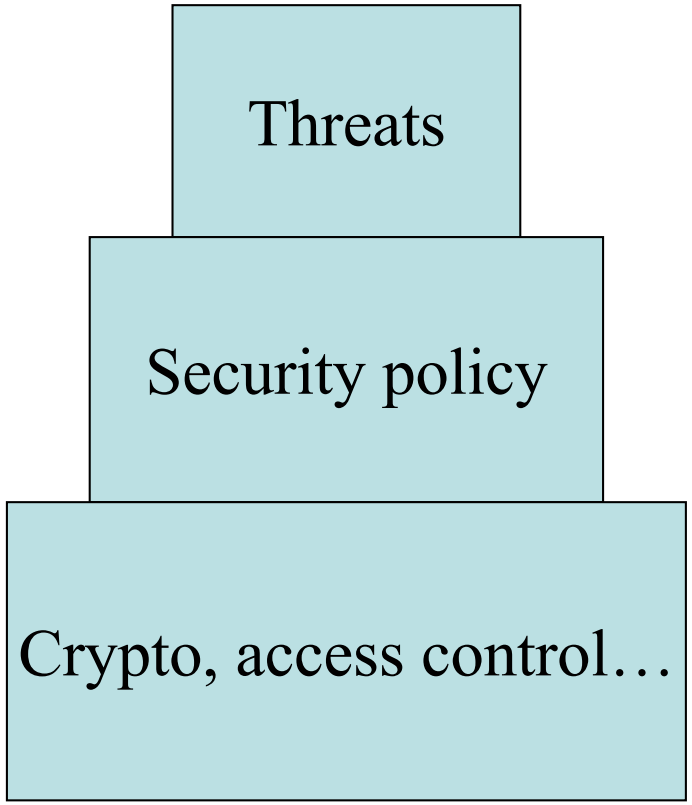
Security Engineering

Threat models and security policies

Design Hierarchy

- What are we trying to stop?
- How are we trying to stop it?
- With what mechanisms?



Terminology

- A *system* can be:
  - a product or component (PC, smartcard,…)
  - some products plus O/S, comms and infrastructure
  - the above plus applications
  - the above plus internal staff
  - the above plus customers / external users
- Common failing: policy drawn too narrowly

Terminology (2)

- A *subject* is a physical person
- A *person* can also be a legal person (firm)
- A principal can be
  - a person
  - equipment (PC, smartcard)
  - a role (the officer of the watch)
  - a complex role (Alice or Bob, Bob deputising for Alice)
- The level of precision is variable – sometimes you need to distinguish 'Bob's smartcard representing Bob who's standing in for Alice' from 'Bob using Alice's card in her absence'. Sometimes you don't

Terminology (3)

- *Secrecy* is a technical term – mechanisms limiting the number of principals who can access information
- *Privacy* means control of your own secrets
- *Confidentiality* is an obligation to protect someone else's secrets
- Thus your medical privacy is protected by your doctors' obligation of confidentiality

Terminology (4)

- *Anonymity* is about restricting access to metadata. It has various flavours, from not being able to identify subjects to not being able to link their actions
- An object's *integrity* lies in its not having been altered since the last authorised modification
- *Authenticity* has two common meanings –
  - an object has integrity plus freshness
  - you're speaking to the right principal

Terminology (5)

- *Trust* is the hard one! It has several meanings:
  1. colloquially, trust is a warm fuzzy feeling
  2. a trusted system or component is one that can break my security policy
  3. a trusted system is one I can insure
  4. a trusted system won't get me fired when it breaks
- I'm going to use number 2 – the defence industry definition. A GCHQ person selling keys to the Russians is trusted but not trustworthy (assuming their action is unauthorized :-)

Terminology (6)

- A *security policy* is a succinct statement of protection goals – typically less than a page of normal language
- A *protection profile* is a detailed statement of protection goals – typically dozens of pages of semi-formal language
- A *security target* is a detailed statement of protection goals applied to a particular system – and may be hundreds of pages of specification for both functionality and testing

What often passes as ‘Policy’

1. This policy is approved by Management.
2. All staff shall obey this security policy.
3. Data shall be available only to those with a ‘need-to-know’.
4. All breaches of this policy shall be reported at once to Security.

What’s wrong with this?

Three security policies

- All assume an insider threat – a disloyal employee, or malware on their laptop
  - In an intelligence agency, tell the opponents or the press what's happening
  - In a health system, look at sensitive personal information such as celebrities' records
  - In a bank, steal money
- In each case, we try to limit the damage

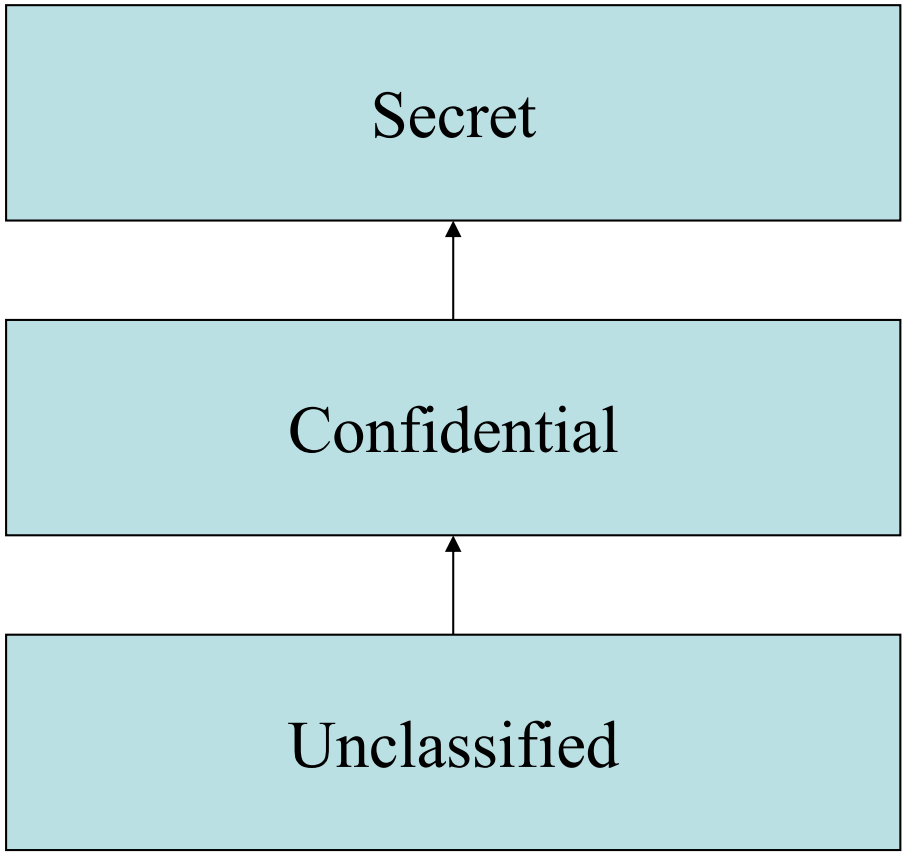
First Policy Example – MLS

- Multilevel Secure (MLS) systems are widely used in government
- Goes back to President Roosevelt, 1940: a clerk with ‘Secret’ clearance can read documents at ‘Confidential’ and ‘Secret’ but not at ‘Top Secret’
- 60s/70s: problems with early mainframes led to Anderson report (1973) for USAF
- Recommendation: with computers, try to keep security policy and enforcement simple

Levels of Information

- Levels include:
  - Top Secret: compromise could cost many lives or do exceptionally grave damage to operations. E.g. intelligence sources and methods
  - Secret: compromise could threaten life directly. E.g. weapon system performance
  - Confidential: compromise could damage operations
  - Official: compromise might embarrass?
- Resources have classifications, people (principals) have clearances. Information flows upwards only

Information Flows



Formalising the Policy

- Initial attempt – WWMCCS – just said that no process could read a resource at a higher level. Not enough!
- Bell-LaPadula (1973):
  - *simple security policy*: no read up
  - **-policy*: no write down
- Theorem: a safe system stays safe
- Ideal: minimize the Trusted Computing Base (set of hardware, software and procedures that can break the security policy) in a *reference monitor*

Objections to Bell-LaPadula

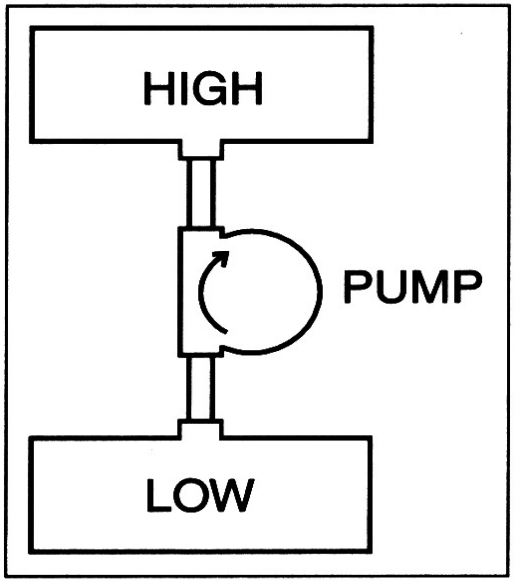
- Processes such as memory management, need to read and write at all levels
- Fix: put them in the trusted computing base
- But: once you put in all the stuff a real system needs (backup, recovery, comms, …) the TCB is too big to be easily verifiable
- And what about apps like license servers?

Covert Channels

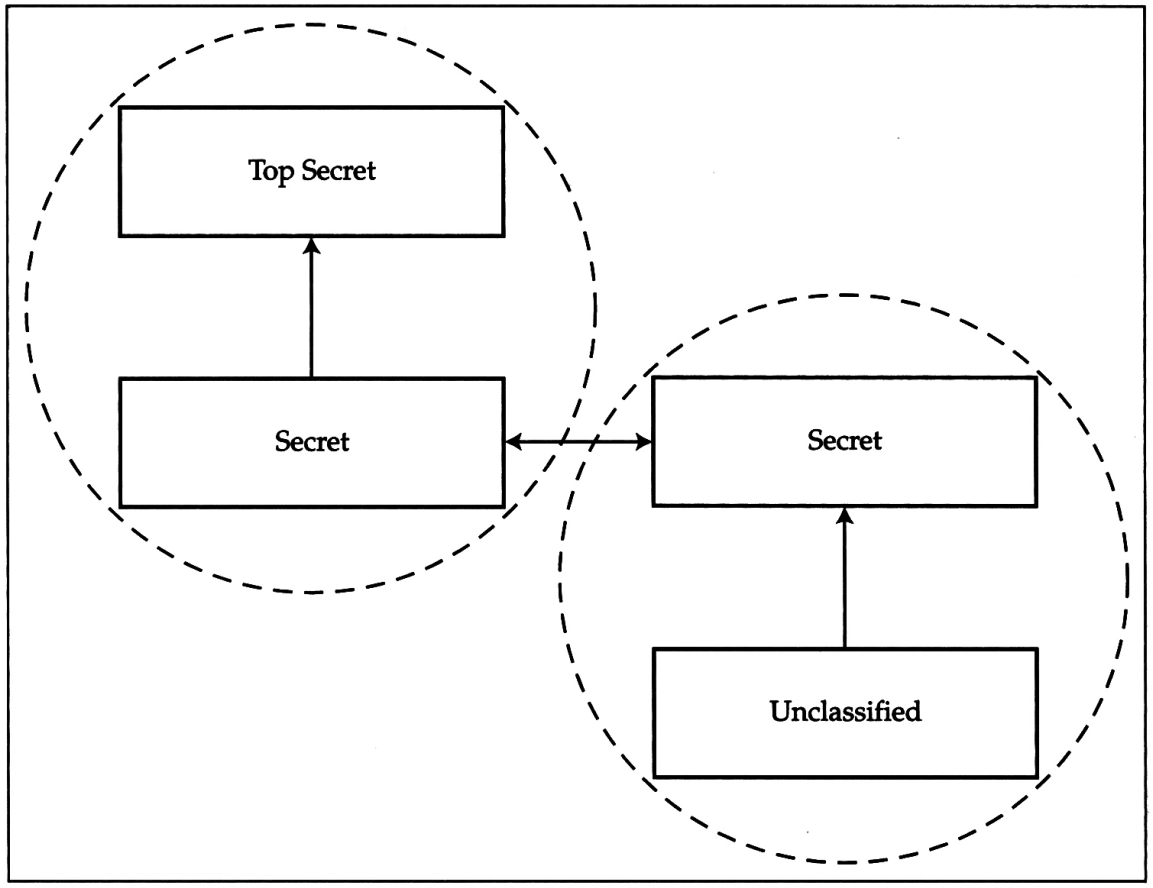
- In 1973 Butler Lampson warned BLP might be impractical because of covert channels: “neither designed not intended to carry information at all”
- A Trojan at High signals to a buddy at Low by modulating a shared system resource
  - Fills the disk (storage channel)
  - Loads the CPU (timing channel)
- Capacity depends on bandwidth and S/N. So: cut the bandwidth or increase the noise
- More on covert channels and side channels later…

Example MLS System

- Pumps, also known as data diodes, copy data continuously up from Low to High with minimal covert channel leakage



Composability



Consistency

- US approach (cover stories):

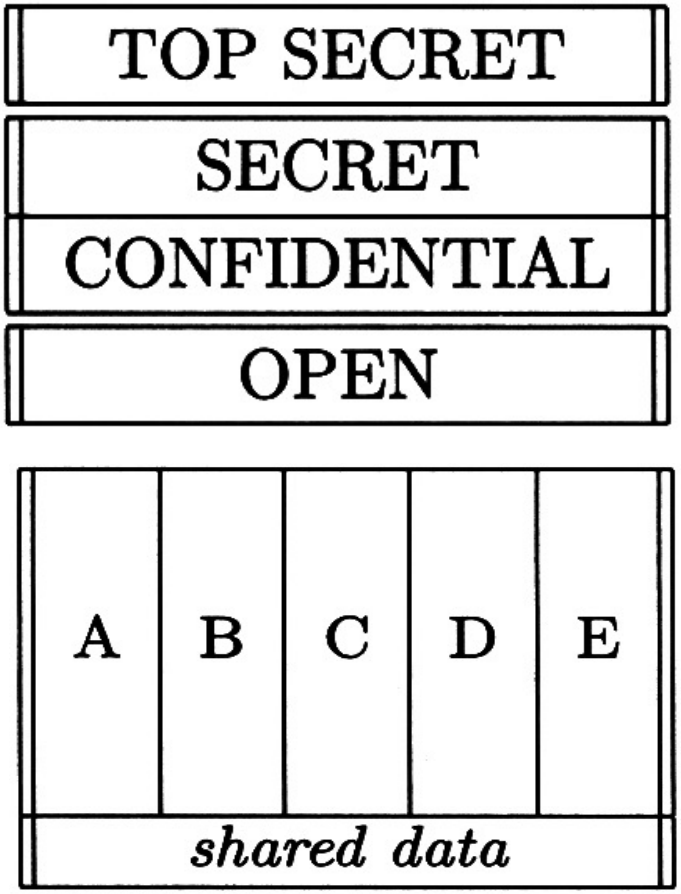
	Cargo	Destination
Secret	Missiles	Iran
Unclassified	Spares	Cyprus

- UK approach (don't tell low users):

	Cargo	Destination
Secret	Missiles	Iran
Restricted	Classified	Classified

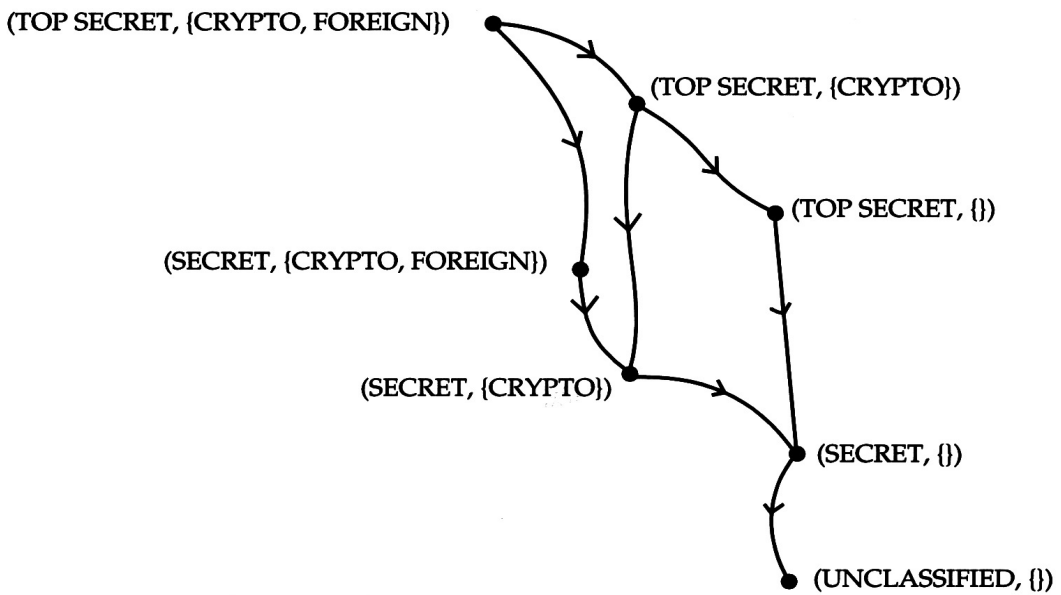
Multilateral Security

- Sometimes the aim is to stop data flowing down
- Other times, you want to stop lateral flows
- Examples:
  - Intelligence
  - Competing clients of an accounting firm
  - Medical records by practice or hospital



The Lattice Model

- This is how intelligence agencies manage ‘compartmented’ data – by adding labels
- Basic idea: BLP requires only a partial order



What didn’t work so well

- 1996: medical records in 11,000 surgeries
- 2021: now on three cloud services
- Idea: access by role and relationship
- How this failed at the coalface
- Repeated opt-out games
- Repeated games around ‘anonymization’
- The OpenSafely Covid project

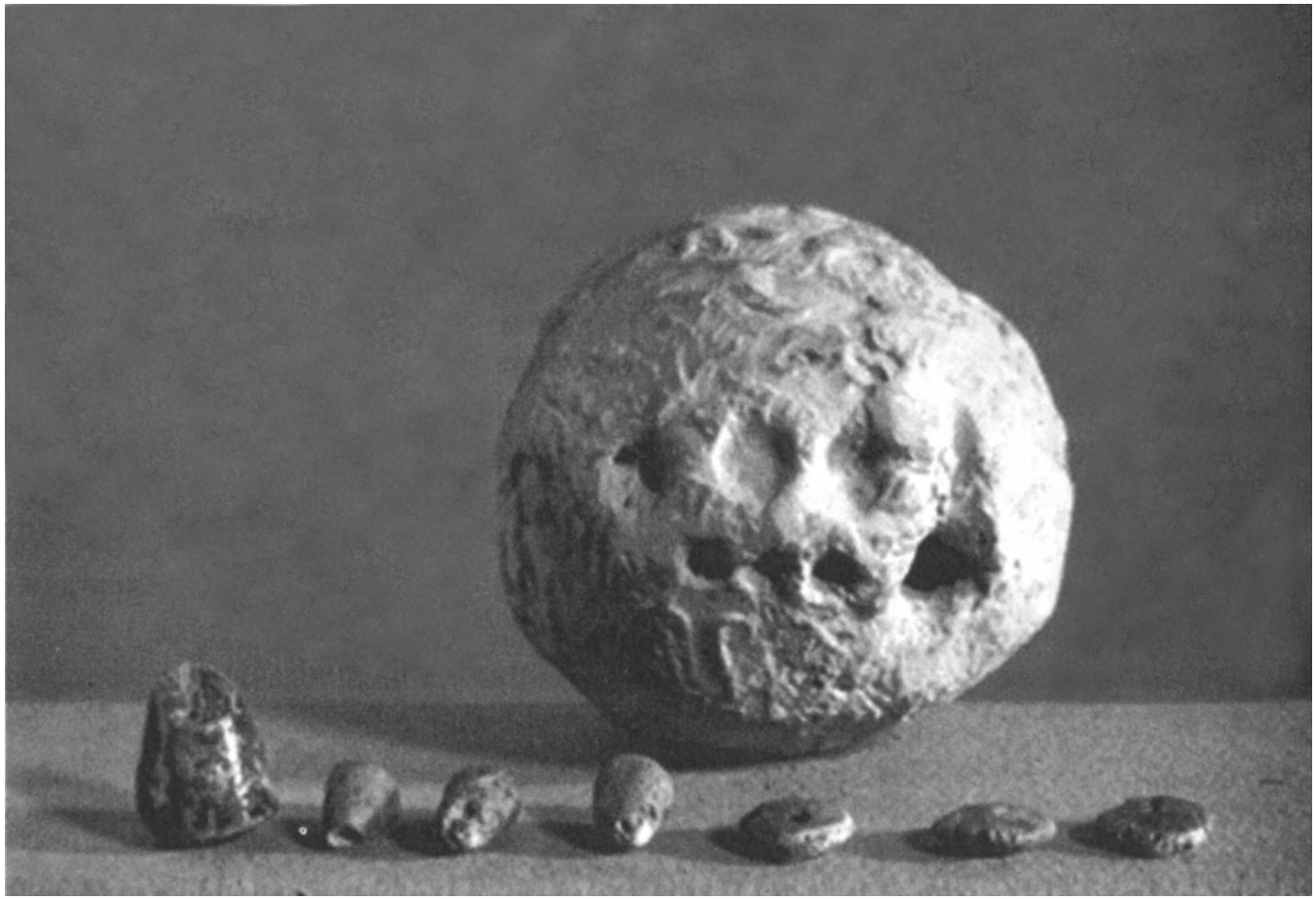
Alternative lateral flow controls

- Chinese Wall Model – in an investment bank or accountancy firm, if you've worked for an oil company, you can't work for a competing oil company for (e.g.) two years
- Delegation – in a retail bank, you only get to see a customer's account details once they've passed authentication
- Honeypots

Multilevel Integrity

- The Biba model – data may flow only down from high-integrity to low-integrity
- Dual of BLP!
- Example 1: medical device calibration / operation
- Example 2: electricity / gas / oil distribution
  - Safety: highest integrity level
  - Monitoring and control: next level
  - Enterprise apps (e.g. billing): third level
- Colonial hack: operator turned off the pipeline when ransomware killed the billing system!

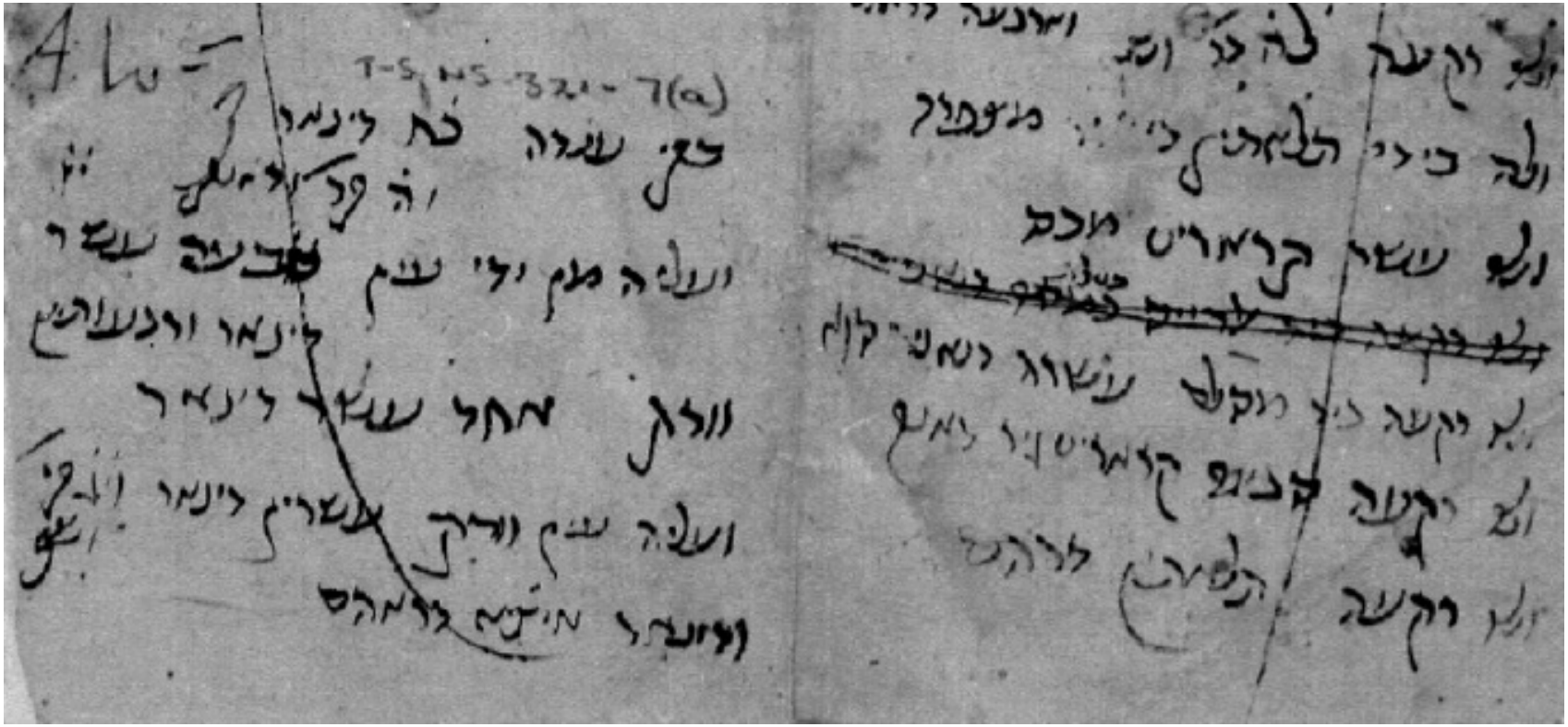
Bookkeeping, c. 3300 BC



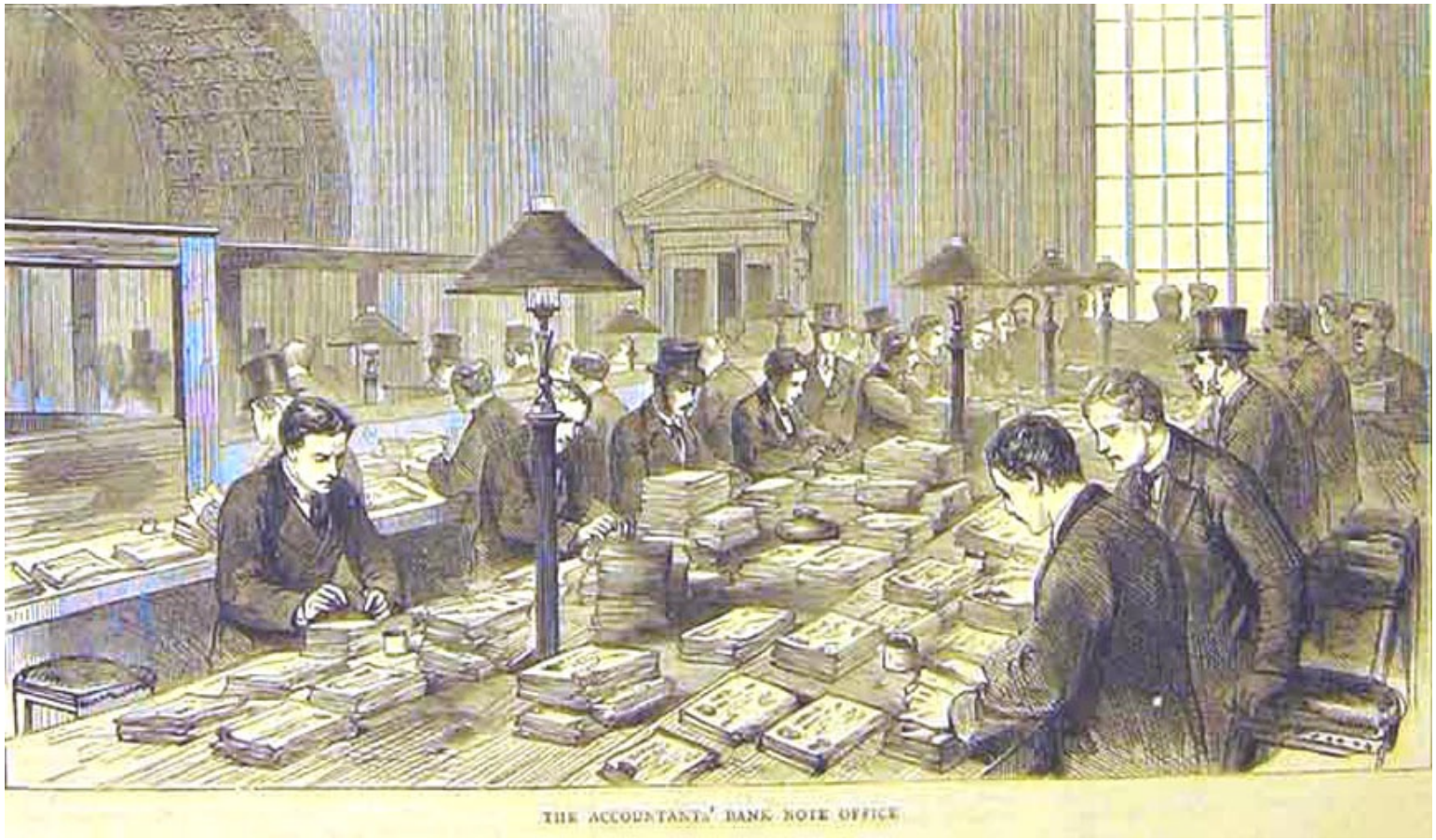
Bookkeeping c. 1100 AD

- How do you manage a business that's become too large to staff with your own family members?
- Double-entry bookkeeping – each entry in one ledger is matched by opposite entries in another
  - E.g. firm sells £100 of goods on credit – credit the sales account, debit the receivables account
  - Customer pays – credit the receivables account, debit the cash account
- Idea: staff must collude to get away with fraud

From the Genizah Collection



Bank of England, 1870



Banking Security Policy

- Threat model:
  - 1% of staff go bad each year
  - Mistakes happen – 1 in 500 paper transactions
  - There are clever fraudsters too
  - Loss of confidence can mean ruin
- Protection goals:
  - Deter/prevent the obvious frauds
  - Detect the rest as soon as possible
  - Be able to defend the bank's actions in court

The Clark-Wilson Policy Model

- Work by David Clark (MIT) and David Wilson (accountant) in 1986 to model real bookkeeping systems
- In addition to the normal objects in your system, which we call unconstrained data items (UDIs), you add constrained data items (CDIs)
- CDIs are acted on by special programs called transformation procedures (TPs)
- Mental model: a TP in a bank must increase the balance in one CDI (account) by the same amount that it decrements another

Clark-Wilson Framework

- There's an IVP to validate CDI integrity
- Applying a TP to a CDI maintains integrity
- A CDI can only be changed by a TP
- Subjects can use only certain TPs on certain CDIs
- Triples (subject, TP, CDI) enforce separation of duty
- Certain TPs act on UDIs to produce CDI output
- Each application of a TP writes enough information to an audit-trail CDI to reconstruct its action
- The system authenticates subjects initiating a TP
- Only special subjects (security officers) can set up and alter triples

Actual Bookkeeping Systems

- How do you do separation of duties?
- Serial:
  - Lecturer gets money from EPSRC, charity, …
  - Lecturer gets finance office to register supplier
  - Gets stores to sign order form and send to supplier
  - Stores receives goods; department gets invoice
  - Department checks delivery and tell finance to pay
  - Lecturer gets statement of money left on grant
  - Audit by grant giver, university, …
- Parallel: two signatures (e.g. where transaction large, irreversible, as in bank guarantee)

Internal Control Theory

- Employees optimise their own utility, not their employers' (the 'agency problem')
- Internal controls should mitigate not just fraud but nepotism, empire-building, …
- Corporate governance rules like Sarbanes-Oxley (USA), Cadbury (UK) set the tone
- The big accountants drive 'good practice'
- People talk of 'risk management' but the process is basically evolutionary

Internal Control Practice

- Balancing the books isn't enough! McKesson and Robbins collapse, 1938, had fictitious trading partners and a bogus Montreal bank
- Wirecard was much the same again, in 2020!
- Enforcement is often cyclical, politicised
- Systematic analysis: trace worst outcomes back along workflow, and also look for greatest opportunities for individual staff (ask them!)
- Deter – prevent – detect – alarm – delay – response

Lessons learned

- No single solution to the insider threat!
- Multilevel security policies were the first to be explored, thanks to the military
- Used for safety/integrity as well as secrecy
- Multilateral policies mitigate effects of scale
- Integrity policies drove commercial IT security, via bookkeeping
- Learn from real-world examples!