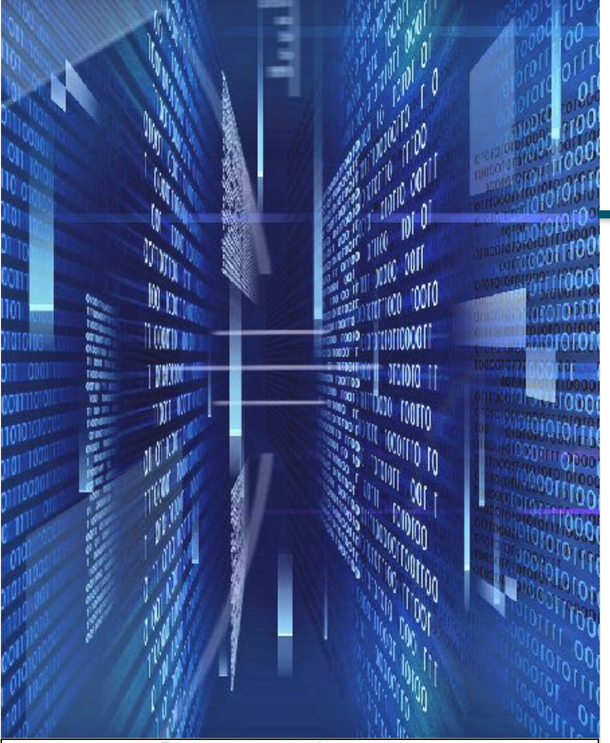

cybersecurity centre
GCHQ Academic Centre of Excellence EPSRC

UNIVERSITY OF Southampton

comp6224 (2016)
week 6: multilevel security

Vladimiro Sassone

Cyber Security Centre

University of Southampton

Cybersecurity Southampton

CyberSecuritySoton.org [w]

@CybSecSoton [fb & tw]

multilevel security or mandatory access control (MAC)

military database systems hold information at different levels of classification (*Confidential, Secret, Top Secret*, . . .) and ensure that data can only be read by a principal whose level is at least as high as the data's classification.

These are interesting because of:

- huge amount of research
- used in several commercial systems, eg telecomms, banks
- integrated in some operating systems, eg Vista, SELinux
- but sometimes they are used where they shouldn't

security policies

typical form of top-down approach to security engineering

threat model — security policy — security mechanisms.

security policy is a document that expresses clearly and concisely what the protection mechanisms are to achieve.

typically it says: which user may access which data.

Example:

Megacorp Inc security 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.

Figure 8.1: A typical corporate information security policy

this is wrong in many ways; mostly, it does not really mean anything

a refined nomenclature

A *security policy model* is a succinct statement of the protection properties which a system, or generic type of system, must have.

A *security target* is a more detailed description of the protection mechanisms that a specific implementation provides, and how they relate to a list of control objectives.

A *protection profile* is like a security target but expressed in an implementation-independent way to enable comparable evaluations across products and versions.

we will discuss some very basic security policies and the mechanism used to meet them

Bell-LaPadula security model policy

It all started in the late ’60s, when it was discovered that the Pentagon’s World Wide Military Command and Control System was vulnerable to Trojan Horse attacks.

A review suggested that secure system should do just one or two things well; and that protection properties should be enforced by mechanisms simple enough to verify and changed only rarely.

Implemented via a *reference monitor* — a component of the OS to mediate each access control decision. It should be small enough to be subject to analysis and tests, the completeness of which could be assured.

In modern parlance, it would form *The Trusted Computing Base* (TCB).

But what are these core security properties that should be enforced above all others?

multilevel classification systems

After WWII, NATO governments moved to a common protective marking scheme for labelling the sensitivity of documents: “labels” which run upwards from *Unclassified* through *Confidential*, *Secret* and *Top Secret*

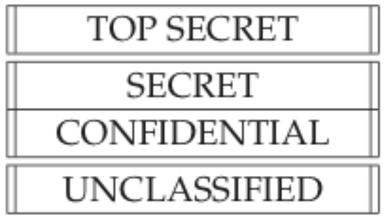

Figure 8.2: Multilevel security

simple policy: an official can read a document only if his clearance is at least as high as the document’s classification.

The effect is that information may only flow upwards, from confidential to secret to top secret; it may never flow downwards unless an authorised person takes a deliberate decision to declassify it.

Very general: eg wiretapping requires an information flow policy where the ‘High’ principal can see ‘Low’ data, but a ‘Low’ principal can’t tell whether ‘High’ is reading any data at all, let alone what data.

In this context of classification the *Bell-LaPadula* model of computer security was born (1973).

It proposes two properties:

- The *simple security property*: no process may read data at a higher level. This is also known as *no read up (NRU)*;
- The **-property*: no process may write data to a lower level. This is also known as no write down (NWD).

The *-property was Bell and LaPadula's critical innovation.

This was to be enforced independently of user actions, called *mandatory access control*, as opposed to the *discretionary access control,* eg in systems like Unix where users can take their own access decisions about their files.

information flow control (ctd)

Re MAC there was a pre-existing culture that security policy was enforced independently of user actions; the move to computers didn't change this.

BLP clarifies it: the security policy must be enforced not just independently of users' direct actions, but of their indirect actions (such as the actions taken by programs they run).

So we must prevent programs running at 'Secret' from writing to files at 'Unclassified', or more generally prevent any process at High from signalling to any object (or subject) at Low.

implementing multi level security

Multilevel security can be implemented in a number of ways.

original idea: build a *reference monitor* by beefing up the part of an operating system which supervises all operating system calls and checks permissions to decide whether the call can be serviced or not.

Yet, as it turns out, it is very difficult to contain the size of this.

Replicate systems. Often physical in the 1990s, and since about 2005 uses virtual machines.

some promising recent work builds on virtualisation products such as VMware and Xen to provide multiple systems at different security levels on the same PC.

criticism of BLP (i)

BLP caused a lot of excitement: a straightforward security policy which was intuitive yet still allowed people to prove theorems.

But it was soon showed that the BLP rules were not in themselves enough. In practice you always need to allow for circumstances where data must be written down.

System Z, defined as a BLP system with the added feature that a user can ask the system administrator to temporarily declassify any file from High to Low. In this way, Low users can read any High file without breaking the BLP assumptions.

Yet, he criticism is that System Z cheats by doing something the model doesn't allow. This, in itself, is a security threat.

criticism of BLP (ii)

The issue is dealt by adding a *tranquility property*.

- **strong tranquility**: security labels never change during system operation;
- **weak tranquility**: labels never change so as to violate a defined security policy

the weak property is that in a real system we want the *principle of least privilege*

> start off a process at the uncleared level, even if the owner of the process were cleared to 'Top Secret'. If she then accesses a confidential email, her session is automatically upgraded to 'Confidential'; and in general, her process is upgraded each time it accesses data at a higher level

this is known as the *high water mark* principle,

According to this, o a process which has read files at 'Secret' and 'Crypto' will thereafter create files marked (at least) 'Secret Crypto'.

If it then reads a file at 'Top Secret Daffodil' then all files it creates after that will be labelled 'Top Secret Crypto Daffodil', and it will not be able to write anymore to any files at 'Secret Crypto'.

criticism of BLP (iii)

biggest problem yet: most application software must be rewritten to work with this.

And you can easily find that everything ends up in the highest compartment, or that the system fragments into thousands of tiny compartments that don't communicate at all with each other.

separating users and processes is relatively straightforward; the hard part is when some controlled interaction is needed.

Most real applications need some kind of 'trusted subject' that can break the security policy

eg a trusted word processor that helps an intelligence analyst scrub a Top Secret document when she's editing it down to Secret.

role-based access control

Role-based access control (RBAC)

a more general framework for MAC in which access decisions depend not on users' names but on the functions they are currently performing within the organisation. It consists of the

- specification of what transactions may be performed by holders of a given role,
- the mechanisms for granting membership of a role, and delegate it.

Roles, or groups, had for years been the mechanism used in practice in organisations such as banks to manage access control; the RBAC model formalises this.

Their strength is to able to give different access rights to 'X as Professor', 'X as member of the Cyber Security Research Group' and 'X reading private email'.

So, RBAC can usefully deal with integrity as well as confidentiality, by enforcing role changes when certain programs are invoked: e.g., a process calling untrusted software might lose the role membership required to write to sensitive system files.

key observation: confidentiality and integrity are dual concepts

- confidentiality is a constraint on who can read a message,
- integrity is a constraint on who can write or alter it.

Eg an electronic medical device such as an ECG may have two separate modes: calibration and use. Calibration data must be protected from corruption by normal users, who will therefore be able to read it but not write to it;

To model this we can use a multilevel integrity policy:

can read data at higher levels (user process can read the calibration data) and write to lower levels (a calibration process can write to a buffer in a user process);

but we must never read down or write up, as either could allow High integrity objects to become contaminated with potentially unreliable Low data.

The Biba integrity model is often formulated in terms of the *low water mark* principle:

the integrity of an object is the lowest level of the objects used in its creation.

information flow security is not compositional

In isolation, this device is provably secure.

However, if feedback is permitted, then the output from H_3 can be fed back into H_2.

The result that the high input H_1 now appears at the low output L.

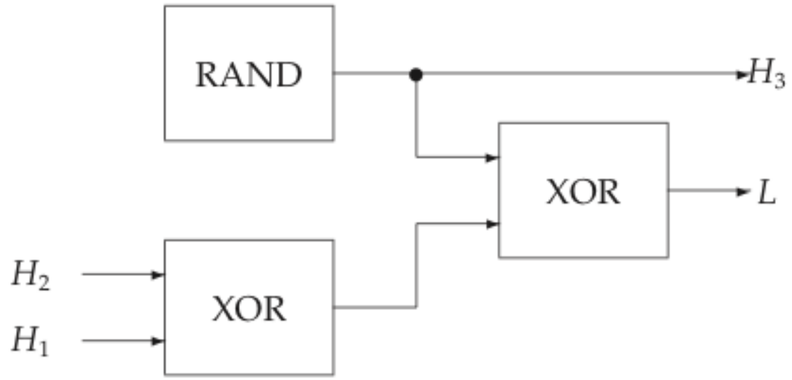

Figure 8.4: Insecure composition of secure systems with feedback

Timing inconsistencies can also lead to the composition of two secure systems being insecure.

Simple information flow doesn't compose; neither do their most diffused variations.

the cascade problem

According to the “Orange Book” security classification, a system evaluated to level B3 is only allowed to process information:

either at Unclassified, Confidential and Secret;

or at Confidential, Secret and Top Secret.

No system is permitted to process Unclassified and Top Secret data simultaneously.

Yet, this security policy is straightforward to subvert:

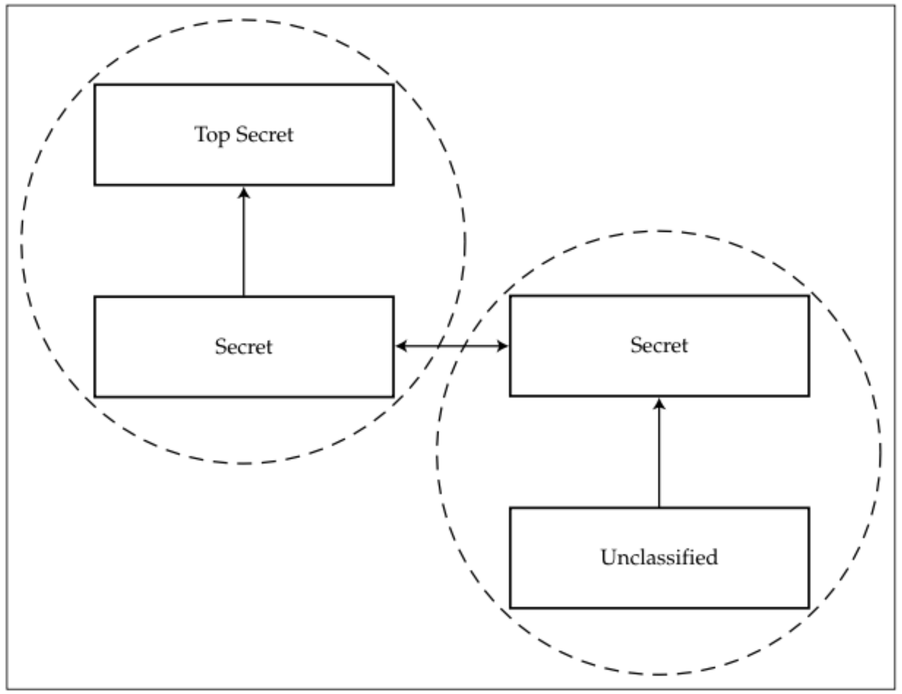

Figure 8.5: The cascade problem

covert channels (Lampson 1973)

A mechanism that was not designed for communication but can be abused to communicate information down from High to Low.

- A typical covert channel arises when a high process can signal to a low one by affecting some shared resource.
- Eg, it could position the disk head at the outside of the drive at time t_i to signal that the *i*th bit in a Top Secret file was a 1, or at the inside to signal that the bit was a 0.
- If a machine is shared between high and low, and resources are not allocated in fixed slices, then the High can signal down eg by filling up the disk drive, or using a lot of CPU or bus cycles, or through process IDs, shared files locks, last access time of files…
- Sometimes classified in *storage channels* and *timing channels.*

All systems with shared resources must find a balance between covert channel capacity, resource utilisation, and fairness.

covert channels (ii)

Counter strategies can enormously reduce the available bandwidth.

It is possible to limit the covert channel capacity by introducing noise. Some machines have had randomised system clocks for this purpose. But some covert channel capacity almost always remains.

The best that developers have been able to do consistently with BLP confidentiality protection in regular time-sharing operating systems is to limit it to 1 bit per second or so

This is considered good for most applications, but it is inadequate if we want to prevent the leakage of a cryptographic key.

- This is one of the reasons for the military doctrine of doing crypto in special purpose hardware rather than in software.

Suppose that our High user has created a file named agents, and our Low user now tries to do the same. If the MLS operating system prohibits him, it will have leaked information — namely that there is a file called agents at High.

Level	Cargo	Destination
Secret	Missiles	Iran
Restricted	—	—
Unclassified	Engine spares	Cyprus

Figure 8.6: How the USA deals with classified data

Level	Cargo	Destination
Secret	Missiles	Iran
Restricted	Classified	Classified
Unclassified	—	—

Figure 8.7: How the UK deals with classified data

It can be fixed for files, but the problem remains a hard one for databases.

Suppose that a High user allocates a classified cargo to a ship. The Low user might think the ship is empty, and try to allocate it another cargo or even to change its destination.

multilateral security

Often our goal is not to prevent information flowing ‘down’ a hierarchy but flowing ‘across’ between departments. This includes most applications where privacy is at stake.

Figure 9.1: Multilevel security

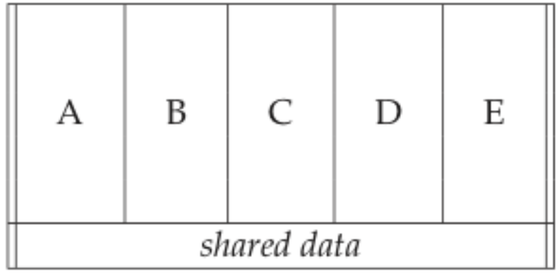

Figure 9.2: Multilateral security

The basic problem is that centralising systems with sensitive information, you create a more valuable asset and simultaneously give more people access to it

.

In the old days, a private investigator who wanted copies of your bank statements had to subvert someone at the branch where your account was kept. Now…

compartmentation and the lattice model

Access control groups can be dealt with using a variant of BLP, called the *lattice model*.

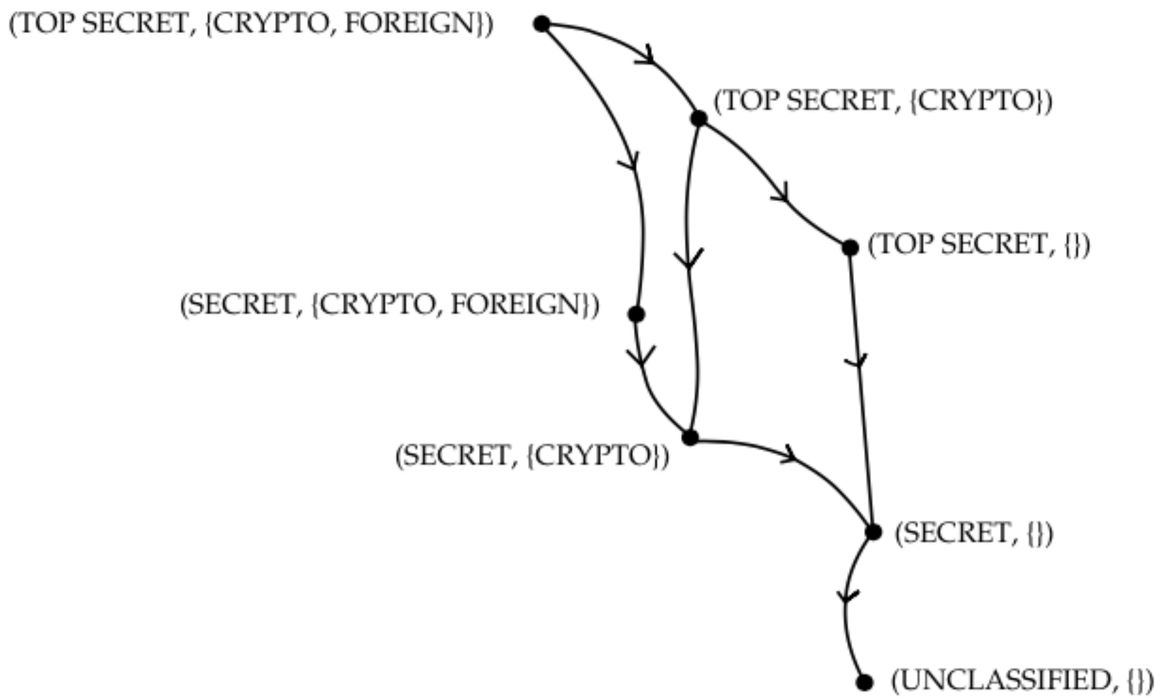

Figure 9.3: A lattice of security labels

compartmentation and the lattice model

Most products built for the multilevel secure market can be reused in compartmented mode. But in practice these are not very effective.

It is easy to use a multilevel operating system to keep data in different compartments separate — just give them incompatible labels (‘Secret Tulip’, ‘Secret Daffodil’, ‘Secret Crocus’, ...).

But the operating system has now become an isolation mechanism, rather than a sharing mechanism; the real problem is how to control information sharing.

In fact, one of the biggest problem facing the U.S. intelligence community since 9/11 is how to handle search over systems with many compartments. Just think about querying a system with that many labels and compartmentalisation…

chinese walls

financial services firms have internal rules designed to prevent conflicts of interest, called Chinese Walls:

an employee who has worked recently for one company in a business sector may not see the papers of any other company in that sector.

writing for object *c: y*(*c*) = company of *c* and *x*(*c*) = conflict-of-interest class of *c*, a chinese wall policy can be expressed as:

- The *simple security property*: a subject s has access to c if and only if, for all c' which s can read, either $y(c) \notin x(c')$ or $y(c) = y(c')$
- The **-property*: a subject s can write to c only if s cannot read any c' with $x(c') \neq \oslash$ and $y(c) \neq y(c')$.

separation of duty: a user may perform transaction A or B, but not both.

Interesting new questions about covert channels:

could a company find out whether a competitor which used the same investment bank is planning a bid for a third company, by asking which specialists were available for consultation and noticing that their number had dropped suddenly?

the british medical association model

Perhaps the most important, interesting and instructive example of MLS so far.

A security policy in which decisions are taken not by a central authority (as in BLP) or by the system's users (as in DAC) but by the data subjects.

The main threat to medical privacy is abuse of authorised access by insiders, and the most common threat vector is social engineering. The Policy:

1. **Access control:** each identifiable clinical record shall be marked with an access control list naming the people or groups of people who may read it and append data to it. The system shall prevent anyone not on the access control list from accessing the record in any way.
2. **Record opening:** a clinician may open a record with herself and the patient on the access control list. Where a patient has been referred, she may open a record with herself, the patient and the referring clinician(s) on the access control list.
3. **Control:** One of the clinicians on the access control list must be marked as being responsible. Only she may alter the access control list, and she may only add other health care professionals to it.
4. **Consent and notification:** the responsible clinician must notify the patient of the names on his record's access control list when it is opened, of all subsequent additions, and whenever responsibility is transferred. His consent must also be obtained, except in emergency or in the case of statutory exemptions.

the british medical association model

5. **Persistence:** no-one shall have the ability to delete clinical information until the appropriate time period has expired.
6. **Attribution:** all accesses to clinical records shall be marked on the record with the subject's name, as well as the date and time. An audit trail must also be kept of all deletions.
7. **Information flow:** Information derived from record A may be appended to record B if and only if B's access control list is contained in A's.
8. **Aggregation control:** there shall be effective measures to prevent the aggregation of personal health information. In particular, patients must receive special notification if any person whom it is proposed to add to their access control list already has access to personal health information on a large number of people.
9. **Trusted computing base:** computer systems that handle personal health information shall have a subsystem that enforces the above principles in an effective way. Its effectiveness shall be subject to evaluation by inde- pendent experts.

inference control for privacy

Inference control refers to the problem of avoiding secondary leaks of information eg through databases for research, cost control and clinical audit.

To avoid this, U.S. Healthcare Finance Administration (HCFA), maintains three sets of records:

1. There are complete records, used for billing;
2. There are *beneficiary-encrypted* records, with only patients' names and social security numbers obscured. These are still personal data and only usable by trusted researchers;
3. there are *public-access* records.

The latter are stripped of identifiers down to the level where patients are only identified generically 'a white female aged 70–74 living in Vermont'.

Nonetheless, researchers have found that many patients can still be identified by cross-correlating the public access records with commercial databases.

inference control

In fact, anonymisation is much more fragile than it seems; and when it fails, companies and individuals can suffer serious consequences.

- OL faced a storm of protest in 2006 when it released the supposedly anonymous records of 20 million search queries; searchers' names and IPs replaced with numbers, but…
- The DVD rental firm Netflix, deanonymised users from their film preferences.

Two broad approaches to the problem:

1. data are de-identified before processing
2. data are de-identified during processing

Eg1: one record in a thousand made available, minus names, exact addresses and other sensitive data; noise added to the data in order to prevent people with some extra knowledge from tracing individuals; local averages were also given for people selected by various attributes; records with extreme values — such as very high incomes — were suppressed.

Eg2: identifiable data are retained in a database, and privacy protection comes from controlling the kind of queries that may be made. This opens interesting questions.

Some terminology

1. A *characteristic formula* is the expression (in some database query language) that selects a specific set, known as the *query set*, of records.
2. The smallest query sets, obtained by the logical AND of all the attributes (or their negations) are known as *elementary sets* or *cells*.
3. The statistics corresponding to query sets may be *sensitive statistics*

The objective of inference control is to prevent the disclosure of sensitive statistics.

If D is the set of statistics disclosed and P those which are sensitive

we need $D \subseteq \mathbf{Comp}(P)$ for privacy the complement of P.

if $D = \mathbf{Comp}(P)$ then the protection is said to be *precise*.

The simplest protection mechanism is to specify a minimum query size:

query set size control with threshold *t* means that between the size of the answer must be between *t* and *N* – *t* for a query to be allowed.

individual tracker is a formula that allows us to calculate the answer to a forbidden query indirectly.

For instance, if there is only one female professor, this is an individual tracker

'Average salary professors?' and 'Average salary male professors?'.

general trackers are sets of formulae which reveal any sensitive statistic.

It has been proved that if the minimum query set size *n* is less than a quarter of the total number of statistics *N*, and there are no further restrictions on the type of queries that are allowed, then we can find formulae that provide general trackers.

So tracker attacks are easy, unless we place severe restrictions on the query set size or control the allowed queries in some other way.

More problems: how to deal with the side-effects of suppressing certain statistics?

in the example below ur minimum query set size is 3 (less and either of the two who studied ‘geology-with-chemistry’ knows the other’s mark).

Then we cannot release the average mark for ‘geology-with-chemistry’.

But if the average mark for chemistry is known, then this mark can easily be reconstructed from the averages for ‘biology- with-chemistry’ and ‘physics-with-chemistry’.

So, we end up with a lot of data cancelled…

Major:	Biology	Physics	Chemistry	Geology
Minor:				
Biology	–	16	17	11
Physics	7	–	32	18
Chemistry	33	41	–	2
Geology	9	13	6	–

Figure 9.4: Table containing data before cell suppression

More problems: how to deal with the side-effects of suppressing certain statistics?

in the example below ur minimum query set size is 3 (less and either of the two who studied ‘geology-with-chemistry’ knows the other’s mark).

Then we cannot release the average mark for ‘geology-with-chemistry’.

But if the average mark for chemistry is known, then this mark can easily be reconstructed from the averages for ‘biology- with-chemistry’ and ‘physics-with-chemistry’.

So, we end up with a lot of data cancelled…

Major:	Biology	Physics	Chemistry	Geology
Minor:				
Biology	–	16	17	11
Physics	7	–	32	18
Chemistry	33	41	–	2
Geology	9	13	6	–

Figure 9.4: Table containing data before cell suppression

Major:	Biology	Physics	Chemistry	Geology
Minor:				
Biology	–	blanked	17	blanked
Physics	7	–	32	18
Chemistry	33	blanked	–	blanked
Geology	9	13	6	–

Figure 9.5: Table after cell suppression

In general, we can compute that:

a database scheme contains *m*-tuples, blanking a single cell generally means suppressing $2^m - 1$ other cells, arranged in a hypercube with the sensitive statistic at one vertex.

So even precise protection can rapidly make the database unusable.

Where the database is open for online queries, we can get a similar effect by *implied queries control*:

we allow a query on *m* attribute values only if all of the 2^m implied query sets given by setting the *m* attributes to true or false, have at least *k* records.

max order control and the lattice method

Maximum order control limits the number of attributes any query can have.

to be effective, the limit may have to be severe. One study found that of 1000 medical records, three attributes were safe while with four attributes, one individual record could be found and with 10 attributes most records could be isolated.

lattices may be used to analyse query controls in some databases

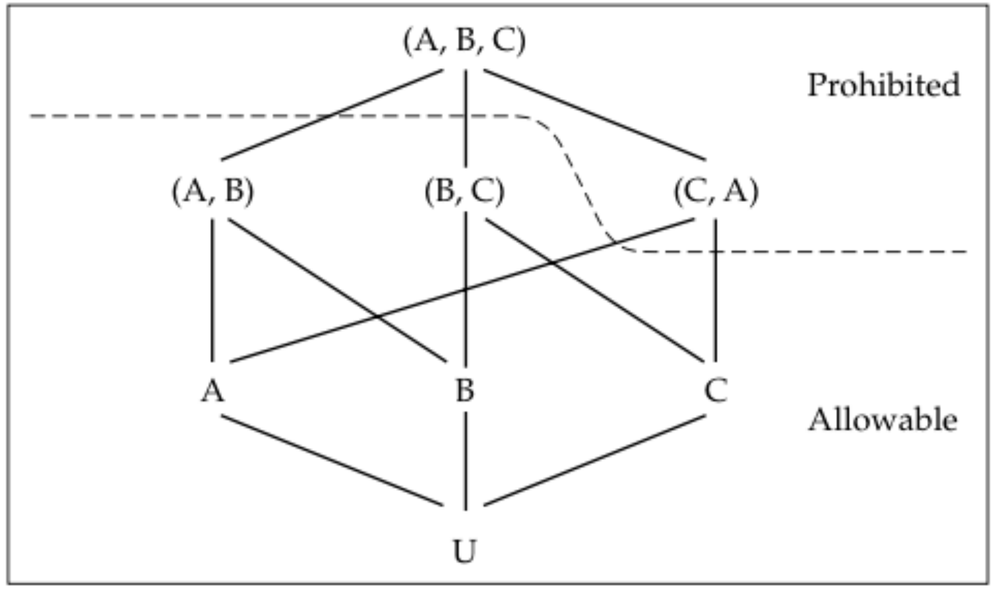

Figure 9.6: Table lattice for a database with three attributes

query overlap control

some systems try to relax the limits imposed by static query control by keeping track of who accessed what.

Known as *query overlap control i*nvolves rejecting any query from a user which, combined with what the user knows already, would disclose a sensitive statistic.

This suffers from two usually unsurmountable drawbacks. First, the complexity of the processing involved increases over time, and often exponentially.

Second, it's extremely hard to be sure that your users aren't in collusion, or use multiple identities, etc.

there exist several randomisation techniques, designed to degrade the signal-to-noise ratio from the attacker's point of view while impairing that of the legitimate user as little as possible.

- *perturbation:* adds noise with zero mean and a known variance to the data. It will tend to damage legitimate's users queries when the result set is small. Also, averaging techniques may remove the noise.
- *controlled tabular adjustment:* identify the sensitive cells and replace their values with sufficiently different ones, then adjust other values in the table to restore additive relationships
- *random sample queries:* make all the query sets the same size, selecting them at random from the available relevant statistics. Thus all the released data are computed from small samples rather than from the whole database.

the value of imperfect information

So doing de-identification right is hard and best avoided.

The main threat to databases of personal information is that once an organisation has access to potentially valuable data, then all sorts of ways of exploiting that value will be developed.

However, depending on the application even an imperfect de-identification system may destroy the value of data sufficienly enough.

Eg, if only five percent of the patients can be identified, and then only with effort, then the bank may decide that is not worth to try and profile borrowers using it, and will adopt more suitable approaches.