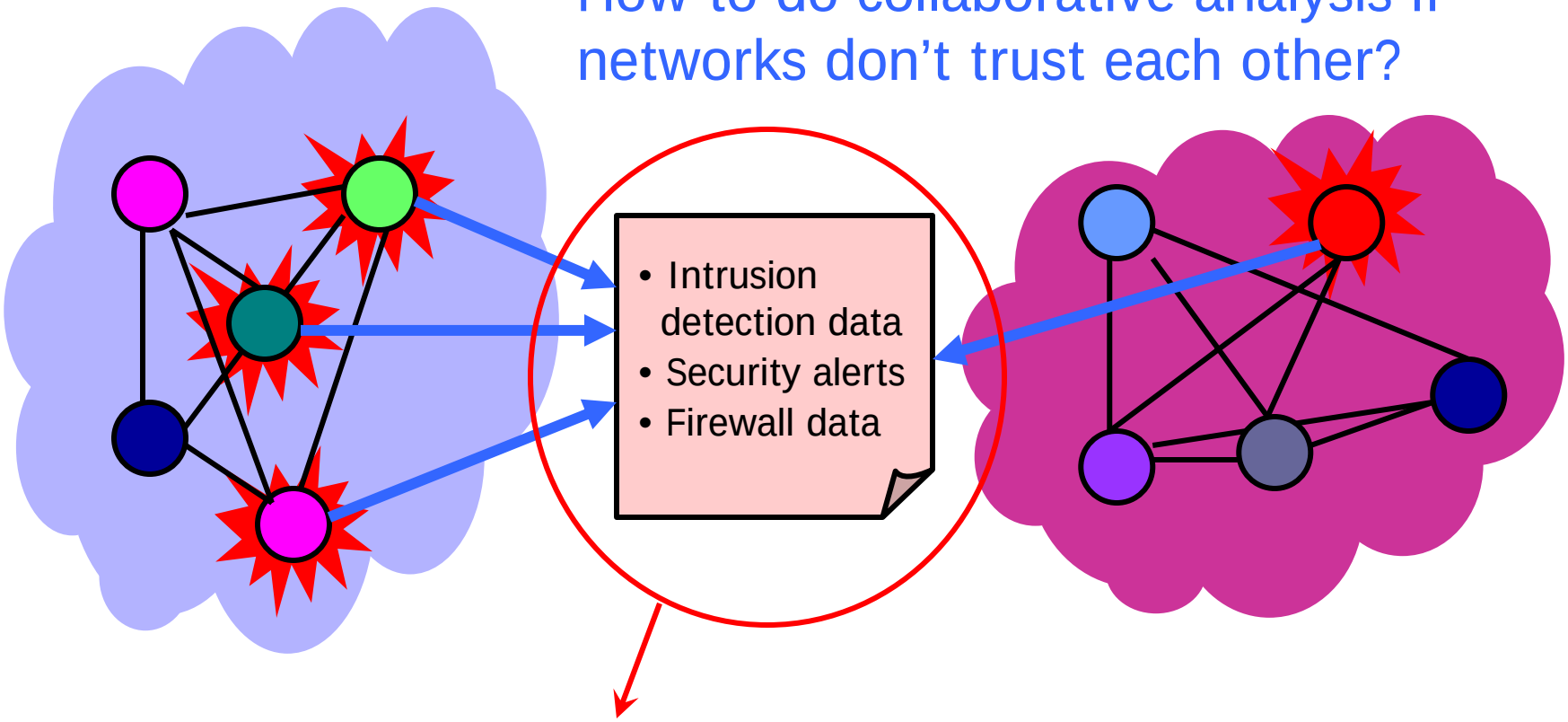
Threats to Anonymized Datasets

Vitaly Shmatikov

The University of Texas at Austin

Big Picture

How to do collaborative analysis if networks don't trust each other?



Can we stop attackers from exploiting these data? Yes.

... AND keep the data useful for legitimate research? **Much harder!**

Sample Security Alert

Source_IP		
Source_Port	FW,ID	Source TCP or UDP port of the machine that initiated the session or transferred the transaction that caused the alert to fire.
Dest_IP	FW,ID,AV	Typically refers to the destination IP address of the machine that ini-
Dest_Port		
Protocol		
Timestamp	FW,ID,AV	May incorporate incident start time, end time, incident report time.
Sensor_ID	FW,ID,AV	May incorporate the brand and model of the sensor and a unique iden-
Count	I	alert or event (*e.g.*, packet) level.
Event_ID	FW,ID,AV	Uniquely defines the alert type for the given sensor.
Outcome	FW,ID,AV	Reports the status or disposition of the reported activity. For firewalls, with an allow or on (*e.g.*, Syman- or quarantined). c.
Captured_Data	ID	Some IDS sensors have the ability to report part or all of the data content in which the alert was applied.
Infected_File	AV	Antivirus logs include the identity of the file that was infected.

- **may contain victim's IP address**
- **reveals relationships with other networks**

- **reveals target's IP address**
- **reveals topology of targeted network and attack propagation**

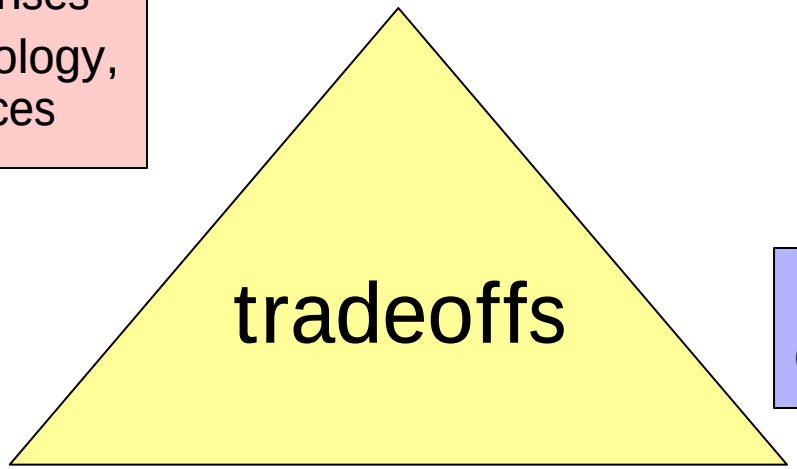
- **may reveal organization that owns it**

- **leaks information stored on targeted systems**

Privacy vs. Utility Tradeoff

- Do not enable attackers to track attack propagation
- Do not announce site defenses
- Do not reveal network topology, configuration, enabled services

privacy and anonymity



Low overhead; no complicated crypto

utility

efficiency

Support (at least) coarse-grained analysis:
event trends, identification of common attack sources, connection patterns, blacklisting, etc.

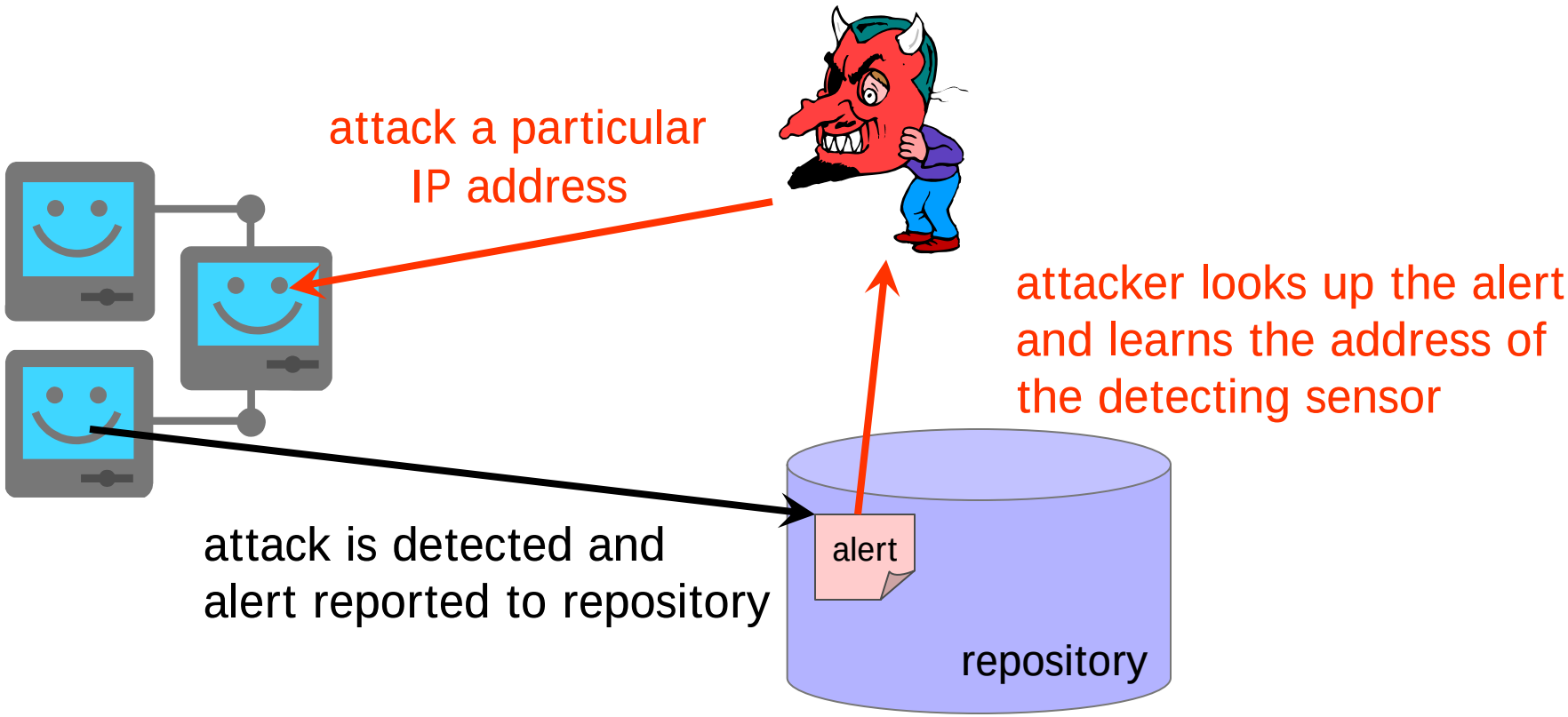
Fundamental Problem

- Open alert repository may be used to track progress of attacks and find new vulnerabilities
- It's difficult to tell the difference between an attacker and a legitimate security researcher

alert database

- Sometimes, the only difference is intent
  - Hard to figure out by looking at data requests

Example: Probe-Response Attack

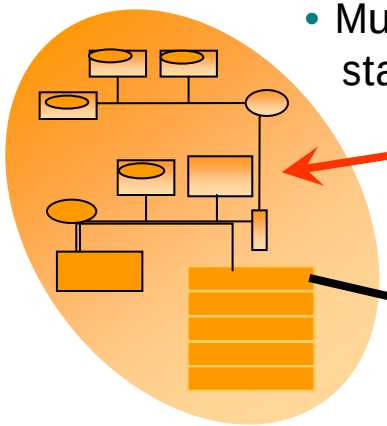


IP hashing doesn't help! Attacker knows targeted subnet, can stage simple dictionary attack with small (<256) dictionary

“Fingerprinting” Attacks

Unique attack signature

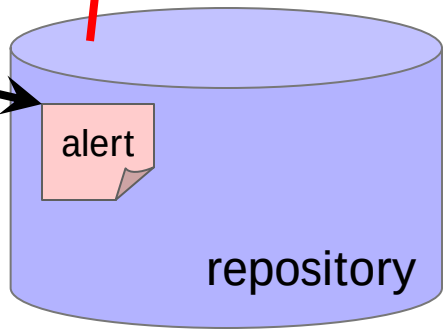
- Port combinations
- Rare Snort rules
- Multiple scans (to cross statistical thresholds)



Attacker wants attack to be detected

Attacker completely maps out network defenses and avoids them in the future

Attack is detected and alert reported to repository



[E.g., see Bethencourt et al., USENIX Security 2005]

What Does Not Work

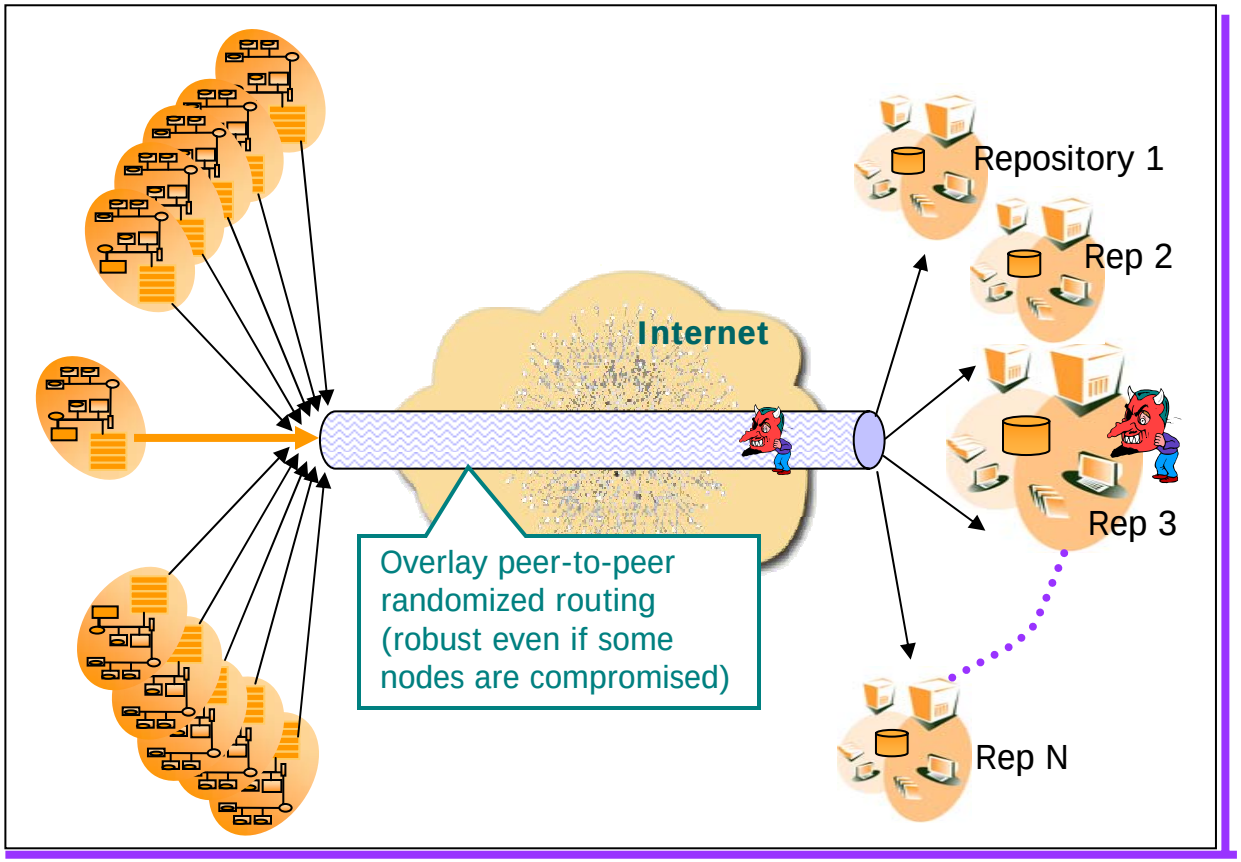
- Hashing doesn't work
  - Dictionary / plaintext space too small
- Keyed hashing (e.g., HMAC) doesn't work
  - Key question: Who knows the key?
  - Nobody – collected data are useless
    - Can't even compare hashes for equality
  - Everybody – same problems as plain hashing
- Prefix-preserving hashing doesn't work
  - Same problems are keyed hashing

Other Ideas

- Re-keying by repository
  - Additional keyed hashing of IP addresses by repository
- Randomized hot list thresholds
  - Publish only the hot list of reported alerts that have something in common (e.g., hash of source IP address)
    - Note: thresholds must be random to prevent flooding attacks
- Delayed alert publication
  - Batch alerts before publishing them
  - Suppress or round up timestamps

... all of these assume that repository has not been compromised!

Randomized Alert Routing



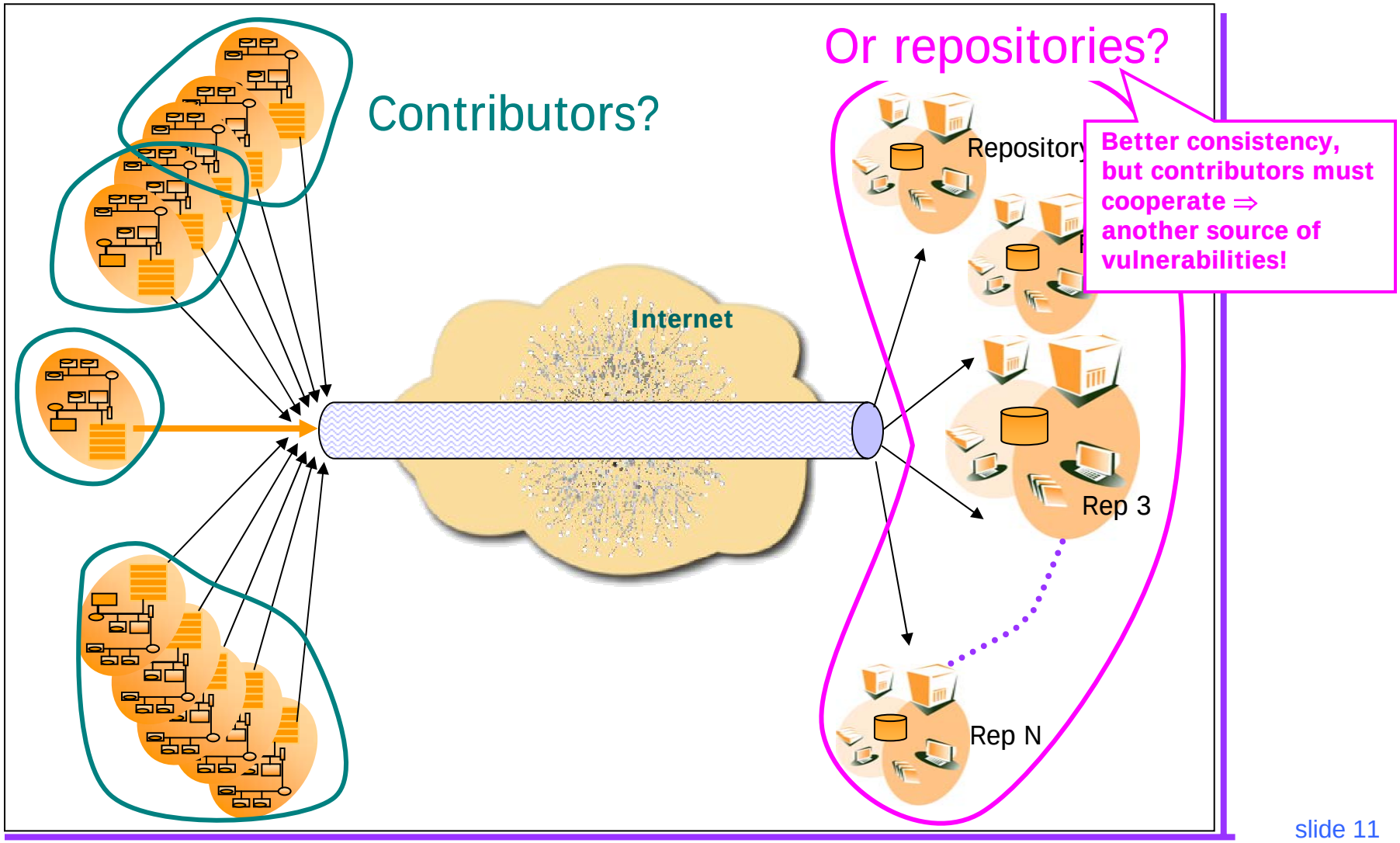
Multiple alert repositories

- Alerts randomly distributed to several locations
- Attacker workfactor increased
- Collaborative pattern finding more difficult, but can still observe large-scale trends

Randomized routing provides source anonymity

- Hide alerts' origins even from the receiving repository

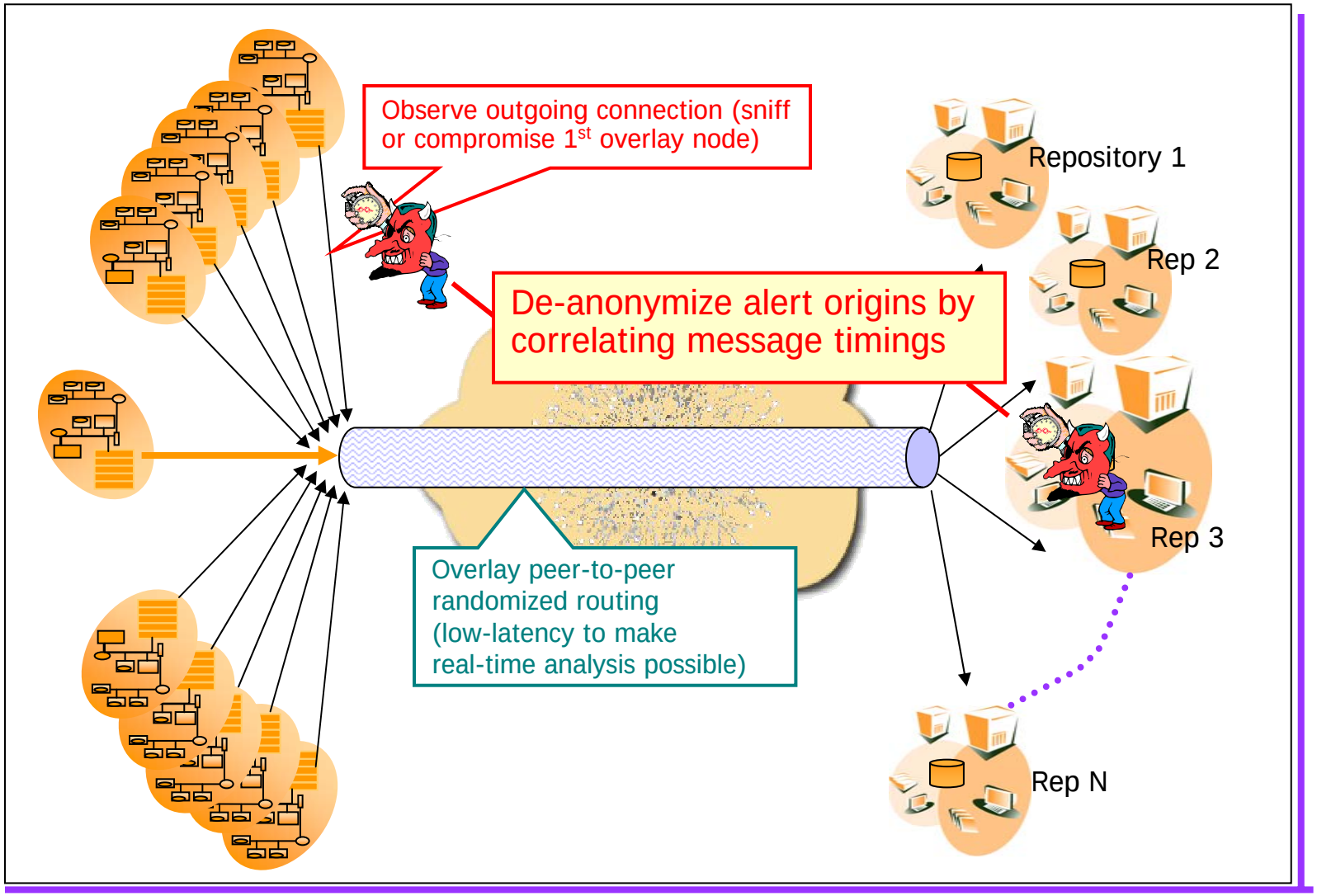
Who Manages Anonymization?



Other Ideas

- Data sanitization methods from other communities
  - E.g., statistical sanitization of US Census data
- Legal defenses and safeguards
  - Non-disclosure agreements; contributor agreements prohibiting reverse-engineering of sanitization policies
  - Issue: does repository have a legal liability?
- Vet people who have access to repository
  - Effectively increase “circle of trust”; make obtaining data more difficult
- Fundamental conflict between privacy and utility
  - Less liability = more privacy = less utility

Timing Attack



Issues with Source Anonymity

- Spamming and denial of service
  - If contributors send garbage anonymously?
- Possible solution: anonymous group credentials
  - Registration is necessary
    - Each contributor is vetted, issued a special crypto credential
  - Contributors are anonymous up to their membership in the authorized group
    - Recipient can verify that the credential is valid, but not the identity of the contributor to whom it has been issued
- This is only as secure as the weakest contributor
  - Once attacker learns a valid credential, it's over