Re-designing computer systems for reliable electronic evidence

Professor Steven Murdoch

University College London

September 2024

When computer systems fail, the legal system is there to undo the mess

- Computers store data, process interactions, and enforce rules
- They are imperfect, due to known (e.g. design constraints) and unknown limitations (e.g. bugs)
- The legal system is the universal undo, which tries to put things back where they should be
- I’ve been involved in numerous court disputes as an expert witness, including on banking security, anonymous communications, Internet traceability, etc…
- Mostly, the legal system works, but there are frustrating problems

Electronic evidence of some form is extremely common in legal cases (90%+)

- The system would collapse if every piece of electronic evidence was challenged for validity
- Every piece of software has bugs; many could be relevant to the evidence they produce
- Designing software to a high degree of accuracy is prohibitively expensive in all but the most critical scenarios
- Still, society is willing to rely on computers in daily life
- **How can the legal system use electronic evidence in the interests of justice, and how can computer system designers help?**

Electronic evidence is shoe-horned into how oral testimony is made, for better or worse

- Legal systems have built up a history that long predates computers
- A fundamental principle is making public statements in court, and being subject to examination
- Penalties for perjury encourage truth; lawyers/judges are skilled at identifying inconsistencies
- Principles for handling computer evidence is based both on documents and witnesses

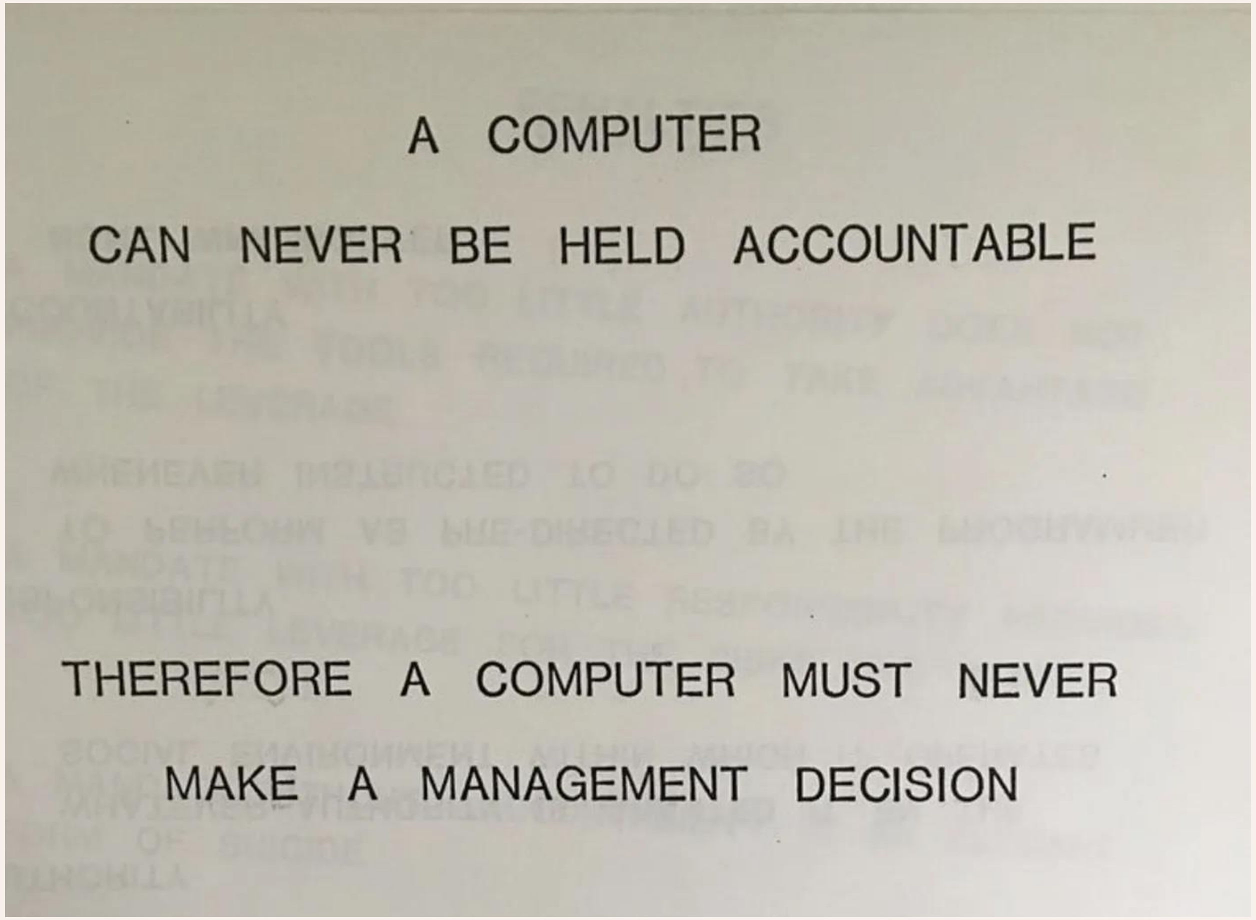
A particular way this has relevance is whether electronic evidence is classed as “hearsay”

- In general, evidence more directly associated with the facts in dispute is preferred by the court
- In particular “[a] statement other than one made by a person while giving oral evidence in the proceedings is inadmissible as evidence of any fact stated” (Cross, 1979)
- Mistakes could have been made about how the fact was perceived or how the statement about the fact was heard or remembered
- The person who originally perceived the fact cannot be cross-examined and so one of the fundamental mechanisms to assess trustworthiness cannot be used

It is sometimes unclear whether electronic evidence is hearsay

- If an electronic document includes someone’s notes, then it’s clear that the person who made the notes should be made available for cross-examination if needed
  - There are other exemptions to the hearsay rule if this is not possible
- If the electronic document contains the results of a measurement device (e.g. from an alcohol meter), then it could be considered to be real evidence and not hearsay
  - This doesn’t mean that the evidence is accepted as true, of course
- The hearsay rule can result in some counterintuitive results, and is becoming less relevant but is still a helpful concept

Unlike witnesses, computers cannot be challenged through cross-examination



A COMPUTER

CAN NEVER BE HELD ACCOUNTABLE

THEREFORE A COMPUTER MUST NEVER

MAKE A MANAGEMENT DECISION

- Computer evidence is normally presented by someone who can answer challenges to this evidence
- In practice, this person might be unable to do so effectively
- Expert witness may also help interpret documents

Lessons about electronic evidence come from legal cases that aren’t chosen at random

- The routine use of electronic evidence is challenged when someone brings a legal action, and it’s often necessary to appeal a judgment to make significant changes to how law works
- Drink-driving/driving under the influence often occurs since the people accused may be more able to afford a lawyer than most and have a strong incentive to challenge a conviction
- Problems have been found, e.g.
  - Dräger Alcotest 7110 MKIII-C ignored a low reading in some cases due to a buffer overflow; subsequent testing found thousands of errors
  - Leaked documents on the Lion Intoximeter 3000 noted incorrect results in the presence of acetone

Jonathan Zdziarski found numerous problems with iOS forensics tools

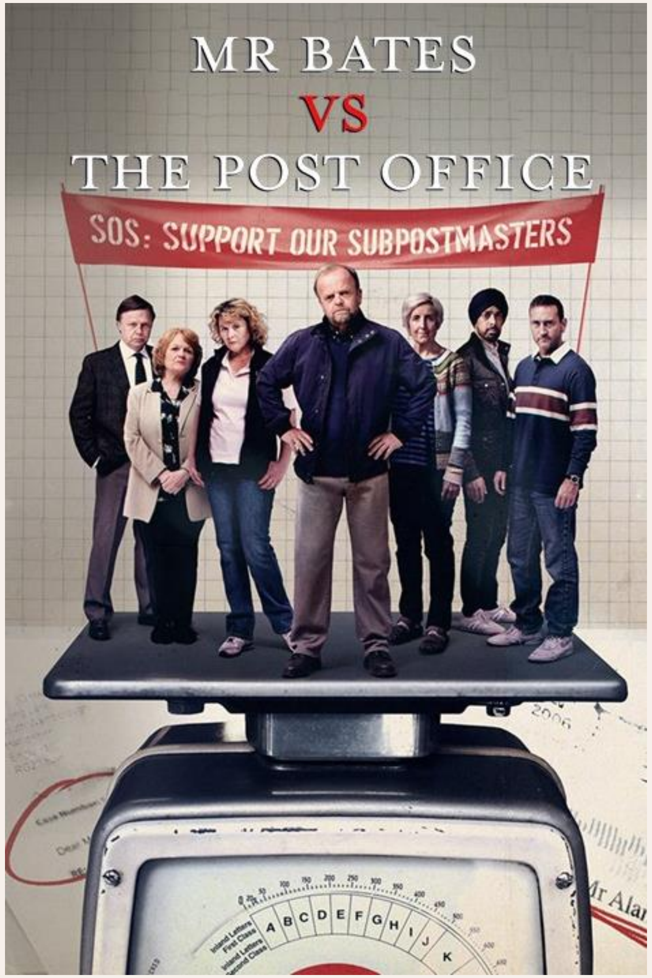
- In the prosecution of Jeffrey Sinclair by the US Army, the timeline of interactions with an iOS device was of critical importance
- Commercial tools had been used to analyse an iPhone 3
- Errors included misreporting times of access and deletions and making up missing data

Horizon Post Office scandal is the largest miscarriage of justice in the UK

- Post Offices in the UK are effectively a bank and handle many financial transactions on behalf of the government
- Horizon is the computer system built by ICL/Fujitsu and provided by the Post Office to manage branches to track stock and funds
- Post Office workers were held personally liable for any losses indicated by Horizon and, in hundreds of cases, were prosecuted

After decades of unsuccessful challenges, the subpostmasters won their case

- Justice for Subpostmasters Alliance showed that Horizon created phantom losses due to various bugs
- Inquiry showed that many problems were known but not disclosed
- The case was incredibly expensive and was only possible because an investment firm paid for the legal costs and was awarded a large proportion of the settlement



Requests to examine software for bugs or otherwise assess validity are often refused

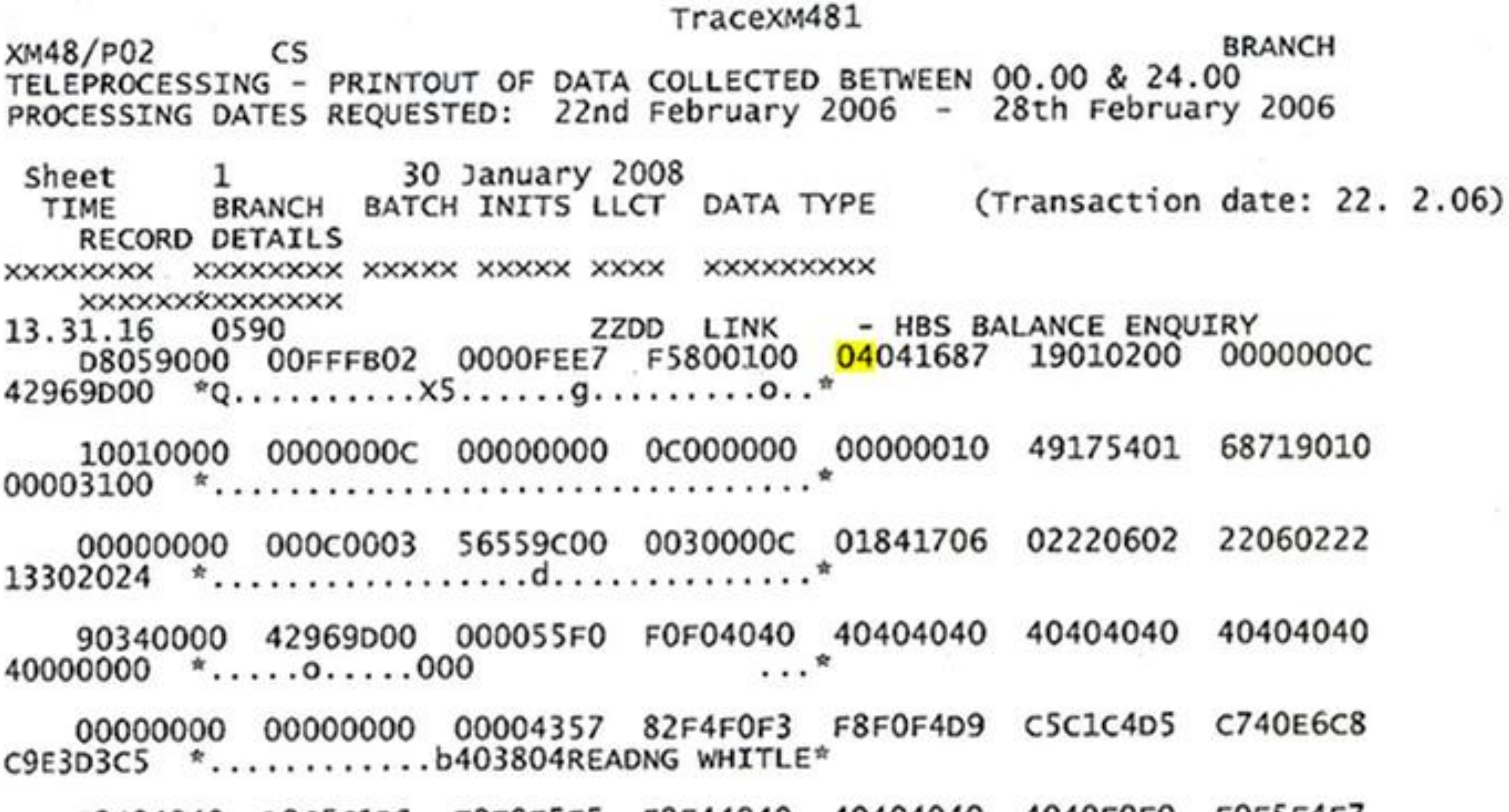
- Before software can be assessed for validity, non-public information must be disclosed to participants
  - Including source code, but also bug reports, audits etc.
- There is often strong resistance for a court to allow this, due to concerns that
  - The intellectual property of software will be infringed
  - Criminals will be able to make use of information disclosed
  - It would be an unjustifiable expense
- In the absence of any successful challenge, in English law, evidence from computers is presumed to be reliable

Improvements to computer systems can address concerns about validating evidence

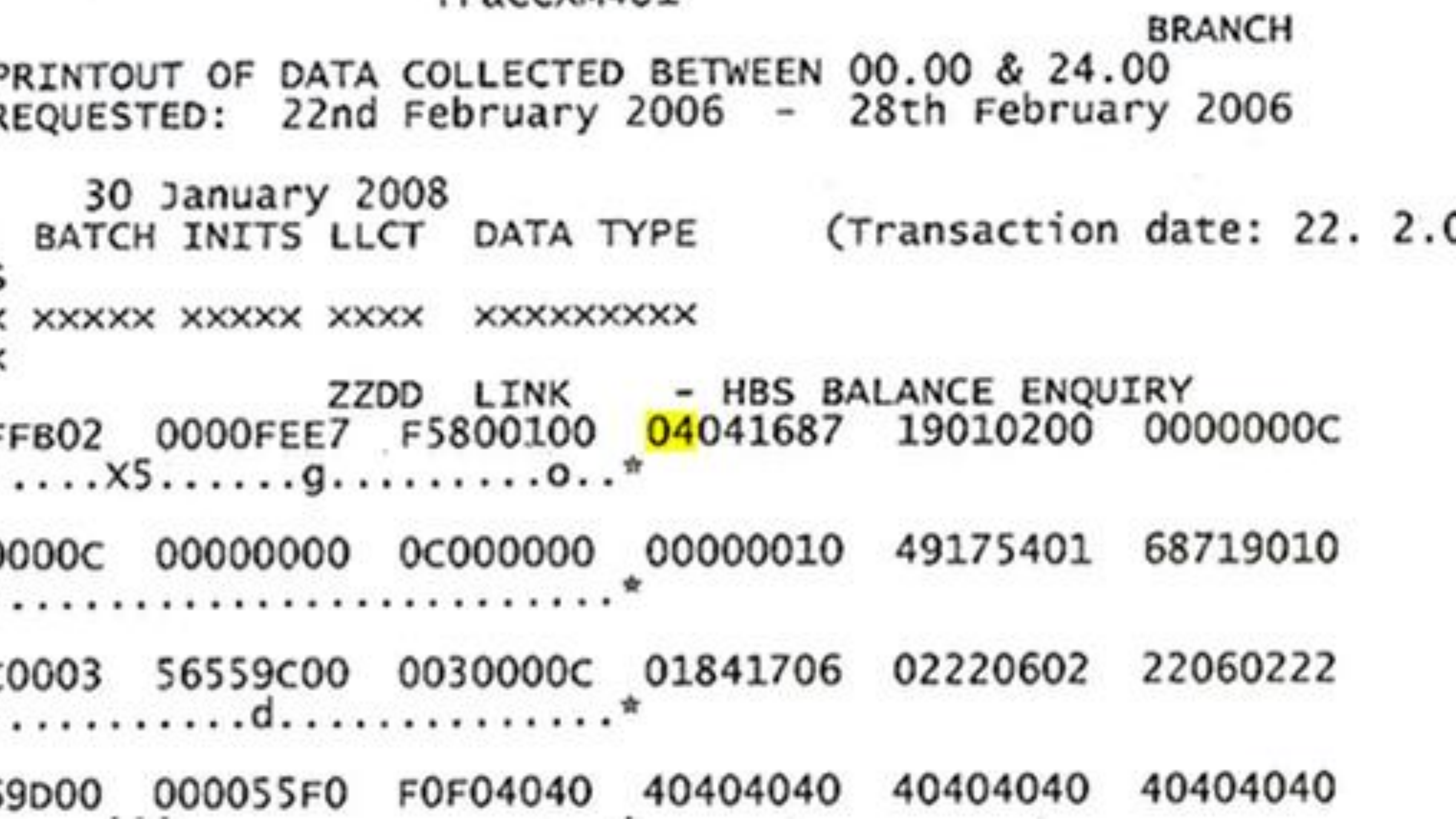
- “The intellectual property of software will be infringed”
  - Design system such that only a small portion of the system needs to be examined to validate evidence and then disclose this
  - This code can also be independently re-implemented
- Criminals will be able to make use of information disclosed
  - Design system with the assumption that it will be disclosed (which is good practice anyway for secure systems)
- It would be an unjustifiable expense
  - Ensure processes to validate evidence are cheap to carry out, well-documented and do not need special skills

EMV/Chip & PIN already has some of the elements discussed, unintentionally

- Cards have a symmetric key shared between bank and the card
- Each transaction has a MAC under this key
- Cards also store a counter of how many transactions are initiated and some even have a log
- Protocol is published, so there are no secrets
- If a transaction is disputed, you can check if MAC is correct, ATC sequence is consistent and (if possible) does card have log entry
- I tried this, with limited success, in Job v. Halifax
- Bank argued it would be unsafe to disclose more details



```
                                   TraceXM481
XM48/P02      CS                                                  BRANCH
TELEPROCESSING - PRINTOUT OF DATA COLLECTED BETWEEN 00.00 & 24.00
PROCESSING DATES REQUESTED:  22nd February 2006  -  28th February 2006

 Sheet      1        30 January 2008
  TIME      BRANCH  BATCH INITS LLCT  DATA TYPE      (Transaction date: 22. 2.06)
   RECORD DETAILS
XXXXXXXX  XXXXXXXX XXXXX XXXXX XXXX  XXXXXXXXX
   XXXXXXXXXXXXXX
13.31.16  0590                  ZZDD  LINK    - HBS BALANCE ENQUIRY
   D8059000  00FFFB02  0000FEE7  F5800100  04041687  19010200  0000000C
42969D00  *Q..........X5......g.........o..*

   10010000  0000000C  00000000  0C000000  00000010  49175401  68719010
00003100  *................................*

   00000000  000C0003  56559C00  0030000C  01841706  02220602  22060222
13302024  *..................d.............*

   90340000  42969D00  000055F0  F0F04040  40404040  40404040  40404040
40000000  *.....o.....000               ...*

   00000000  00000000  00004357  82F4F0F3  F8F0F4D9  C5C1C4D5  C740E6C8
C9E3D3C5  *............b403804READNG WHITLE*
```



```
                                                              BRANCH
PRINTOUT OF DATA COLLECTED BETWEEN 00.00 & 24.00
REQUESTED:  22nd February 2006  -  28th February 2006

    30 January 2008
  BATCH INITS LLCT  DATA TYPE     (Transaction date: 22. 2.0
S
x xxxxx xxxxx xxxx  xxxxxxxxx
x
                ZZDD  LINK     - HBS BALANCE ENQUIRY
FFB02  0000FEE7  F5800100  04041687  19010200  0000000C
 ....X5......g..........o..*

0000C  00000000  0C000000  00000010  49175401  68719010
 .........................*

C0003  56559C00  0030000C  01841706  02220602  22060222
 ..........d..............*

59D00  000055F0  F0F04040  40404040  40404040  40404040
```

If validation is cheap, it can be done as a matter of course and so work when needed

- Communications data is probably the most common type of electronic evidence in criminal cases
  - Who called who, what address is registered to this phone number, where was this phone at this time, etc.
- ETSI TS 103 307 allows telcos to create a database of hashes of evidence packages so that anyone can query entries
  - Standard also recommends regular testing of the interfaces
- Security usually needs to come with some other benefits
  - e.g. SSH caught on because of display forwarding rather than security
  - This system allows telcos to avoid sending someone to court to validate
  - Telco can retain anonymised hashes indefinitely under GDPR

Courts need to be able to tell how reliable computer evidence is

One useful test is whether the company producing it relies upon it for its own purposes:

> *"… the book was at the time of the making of the entry one of the ordinary books of the bank, and that the entry was made in the usual and ordinary course of business, and that the book is in the custody or control of the bank."*
> (Bankers Books Act, 1879)

- The accuracy of logs from Horizon was more than sufficient for managing the business, since errors were rare and **in aggregate**, immaterial to the company accounts, despite being devastating for individuals affected by errors

Improper statistical reasoning about software bugs can lead to incorrect conclusions

- The Post Office expert witness argued (in essence) that it is implausible that subpostmasters suffered the claimed losses
  - Horizon very rarely makes errors (true)
  - The average loss to an individual subpostmaster will be small (true)
  - It is implausible that 550 subpostmaster claimants out of 10,000 would see the large losses reported
- The logical error was to implicitly assume that the 550 subpostmasters were selected at random when they were, in fact, selected specifically because they reported seeing large losses

Evidence from a very reliable computer system isn’t necessarily sufficient

- Horizon failed (for the sake of argument) with 1 in a million probability by falsely accusing someone of fraud
- The prosecutor’s fallacy would be to argue that anyone identified by Horizon as having taken money is guilty with 99.9999% confidence
- Actually, Horizon completes about 6 million transactions per day so there will be about 6 false accusations each day
- To better assess the guilt, evidence can be sought that is independent of the decision to investigate someone for fraud

We fortunately don’t need to apply safety critical systems engineering everywhere

- High assurance engineering techniques work, but are incredibly expensive
- They are used to ensure that a system always works correctly, and acts in a timely manner
- To assure the validity of evidence we only need that a system will not undetectably fail
  - Detectable failures may cause the guilty to go free, but that’s less harmful than convicting the innocent
  - Validation can take time and incur reasonable costs

Bugs can cause a system to change into an incorrect state, but how likely is it to be valid?

- For example, consider a CCTV camera in a shop that shows someone stealing something – how plausible is it that a failure shows a different person or them doing something different
  - If processing is simple, errors in processing will almost certainly be something that is obviously wrong, e.g. corruption, freeze-frame etc.
- Consider an accounting system like Horizon – how plausible is it that a bug will cause an apparent loss
  - Almost any change to a transaction will create a valid transaction but one which loses money for some party
- If space of valid states is sparse, and processing is unaware of validity, then errors are likely detectable

Principles for designing computer systems for creating reliable evidence. Discuss!

- Validation procedures should be publicly disclosed, require a minimum of special knowledge to carry out, and be tested to ensure they work when needed
- Some tests should be kept back and only used as independent validation of a decision to take legal action
- Records that can be used to argue the reliability of a system, such as audit logs and bug tracker information, should be the same that are used internally for maintaining quality
- Design systems for explainability and resist inadvertently flipping between valid states