 digits (each holding 4 bits). Unfortunately we run out of “natural” digits at 9, but we can enlist the first letters of the alphabet to make up the rest (10=A, 11=B, ..., 15=F).

Column:	...	3	2	1	0
Place Value:	...	16^3	16^2	16^1	16^0
Place Value:	...	4096	256	16	1

Example: **0x7F** $= 7 \times 16 + 15 \times 1 = 127$

Hexadecimal in Practice

Hexadecimal appears frequently in computing and information assurance, whenever binary data need to be presented in a more human-readable form. For example, file fingerprints, cryptographic keys, digital signatures, IPv6 and link-layer MAC addresses in networking are usually written in hex.

Binary Numbers (summary)

Now you know why computer people:

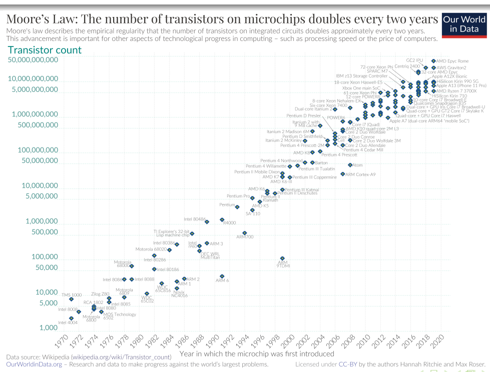
- like to count from 0, not 1
- can often recite many powers of 2
- sometimes confuse Christmas and Halloween (Oct 31 = Dec 25)!

Exponential Trends: Moore’s Law

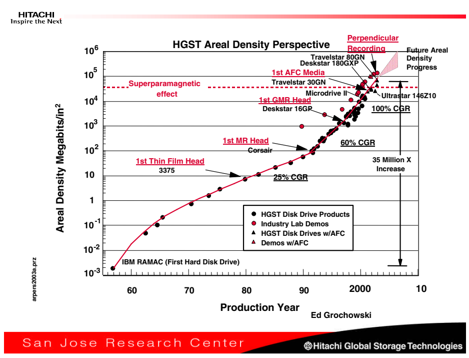
Advances in computing technology have been characterised by exponential growth, most famously in Moore’s Law.

- Observation that transistor count on integrated circuits doubles every 1–2 years
- First posited in 1965 by Intel co-founder Gordon Moore
- Rate of advancement increases over time (hurtling toward The Singularity?)
- Similar trends have held for storage capacity, CPU clock speeds, etc.
- Other fields benefit proportionately (e.g. gene sequencing)

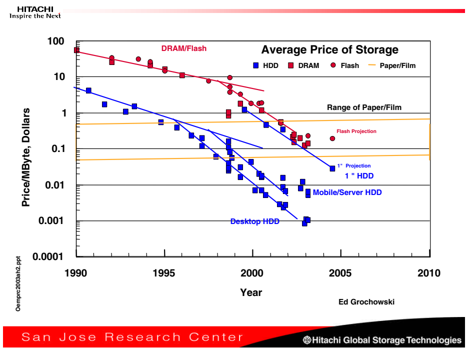
Exponential Trends: Moore’s Law



Exponential Trends: Storage Density



Exponential Trends: Storage Cost



Today: Computers Everywhere

In the modern era, computers are not just boxes in server rooms and on desktops.

- Desktop and laptop computers
- Servers and cloud storage
- Network routers and network storage devices (NAS/SAN, media boxes)
- Smart TVs, set-top boxes, Chromecasts, PVRs

...and I mean EVERYWHERE

- Mobile phones, tablets, watches, glasses
- Digital cameras
- Photocopiers
- Answering machines/services
- Cars, trucks, trains, planes, boats, blimps, spaceships, ...
- Building/plant control systems
- Internet of Things (IoT) devices (fridges, coffee machines, ...)
- Embedded systems generally

All could hold forensically significant data.

Section 3

Digital Forensic Procedure

Computers as Witless Witnesses

Computers are data-processing machines. They are *stupid*. They have no common sense, initiative, or understanding of the real world, and will do only what they are told (programmed) to do.

Computers as Witless Witnesses

Think of a computer as an extremely naïve witness with a photographic memory.

Computers as Witless Witnesses

Therefore, *you* will have to tell their story for them.

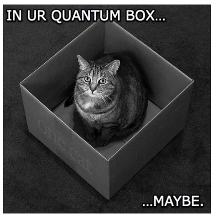
Expert Witnesses

Hence the need for expert witnesses. Some tips:

- Discuss issues with lawyers before testifying
- Don't make unnecessary notes (could be subject to discovery)
- Be brief
- Don't volunteer explanations (wait to be asked)
- If you don't know or don't recall, be honest and don't guess
- Ask to consult your notes if necessary
- Speak slowly and clearly to aid understanding, transcription

Observer Effect

Gathering information from a running system can be very challenging, not least because of the *observer effect*. To minimise this, your first priority should be to capture and preserve the information available.



Primary Tasks for the Digital Forensic Investigator

1. Triage: examine equipment for possible relevance (use write-blocker if browsing)
2. Acquisition: take a bit-wise copy (evidentiary image)
3. Authentication: verify that the copy matches the original
4. Analysis: search, recover, link
5. Reporting: compile, analyse and present evidence; defend methods

Preparing for an Electronic Investigation

- Plan the seizure
- Know what you're looking for
- Watch for other electronic media
- Supplement with physical evidence
- Anticipate complications
- Document thoroughly

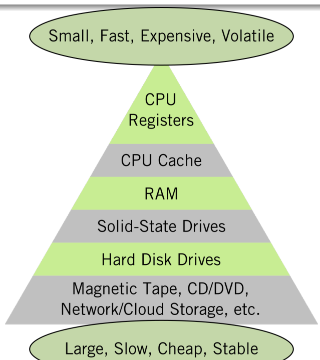
Section 4

Digital Data Storage

Digital Storage and Forensics

Digital forensic work is generally concerned with information that has been stored in the memory of a computer or other digital electronic device. Some of the primary objectives are securing and preserving the data, verifying and ensuring its integrity, searching (discovery) for relevant data, and analysing, interpreting and presenting the findings.

Computer Memory Hierarchy



Volatility

An important concept in digital forensics is *volatility*—how readily data will vanish when power is removed. Volatile media include:

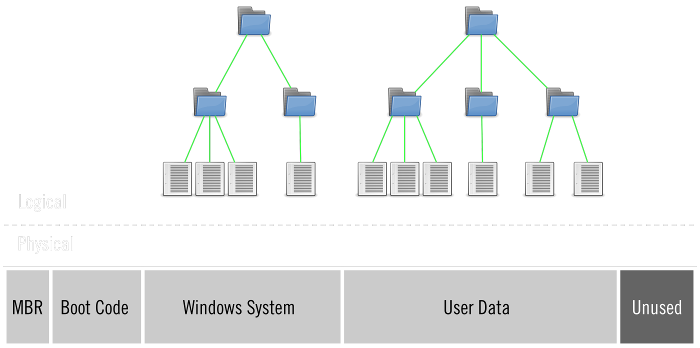
- CPU registers and cache
- (D)RAM (main memory)[1]
- Data "in flight" on the network
- Pixels on a display, sound output, other I/O

(Relatively) non-volatile media:

- Hard disk storage
- Flash memory and SSDs
- Hard copy (paper printout)
- Some next-generation memory technologies (e.g. MRAM)

[1] Interestingly, DRAM can retain data for up to several minutes if kept cool.

Logical vs Physical Storage Layers



Block Storage Structure

Digital storage devices (hard disks, flash drives and SSDs, CD- and DVD-ROM, even floppy disks and tape) have traditionally subdivided their capacity into uniform chunks known as blocks or sectors.[2]

- To the outside world, a modern hard drive appears as a one-dimensional array of sectors
- Addressed using sector address (LBA or Logical Block Address), e.g. "read from sector 1234", "write <data> to sector 9876"
- Earlier drives would expose a (real/fictitious) geometry of cylinders, heads, sectors
- A long-standing tradition for hard drives was 512 (user) bytes per sector

[2] We prefer the term *sector* when referring specifically to the device storage level, and *block* for the logical filesystem and above.

Block Storage Structure

LBA 0	LBA 1		LBA *n*
0x48 0x65 0x6C 0x6C 0x6F ...	0x53 0x65 0x63 0x6F 0x6E 0x64 ...	...	0x49 0x69 0x6E 0x61 0x6C ...

512 bytes per sector

Disk Partitioning

A single storage device such as a hard disk or SSD may be divided into multiple regions or *partitions*. At the beginning of the drive will normally be a *partition table* describing how the disk is broken up. In traditional PCs, this is combined with initial boot code in the Master Boot Record (MBR) at LBA 0. Secondary boot code may reside in the following sectors. Subsequent partitions normally contain filesystem structures, and there may be unused or "slack" space at the end or between partitions.

MBR	Boot Code	Windows System	User Data	Unused

Section 5

Acquisition

Triage

When performing a forensic system capture, attend to the most volatile storage first. Only then should you move on to less volatile media.

A cautious computer criminal may take steps to stop investigators finding incriminating data. Software and hardware booby-traps are not unheard of, and attempting to use the captured system normally may result in important data being irrevocably wiped.

The initial dealing with a suspect system should be handled by trained personnel following the correct procedures.

Evidentiary Copying (Imaging)

- Process of *cloning* data from the original media for analysis
- Preserves original media by minimising its use
- Data may reside in deleted files, hidden partitions, unused "slack space" within filesystems or between partitions, etc.
- Ordinary "drag-and-drop" copying is not sufficient!
- Instead, use a bit-wise, sector-by-sector copy (image) of *all* data, from sector 0 to the last addressable sector
- Copies can be *validated* as complete and unaltered
- Discovery and analysis can proceed in parallel on multiple copies

Evidentiary Copying (Imaging)

Basically, turning a *disk* into a *file*.

Disk Imaging Tools

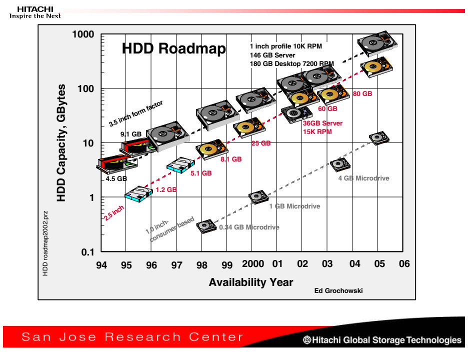
- Evidentiary copying should avoid modifying the original media
- May be able to set drive jumpers for read-only operation
- Can use a hardware write-blocker such as Tableau
- Software write-blockers also exist, may be less reliable
- Free/open-source imaging software exists, e.g. **`ddrescue`**
- (You might find **`ddrescue`** useful for personal data recovery too)

Disk Imaging Challenges

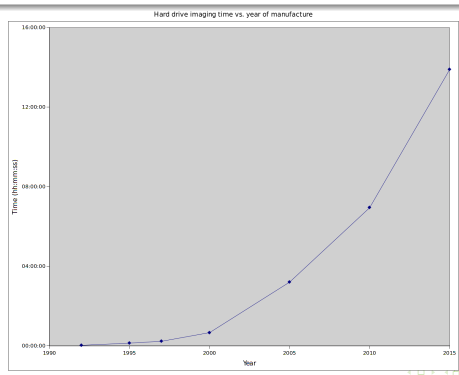
Evidentiary copying can be a time-consuming process. Hard disk capacities have increased exponentially, but sequential read performance has languished. As a result, a large modern drive can take the best part of a day to image.

Also, because of the proliferation of digital devices and the tumbling cost of online storage, the amount of data per person is increasing dramatically all the time.

Another exponential trend: Storage Capacity vs Time

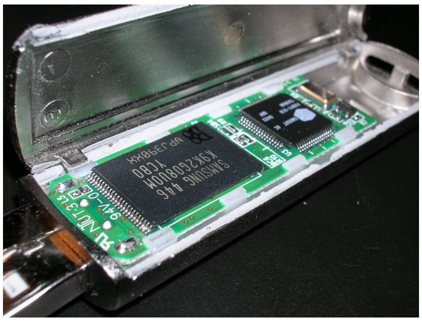


Time to Image vs Year of Manufacture



Flash Memory Complications

Solid-state flash memory (as used in USB flash drives, smartphone and camera storage, SD Cards, SSDs, hybrid drives) has unusual characteristics that make it more challenging forensically.



Flash Memory Complications (not examinable)

Specifically:

- Flash memory is arranged in blocks that are programmed together.
- Flash memory cells have a limited lifespan (in terms of write cycles)
- Block erase is more complex (e.g. Trim command)
- Controllers perform wear-levelling and garbage-collection operations
- Filesystem defragmentation neither required nor recommended
- No longer a clean 1:1 mapping between addressed user sectors and physical storage

However, flash memory is physically very robust—a flash memory chip in a device that's been run over by a car will likely still be readable.

Obscure Media

- Canny criminals may choose to use obscure or obsolete storage media
- Floppy diskette, magnetic tape, etc.
- May require specialised hardware and know-how

Unreliable Media

Old or damaged media may not be 100% readable. There may be bad sectors, corruption of filesystem structures, or faults with drive electronics.

Tools such as **`ddrescue`** can perform exhaustive retries, quickly reading good sectors, and mapping out unreadable regions for later retry (also useful for personal data recovery).

Validating Evidentiary Images

To ensure the integrity of an evidentiary image (preserving the chain of evidence), evidentiary copies must be hashed (fingerprinted).

This also allows other investigators to clone and perform investigations in parallel while being assured that their copy holds the same data as the original.

Cryptographic Hashes ("Digital Fingerprints")

Think of a cryptographic hash value as like a digital fingerprint: practically unique for a given input.

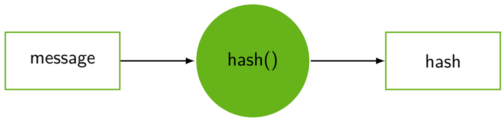
What is a hash function?

A hash function yields a small, distinctive value (the *hash* or *digest*) from arbitrarily-sized input (the *message*).

Often characterised as a *one-way function*.

Hash value often displayed as hexadecimal, e.g.
`adc83b19e793491b1c6ea0fd8b46cd9f32e592fc`

Hashing: General Scheme



Hash Functions: Ideal Properties for Forensics

- Maps input data of arbitrary size to a fixed-size output
- Accounts for every bit of information in a message
- Highly sensitive to changes in input (avalanche effect)
- Deterministic (same input gives same output)
- Non-invertible (one-way)
- Extremely difficult to forge

Hashing: Other Uses

- Anti-virus/anti-malware
- Preserving data backups and archives
- Authenticating software to be installed
- Cryptography in general

Linking the Accused to the Equipment

- Interview the owner/user if possible
- Take physical evidence: photographs, video, fingerprints, hair, etc.
- Work to establish and document the chain of custody
- Data analysis may uncover further evidence linking to the accused

Documentation

Take sufficient documentation to be able to reconstruct the configuration of seized equipment in the lab.

In Summary...

- Digital devices are rich sources of forensic information
- ...but they need special handling
- Use disk imaging to capture all available data
- Use hashing (fingerprinting) to ensure integrity

Next Time...

- Data discovery and analysis
- Digital data preservation
- Hiding and recovering data
- Cryptography and steganography

Further Reading

- Farmer and Venema's *Forensic Discovery* (free e-book)
- Security expert Bruce Schneier's writings on computer forensics

The End

Thank You!