The University of Auckland

Trusted Computing: Open, Closed, or Both?

Seminar to HP Labs
Bristol

Prof. Clark Thomborson

24th April 2006

Topical Outline

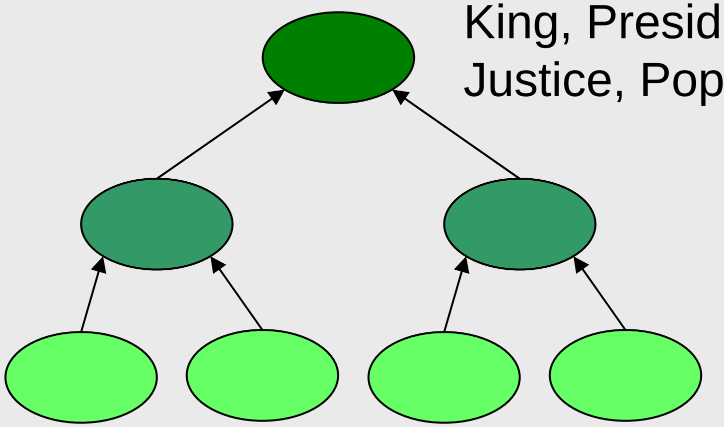
- Three types of trust:
  - Hierarchical, bridging, peering
- Three use cases:
  - Email, B2B e-commerce, DRM
- Three OS development methodologies:
  - Open, closed, hybrid

Technical and non-technical definitions of Trust

- In security engineering, placing trust in a system is a last resort.
  - It’s better to rely on an assurance (e.g. a proof, or a recourse mechanism), than on a trusting belief that “she’ll be right”.
- In non-technical circles, trust is a good thing: more trust is generally considered to be better.
- Trustworthiness (an assurance) implies that trust (a risk-aware basis for a decision) is well-placed.
  - A completely trustworthy system (in hindsight) is one that has never violated the trust placed in it by its users.
  - Just because some users trust a system, we cannot conclude that the system is trustworthy.
  - A rational and well-informed person can estimate the trustworthiness of a system.
  - Irrational or poorly-informed users will make poor decisions about whether or not, and under what circumstances, to trust a system.

Privilege in a Hierarchy

- Information flows upwards, toward the leading actor (at the root) of a secret society.
- Commands and trust flow downwards.
- The King is the most privileged.
- The peons are the most trusted.



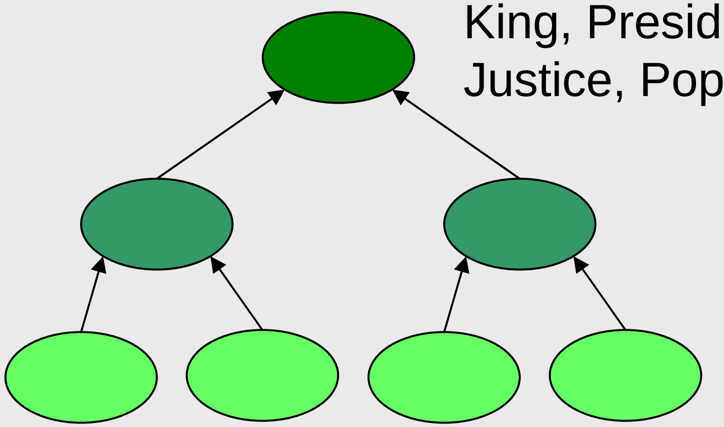
King, President, Chief Justice, Pope, or …

Peons, illegal immigrants, felons, excommunicants, or …

- Information flowing up is “privileged”.
- Information flowing down is “trusted”.
- Orange book TCSEC, e.g. LOCKix.

Trustworthiness in a Hierarchy

- In a secret society, information flows upwards, toward the most powerful actor.
- Commands and trust flow downwards.
- Peons must be trusted with some information!
- If the peons are not trustworthy, then the system is not secure.



King, President, Chief Justice, Pope, or …

Peons, illegal immigrants, felons, excommunicants, or …

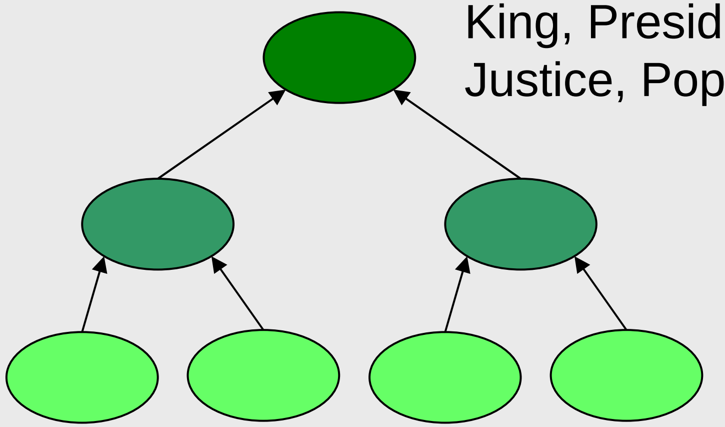
- If the King does not show good leadership (by issuing appropriate commands), then the system will not work well. “Noblesse oblige”!

Email in a Hierarchy

- Information flows upwards, toward the leading actor.

⇒ Actors can send email to their superiors.

- Non-upwards email traffic is trusted:
  - not allowed by default;
  - should be filtered, audited, …



King, President, Chief Justice, Pope, or …

Peons, illegal immigrants, felons, excommunicants, or …

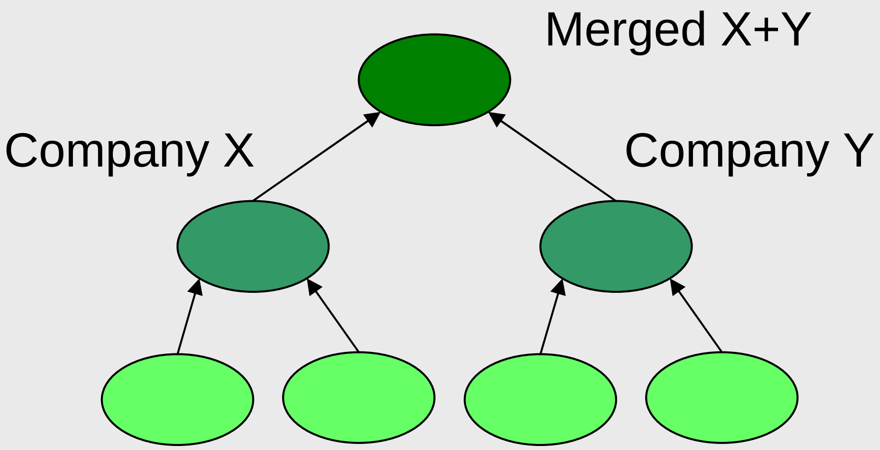
- Email up: “privileged” (allowed by default)
- Email down: “trusted” (disallowed by default, risk to confidentiality)
- Email across: privileged & trusted routing

Email across Hierarchies

Q: How should we handle email between hierarchies?

Answers:

1. **Merge**
2. Subsume
3. Bridge



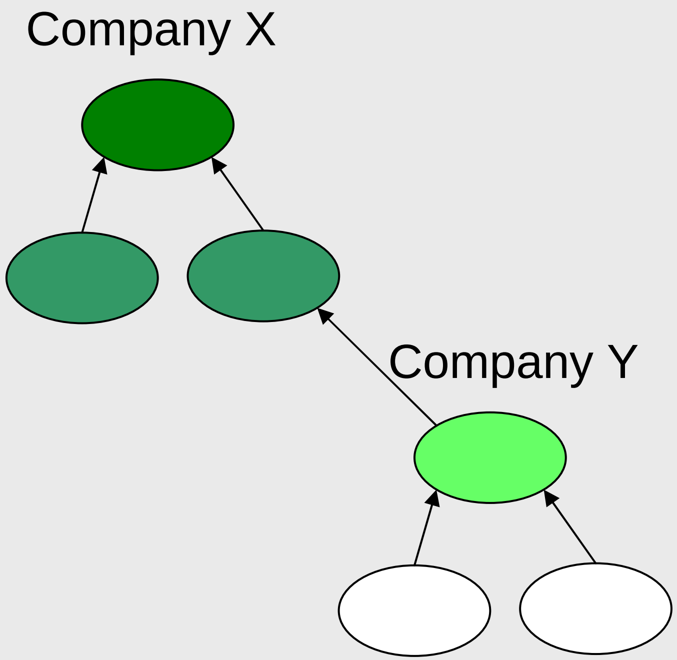
- Not often desirable or even feasible.
- Cryptography doesn’t protect X from Y, because the CEO of the merged company has the right to know all keys.
- Can a noble CEO(X+Y) be found?

Email across Hierarchies

Q: How can we manage email between hierarchies?

Answers:

1. Merge
2. **Subsume**
3. Bridge

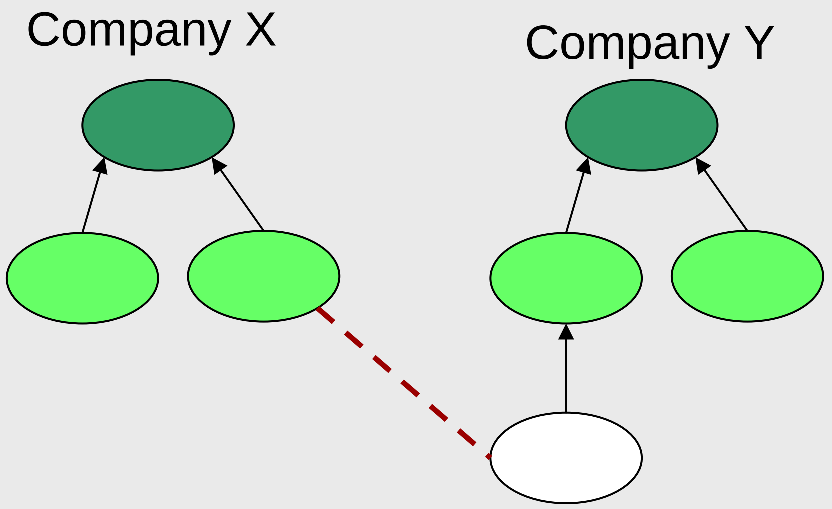


Email across Hierarchies

Q: How can we manage email between hierarchies?

Answers:

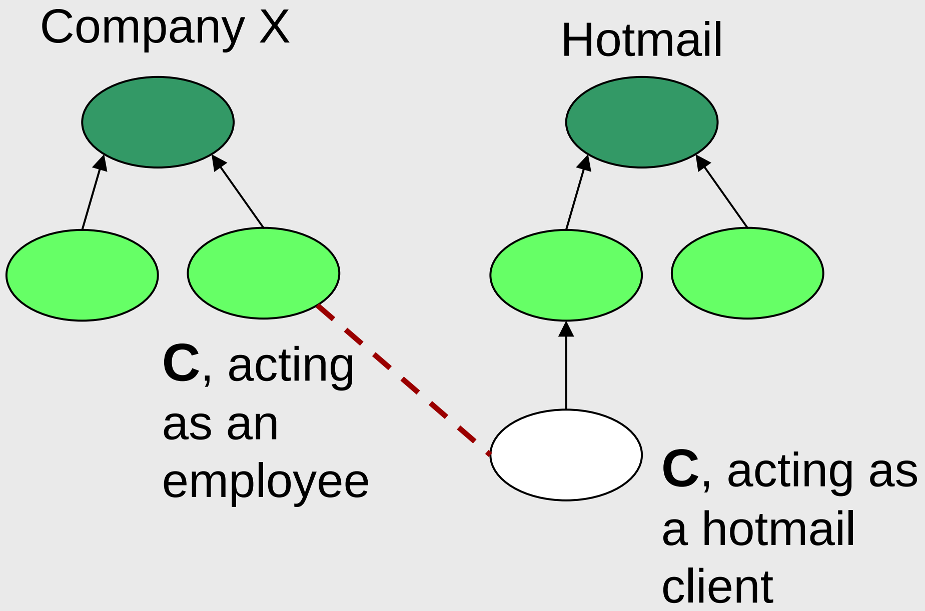
1. Merge
2. Subsume
3. **Bridge!**



- Bridging connection: trusted in both directions.

Bridging Trust

- We make bridges every time we send personal email from our work computer.
- We make bridges every time we send work-related email from our home computer.
- Even Kings can form bridges.
- However Kings are most likely to use an actual person, e.g. their personal secretary, rather than a bridging persona.

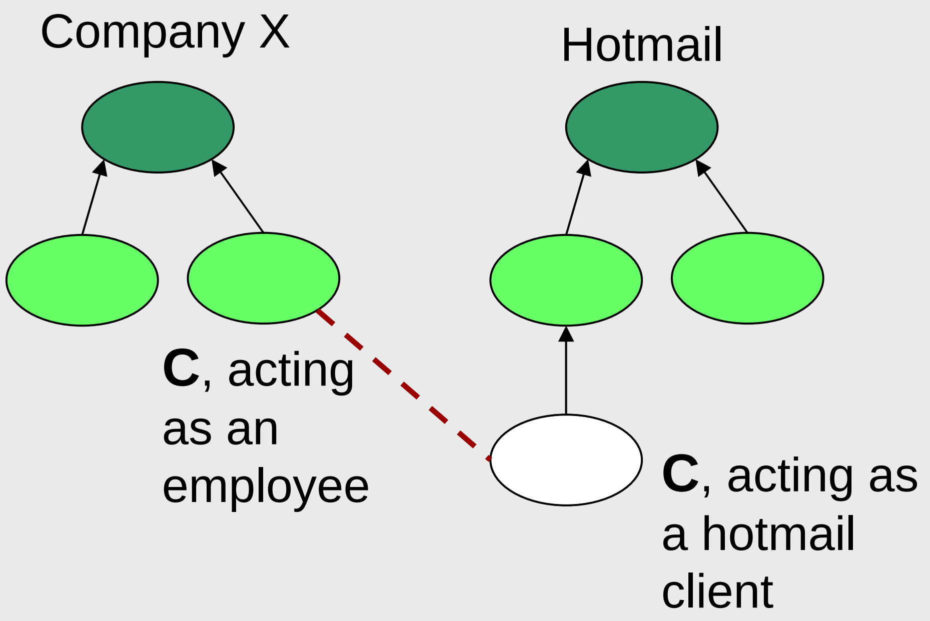


Company X
Hotmail
C, acting as an employee
C, acting as a hotmail client

- Bridging connection: bidirectional trusted.
- Used for all communication among an actor’s personae.
- **C** should encrypt all hotmail to avoid revelations.

Personae, Actors, and Agents

- I use “actor” to refer to
  - an agent (a human, or a computer program),
  - pursuing a goal (risk vs. reward),
  - subject to some constraints (social, technical, ethical, …)
- In Freudian terms: ego, id, superego.
- Actors can act on behalf of another actor: “agency”.
- In this part of the talk, we are considering agency relationships in a hierarchy.



Company X

Hotmail

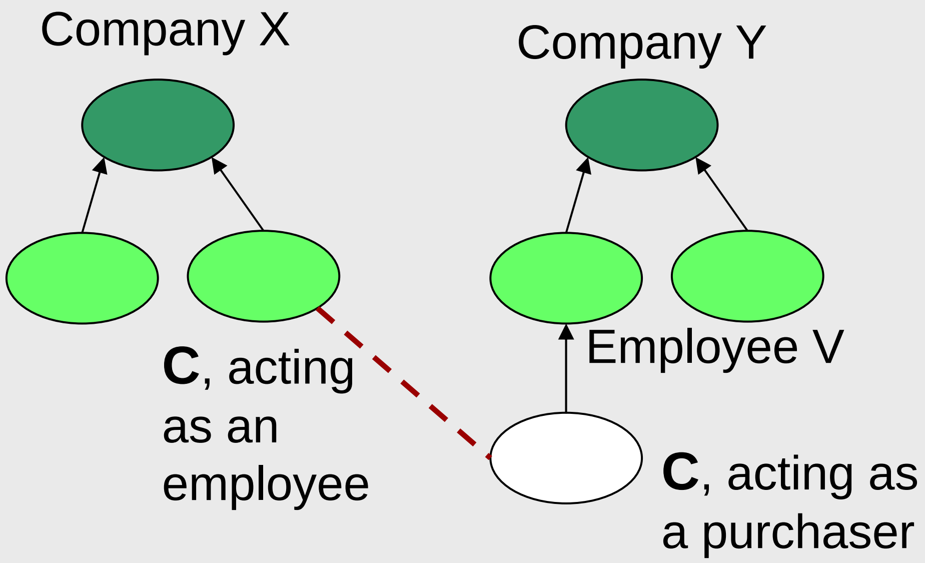
C, acting as an employee

C, acting as a hotmail client

- When an agent takes on a secondary goal, or accepts a different set of constraints, they create an actor with a new “persona”.
- Bridging connection: bidirectional trusted, models communication among an agent’s personae.

Bridging Trust: B2B e-commerce

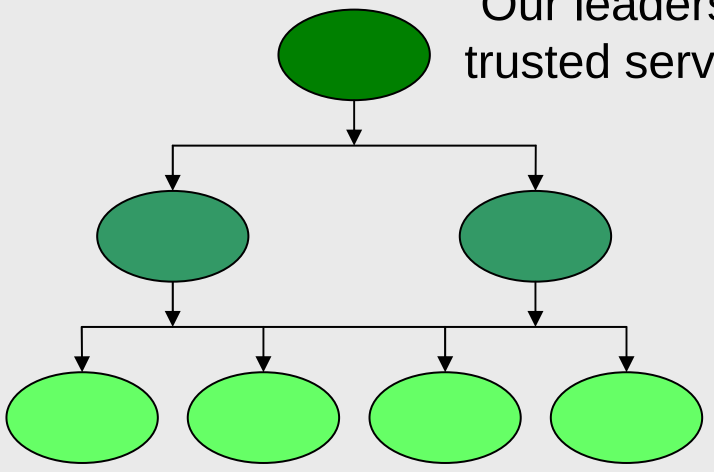
- Use case: employee **C** of X purchasing supplies through employee V of Y.
- Employee **C** creates a hotmail account for a “purchasing” persona.
- Purchaser **C** doesn’t know any irrelevant information.



- Most workflow systems have rigid personae definitions (= role assignments).
- Current operating systems offer very little support for bridges. Important future work!

Why can’t we trust our leaders?

- Commands and trust flow upwards (by majority vote, or by consensus).
- Information flows downwards by default (“privileged”).
- Upward information flows are “trusted” (filtered, audited, etc.)
- In a peerage, the leading actors are trusted, have minimal privilege, don’t know very much, and can safely act on anything they know.



“Our leaders are but trusted servants…”

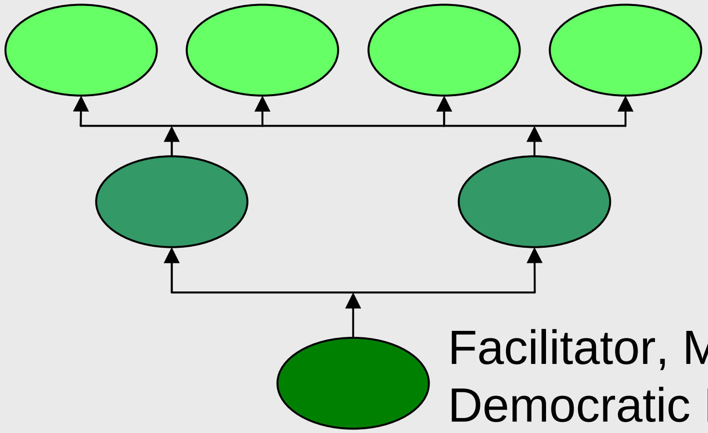
Peers

- By contrast, the King of a hierarchy has an absolute right (“root” privilege) to know everything, is not trusted, and cannot act safely.

Turn the picture upside down!

- Information flows upwards by default ("privileged").
- Commands and trust flow downwards.
- Downward information flows are "trusted" (filtered, audited, etc.)
- A peerage can be modeled by Bell-La Padula, because there is a partial order on the actors' privileges.

Peers, Group members, Citizens of an ideal democracy, …



Facilitator, Moderator, Democratic Leader, …

- Equality of privilege is the default in a peerage, whereas inequality of privilege is the default in a hierarchy.

Peer trust vs. Hierarchical trust

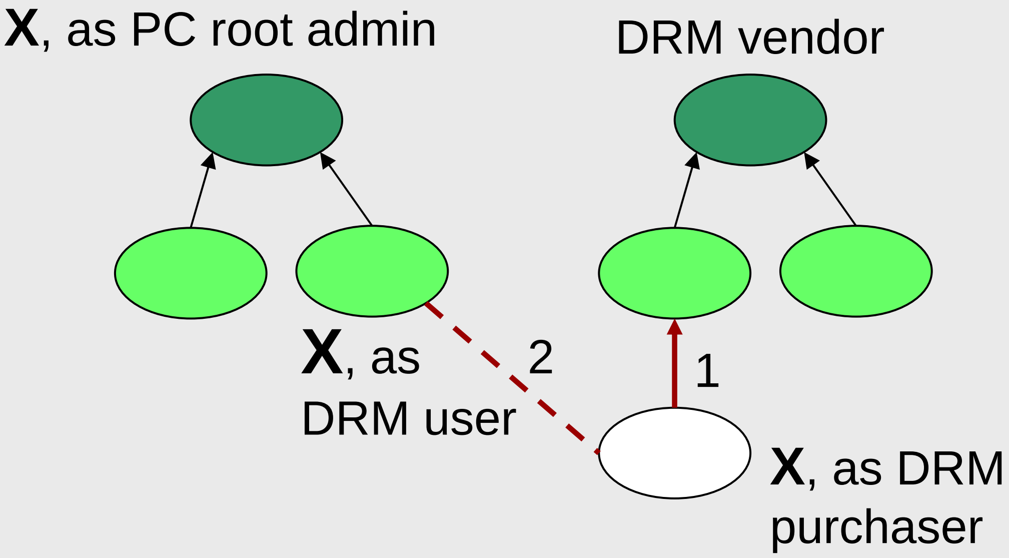
- Trusting decisions in a peerage are made by peers, according to some fixed decision rule.
  - There is no single root of peer trust.
  - There are many possible decision rules, but simple majority and consensus are the most common.
  - Weighted sums in a reputation scheme (e.g. eBay for goods, Poblano for documents) are a calculus of peer trust -- but “we” must all agree to abide by the scheme.
  - “First come, first serve” (e.g. Wiki) can be an appropriate decision rule, if the cost per serving is sufficiently low.
- Trusting decisions in a hierarchy are made by its most powerful members.
  - Ultimately, all hierarchical trust is rooted in the King.

Legitimation and enforcement

- Hierarchies have difficulty with legitimation.
  - Why should I swear fealty (give ultimate privilege) to this would-be King?
- Peerages have difficulty with enforcement.
  - How could the least privileged actor possibly be an effective facilitator?
- This isn’t Political Science 101!
  - I won’t argue whether ideal democracies are better than ideal monarchies.
  - I will argue that hierarchical trust is quite different to peer trust, that bridging trust is also distinct, and that all three forms are important in our world.
- My thesis: Because our applications software will help us handle all three forms of trust, therefore our operating systems should support all three forms.

Trust in DRM on home PCs

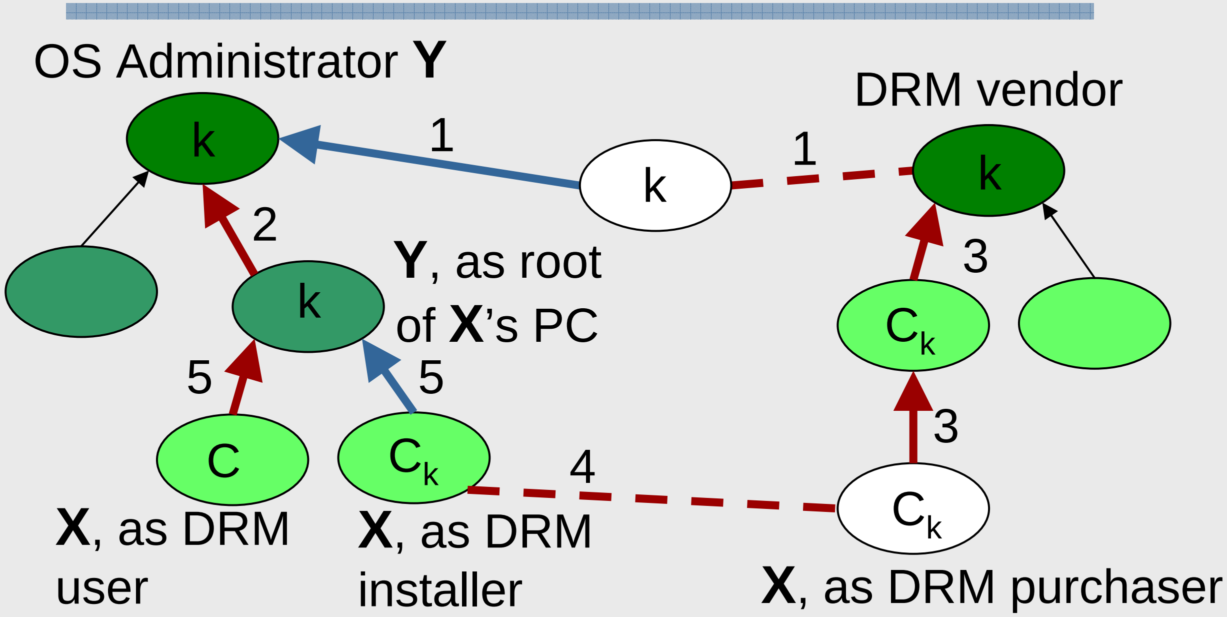
- Let us assume that user **X** is has root privilege on their home PC.
- User **X** can read and write anything stored on their PC.
- Anyone who sells DRM content to **X**'s DRM purchasing persona must trust **X**'s root-admin persona.



1. The DRM vendor makes a trusting transfer of information (sale of DRM content).
2. User **X** makes a trusting transfer of information (storing DRM content on their PC) between their purchasing persona and their using persona.

DRM on a trusted PC

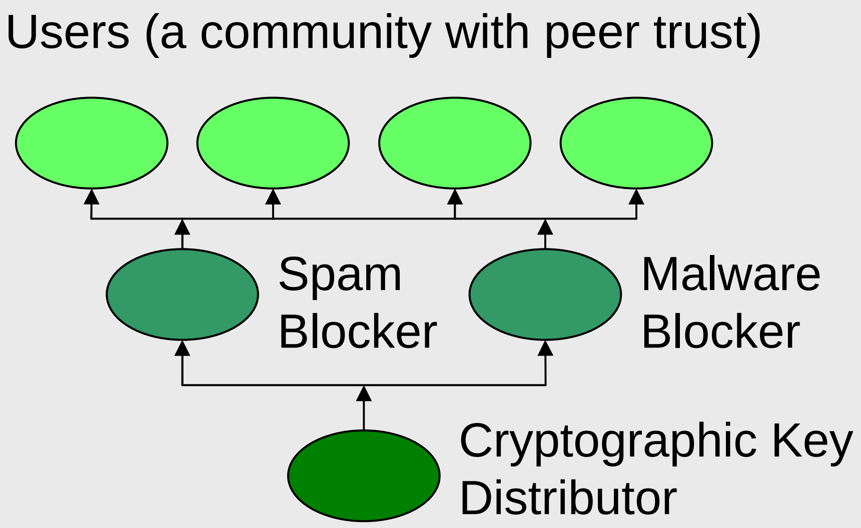
- **X** has given root privilege on their PC to an OS Admin **Y**.
- The DRM vendor must trust **Y** not to redistribute this content.



1. A persona of a DRM vendor has a contractual agreement with an OS Admin **Y**, under which **Y** is given privileges to a content-decryption key.
2. OS Admin **Y** writes a key into **X**'s kernel. This is a privileged transfer.
3. DRM vendor makes a trusting sale of encrypted DRM content.
4. User **X** makes a trusting storage of DRM-protected content (http://www.e.govt.nz/policy/trust-security/).
5. User X plays the decrypted content in an OS partition under Y's control.

Peer administration of trusted PCs

- Information flows upwards by default ("privileged").
- Commands and trust flow downwards.
- Downward information flows are "trusted" (filtered, audited, etc.)
- The users must refrain from reading the state of the key generator: this is an enforcement problem.
- Peerages have enforcement problems, and hierarchies have legitimation problems.
- Enforcement problems are manageable if no single infraction causes much damage, and if the cost of detection & response is small.


Users (a community with peer trust)
Spam Blocker
Malware Blocker
Cryptographic Key Distributor

- Disclaimer: this is vapourware. Some components are available.
- How could this system gain the trust of a major corporation? (Let's look at our use cases….)

Trusted email

- When a hierarchical organisation receives email, the first question is “who is it from?”
  - The answer to this question determines the privilege level of the incoming data.
  - Email from a privileged persona is confidential – it can be delivered only to an actor
  - Email from an unprivileged persona can be read by anyone in the hierarchy.
- When a hierarchy sends email, the first question is whether it is low-high (privileged), high-low (trusted), or incomparable (cross-hierarchy: low-high-low).
  - Trusted email should be filtered and audited.
- The first questions can be answered accurately if the hierarchy can reliably associate personae with their cryptographic key(s).

Public Key Infrastructures

- Cryptographic keys can be associated with user-ids, using a “digital certificate” in a Public Key Infrastructure (PKI).
- Verisign issues certificates linked to credit card numbers.
  - These are suitable for e-commerce, but not for email (except in wealthy communities).
- Some PKIs issue certificates linked to email addresses.
  - It is far from clear how we can maintain a secure associations between an email address and a persona.
  - The Web of Trust issues digital certificates using a peer-trust model, however these certificates have not met widespread acceptance (perhaps because we don’t yet use personae!).
  - Important future work!

E-commerce

- Corporations use idiosyncratic databases to identify their customers.
- Federated networks (e.g. Liberty Alliance) offer a compatibility layer (involving cryptographic keys) to organizations who agree to correlate their identity databases.
  - “Single sign-on”: (Employee X of Company Y) == (Customer W of Company Z)
  - (Shopping persona of X) == (Credit-card possessing persona of X)
  - This allows organizations to model bridging trust!
- Nothing (other than a lack of trust) prevents peerages from sharing keys with hierarchical organisations, in a federated identity network.
- Corporate organizations rely on financial and legal structures
  - to legitimate their hierarchies of privilege, and
  - to enforce obedience to hierarchical commands (trust).
- Peerages will need financial and legal support, especially for enforcement, before they would be trusted by corporates. This is conceivable: “identity theft” is popularly understood to be a crime.

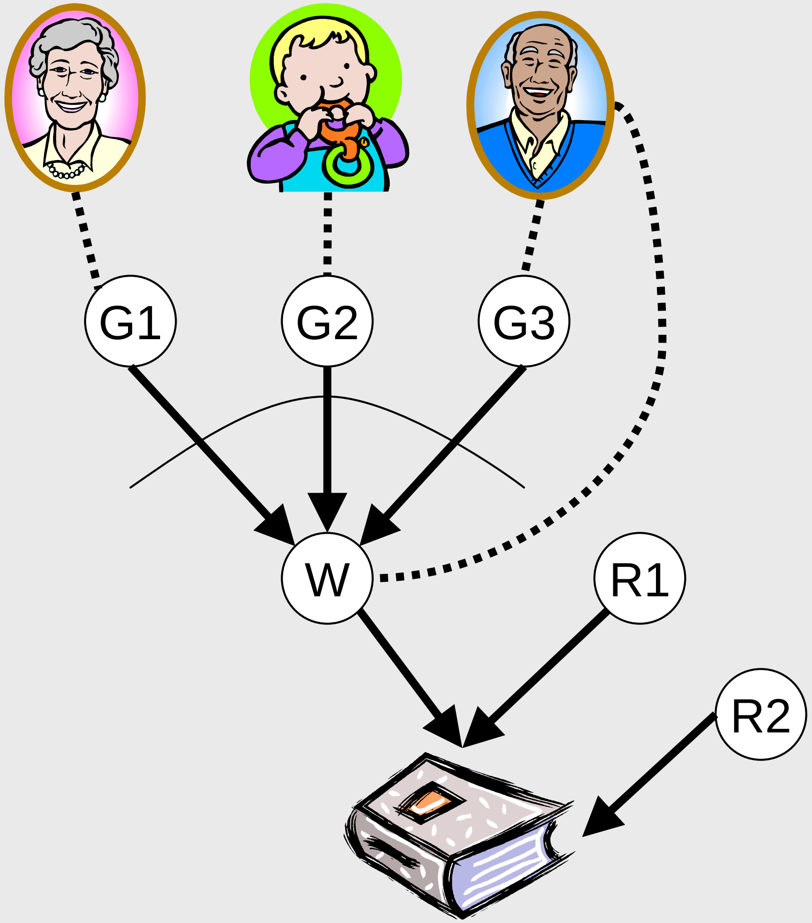
Peer-trust DRM

- A peerage could establish a good reputation for “fair trading” in DRM objects.
- Such a peerage would have to self-enforce a widely-accepted set of DRM rules.
- iTunes is an interesting test case
  - It is easy to bypass DRM restrictions in iTunes, implying that users have privileged access to their DRM stores, and that they are trusted by the DRM vendor not to abuse this privilege.
- iTunes is hierarchical, but a peer-trust system might be able to enforce similar rules.
  - An important leadership role in a peer-trust system is to evict peers who don’t follow the rules.

DRM for Baby Books

- Grandparents, parents, child are the Guardians of a Baby Book.
  - They collectively control access rights.
  - They may decide to add (or delete) Guardians.
  - They must trust each other to “do the right thing” when in physical possession of the Baby Book.
  - Guardians may allow others to read the Book, if this will not be inappropriate (as judged by other Guardians).
  - At most one Guardian can write, at any given time.
  - The Book should not be lost or damaged.
- Can an online Baby Book meet these requirements?

Trust, for Baby Books



- Guardian **G3** is in a bridge-trust relationship with Writer **W**.
- **W** is trusted by (**G1**, **G2**, **G3**) not to abuse the write-privilege.
- **G3** is in a peer-trust relationship with (**G1**, **G2**).
- **G3** is trusted by (**G1**, **G2**) to take good care of the Book.
- Reader personae (**R1**, **R2**) are not currently animated.
- Actions of all these personae must be subject to review by (**G1**, **G2**, **G3**): auditable, non-repudiable, and confidential.
- The Book must be confidential, have integrity, and be available.

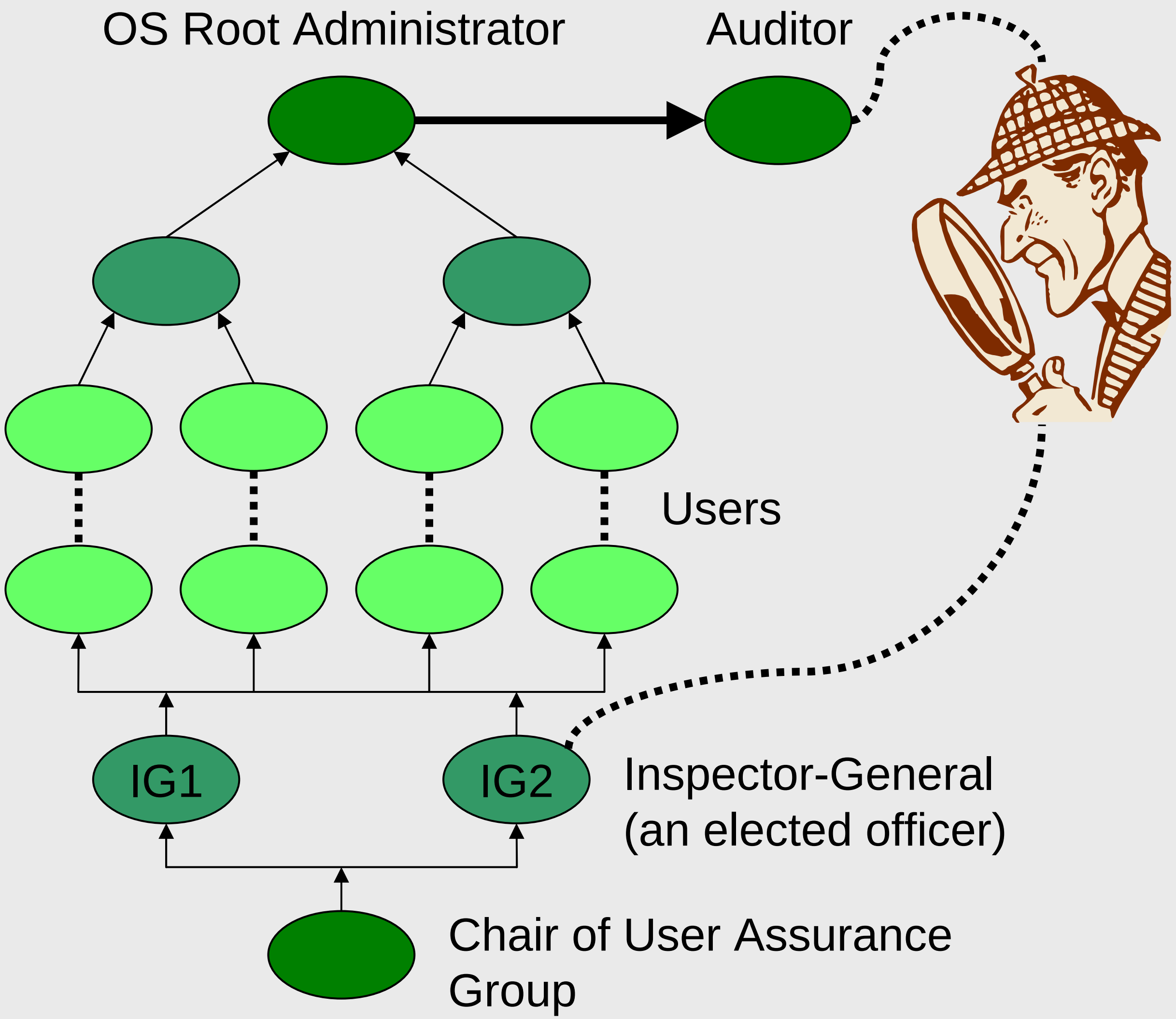
Open vs. Closed source

- Closed-source methodology is hierarchical.
  - Only personae with sufficiently high privilege can read the source.
  - An obfuscated OS kernel can hold, and manipulate, cryptographic keys for a remote, trusted, OS administrator.
  - A trusted computer base (TCB) can offer additional security through hardware (TPM) enforcement of privilege.
- Open-source development is based on peer trust.
  - The OS kernel can’t hold any secrets if it is open source.
  - A trusted computer base (TCB) can hold keys in privileged (closed) hardware.

More vapourware

- Closed-source methodology is appropriate for designing hierarchical systems.
  - These systems have trouble with legitimation.
  - Why should a user trust that the system designers (and administrators) won't abuse their privilege?
- Open-source methodology is appropriate for designing peerage systems.
  - These systems have trouble with enforcement.
  - Why should anyone trust a user not to abuse their privilege?
- Real-world peerages can legitimise hierarchies, and hierarchies can enforce peerages.
  - Why shouldn't our next-generation OS use this design pattern?

A Legitimised Hierarchy



- Each assurance group may want its own Audit (different scope, objectives, Trust, …).
- The OS Administrator may refuse to accept an Auditor.
- The OS Administrator makes a Trusting appointment when granting auditor-level Privilege to a nominee.
- Assurance organizations may be hierarchical, e.g. if the Users are governmental agencies or corporate divisions.

Review & Future Work

- Three types of trust: hierarchical, bridging, peering.
  - Information flows are either trusted or privileged.
- Hierarchical trust has been explored thoroughly in the Bell-La Padula model.
  - A subordinate actor is trusted to act appropriately, if a superior actor delegates some privileges.
  - Bell-La Padula, when the hierarchy is mostly concerned about confidentiality.
  - Biba, when the hierarchy is mostly concerned about integrity.
  - ***A general purpose OS must support all concerns of a hierarchy.***
- Actors have multiple personae.
  - Bridging trust connects all an actors' personae.
  - ***A general purpose OS must support personae.***
- Peering trust is a shared decision to trust an actor who is inferior to the peers.
  - Peerages have trouble with enforcement; hierarchies have trouble with legitimation.
  - ***A trusted OS must be a legitimate enforcement agent!***
- We are starting to develop a dynamic theory of trust.
  - When we accept a subordinate role in a hierarchy or in a peerage, we make a trusting decision.
  - ***"Subordination trust" is required when installing a trusted OS.***
  - Dynamic trust is also required to model changes in membership of a peerage.
  - Dynamic trust/distrust is required to create/destroy an arc in our diagrams. …

Acknowledgements & Sources

- Privilege, trust: Richard O'Brien, Clyde Rogers, “Developing Applications on LOCK”, 1991.
- Personae: Jihong Li, “A Fifth Generation Messaging System”, 2002; and Shelly Mutu-Grigg, “Examining Fifth Generation Messaging Systems”, 2003.
- Reputation systems: Benjamin Lai, Trust in Online Trading Systems, 2004.
- Trusted OS, use cases: Matt Barrett, “Towards an Open Trusted Computing Framework”, 2005; and “Using NGSCB to Mitigate Existing Software Threats”.
- Use cases: Qiang Dong, “Workflow Simulation for International Trade”, 2002.
- Use cases: WTC, Telecom NZ, ADLS, SOEI, Microsoft.
- I want to find more NZ corporate partners with an interest in trusted computing.
- I want to find overseas collaborators/contributors to this research.