Theory and Design of Low-latency Anonymity Systems (Lecture 4)

Paul Syverson
U.S. Naval Research Laboratory
syverson@itd.nrl.navy.mil
http://www.syverson.org

Course Outline

Lecture 1:

- Usage examples, basic notions of anonymity, types of anonymous comms systems
- Crowds: Probabilistic anonymity, predecessor attacks

Lecture 2:

- Onion routing basics: simple demo of using Tor, network discovery, circuit construction, crypto, node types and exit policies
- Economics, incentives, usability, network effects

Course Outline

Lecture 3:

- Formalization and analysis, possibilistic and probabilistic definitions of anonymity
- Hidden services: responder anonymity, predecessor attacks revisited, guard nodes

Lecture 4:

- Link attacks
- Trust

Link attacks overview

Background

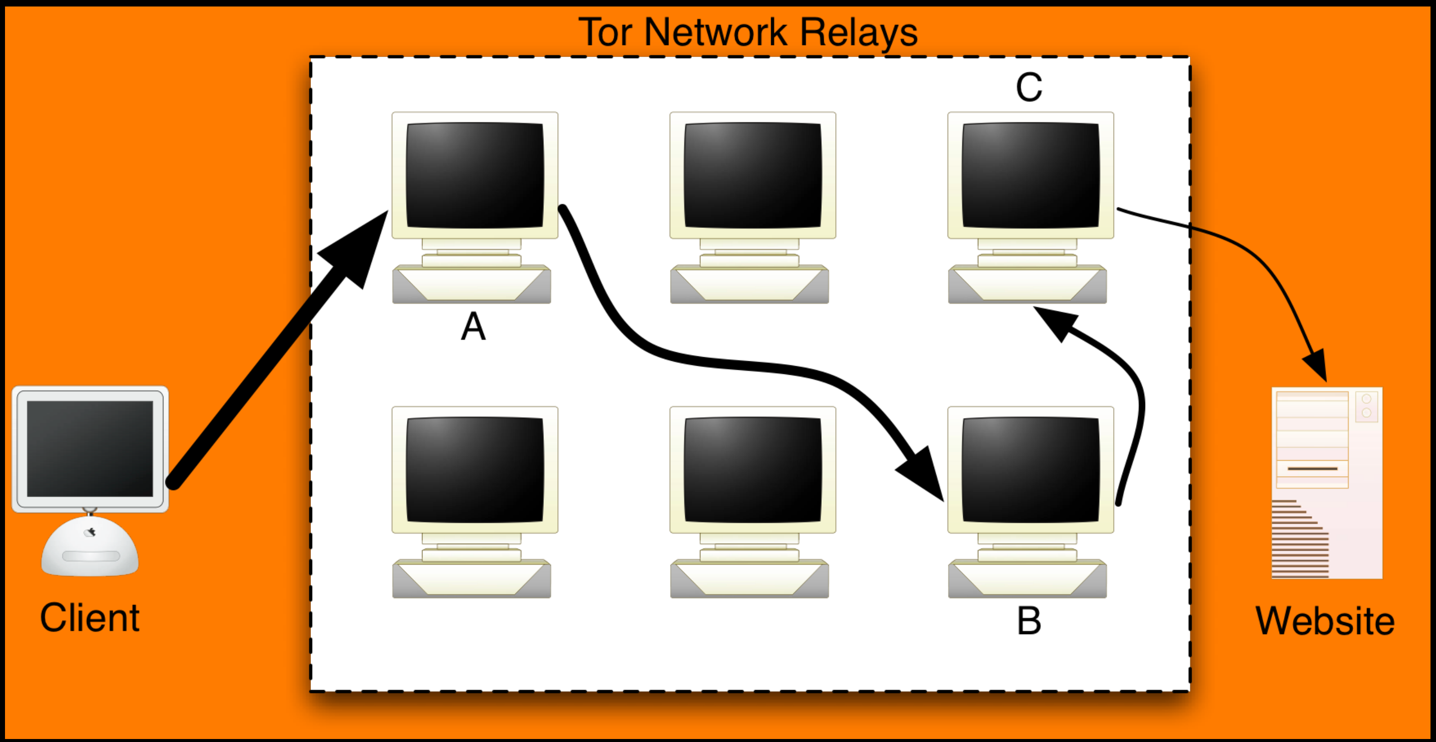
AS Path Inference

Analysis of Tor network growth

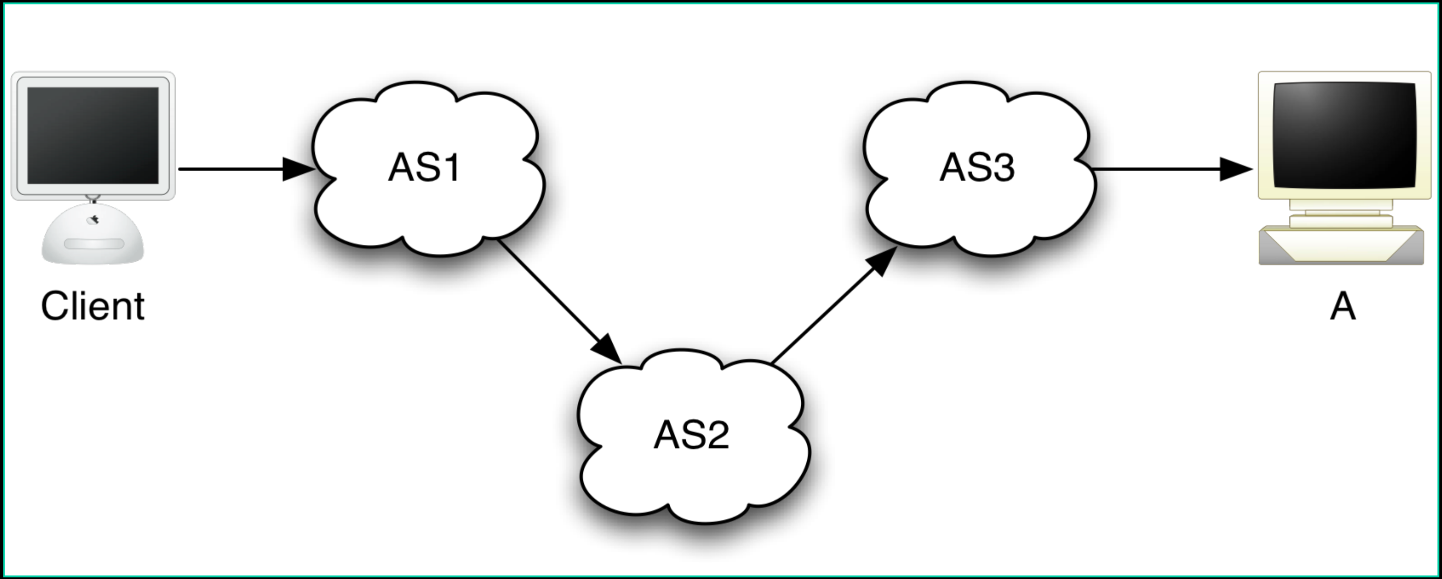
Tor AS statistics

Proposed path selection heuristics

Tor: A three-hop onion routing network

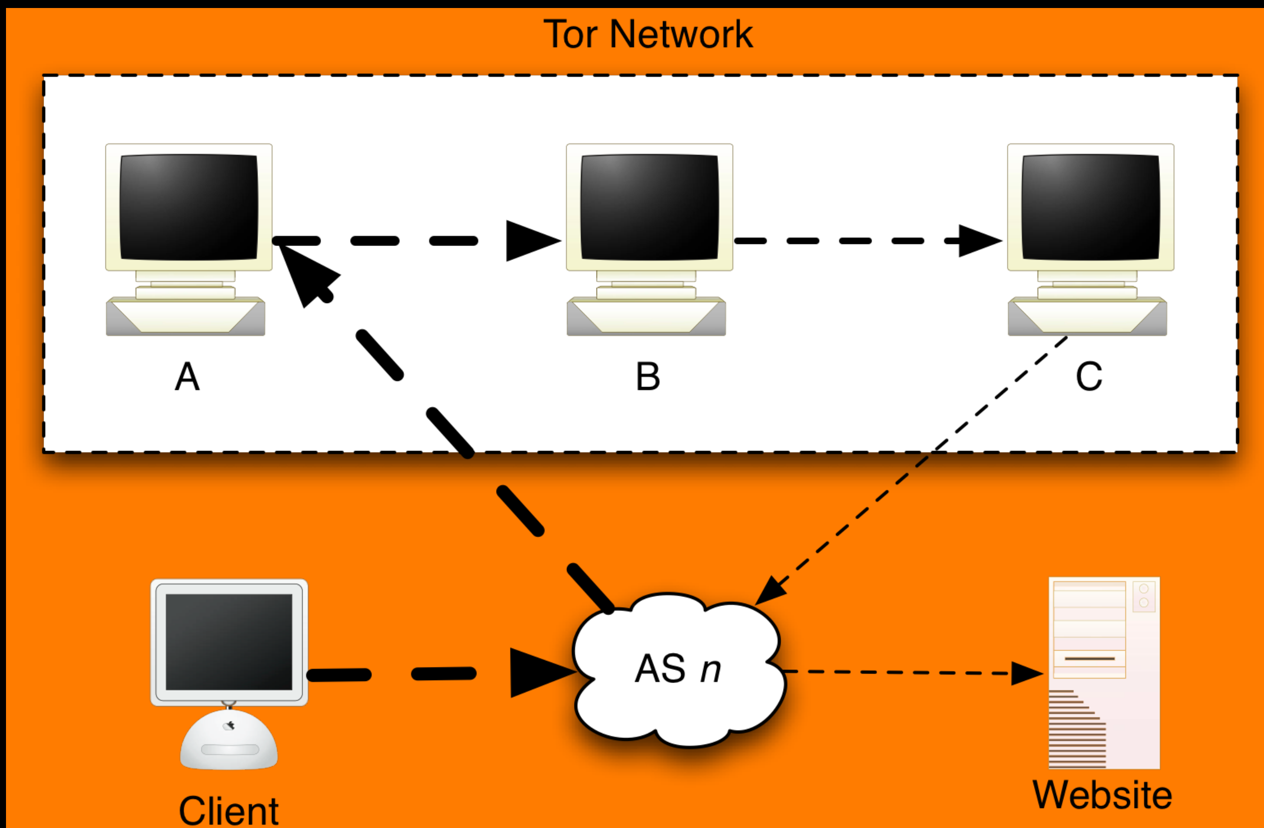


Links have structure



Network routing paths often traverse multiple ASes

AS-level observers



Previous Work

- Feamster & Dingledine (2004)
  - First analyzed the threat of AS-level observers against the Tor and Mixminion networks
  - Conducted when Tor was still in its infancy
- Murdoch & Zielinski (2007)
  - Further considered the threat of IXes against Tor clients in the UK
  - Used same list of destinations as FD04

Our Contributions

Validate previous results using an improved path selection algorithm

Examine how Tor's evolution has affected its resilience to AS-level observers

Provide a model of typical client and destination ASes on the current Tor network

Propose and evaluate several simple "AS-aware" path selection algorithms

Link attacks overview

Background

AS Path Inference

Analysis of Tor network growth

Tor AS statistics

Proposed path selection heuristics

AS Path Inference

Tries to predict route packets will take on the Internet

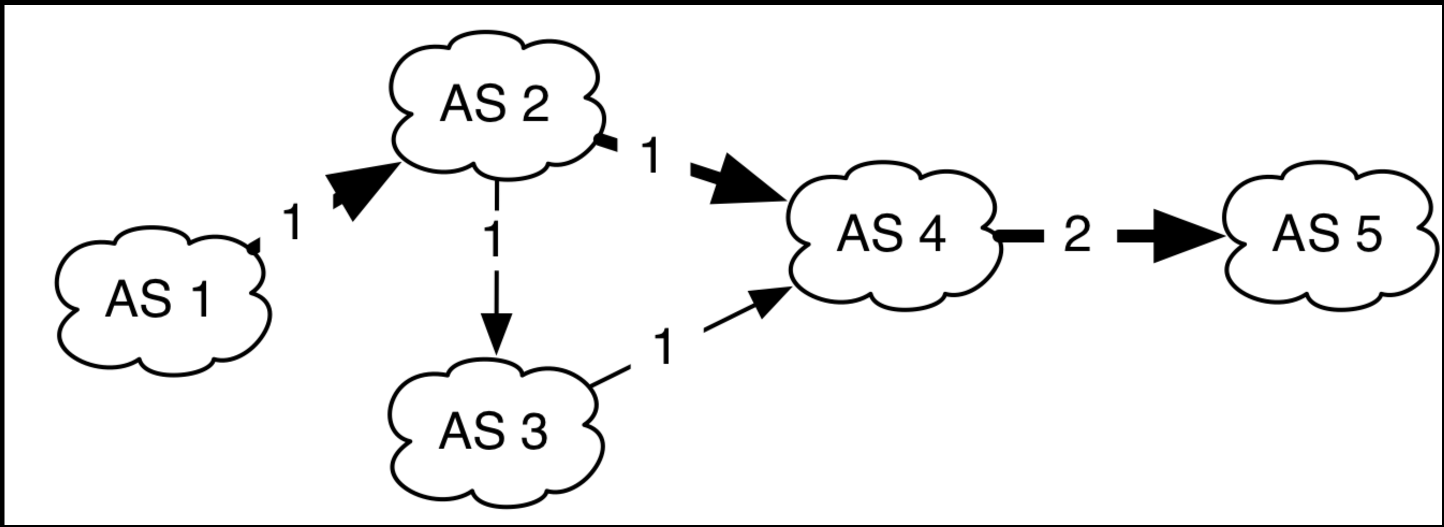
We do not have access to routing tables for the entire Internet

We cannot traceroute from arbitrary hosts

AS relationships are not often publicized for contractual reasons

AS Path Inference

Deriving AS Paths from Known Paths (Qiu & Gao 2006)



{1,2,3}, {2,4,5} and {3,4,5} are *known* paths

{1,2,4,5} is a *derived* path (must satisfy *valley-free* property)

AS Path Inference

- Used input routing tables from multiple Internet vantage points
  - OIX, Equinix, PAIX, KIXP, LINX, DIXIE
  - 1.47 GB, 15.7 million paths, 29,000 ASes, 132,000 edges
- Implementation
  - Implemented in C
  - Used Gao's (2000) algorithm for relationship inference
  - Modified slightly for better parallelization
  - All experiments done on a commodity Dell workstation

Outline

Background

AS Path Inference

Analysis of Tor network growth

Tor AS statistics

Proposed path selection heuristics

Conclusions & future work

Tor Grows Up

	June 2004 (33 relays)						**September 2008** (1239–1303 relays)					
Sender	2914	11643	12182	15130	15169	26101	2914	11643	12182	15130	15169	26101
209	0.49	0.45	0.40	0.39	0.19	0.30	0.17	0.26	0.19	0.51	0.23	0.25
1668	0.39	0.24	0.30	0.30	0.19	0.32	0.18	0.23	0.20	0.25	0.13	0.16
4355	0.38	0.27	0.28	0.27	0.43	0.51	0.13	0.29	0.12	0.20	0.19	0.14
6079	0.62	0.45	0.48	0.24	0.43	0.71	0.12	0.30	0.15	0.22	0.20	0.17
18566	0.39	0.42	0.41	0.32	0.56	0.73	0.18	0.36	0.20	0.31	0.20	0.16
22773	0.56	0.35	0.37	0.21	0.34	0.54	0.21	0.14	0.20	0.20	0.17	0.19
22909	0.21	0.24	0.26	0.22	0.22	0.37	0.19	0.30	0.24	0.25	0.21	0.19
23504	0.39	0.29	0.37	0.33	0.42	0.54	0.49	0.22	0.23	0.19	0.16	0.12

Used 3 separate Tor consensus snapshots from September 2008

Mean overall probability of an AS-level observer decreased from 37.74% to 21.86%

Tor Grows Up

	June 2004 (33 relays)						September 2008 (1239–1303 relays)					
Sender	2914	11643	12182	15130	15169	26101	2914	11643	12182	15130	15169	26101
209	0.49	0.45	0.40	0.39	0.19	0.30	0.17	0.26	0.19	0.51	0.23	0.25
1668	0.39	0.24	0.30	0.30	0.19	0.32	0.18	0.23	0.20	0.25	0.13	0.16
4355	0.38	0.27	0.28	0.27	0.43	0.51	0.13	0.29	0.12	0.20	0.19	0.14
6079	0.62	0.45	0.48	0.24	0.43	0.71	0.12	0.30	0.15	0.22	0.20	0.17
18566	0.39	0.42	0.41	0.32	0.56	0.73	0.18	0.36	0.20	0.31	0.20	0.16
22773	0.56	0.35	0.37	0.21	0.34	0.54	0.21	0.14	0.20	0.20	0.17	0.19
22909	0.21	0.24	0.26	0.22	0.22	0.37	0.19	0.30	0.24	0.25	0.21	0.19
23504	0.39	0.29	0.37	0.33	0.42	0.54	0.49	0.22	0.23	0.19	0.16	0.12

Used 3 separate Tor consensus snapshots from September 2008

Mean overall probability of an AS-level observer decreased from 37.74% to 21.86%

≈12.5% AS pairs were worse off than before

Link attacks overview

Background

AS Path Inference

Analysis of Tor network growth

Tor AS statistics

Proposed path selection heuristics

Tor AS Distribution Model

Data Collection

Ran two relays for 7 days in early September 2008

Mapped client and destination IP addresses to AS numbers

Kept only aggregated statistics at AS level

- *Never wrote IP addresses, timestamps or other metadata to disk*

Tor AS Distribution Model

Results

20638 client connections

- 2251 distinct ASes
- 85% produced fewer than 10 connections
- >50% produced only a single connection

116781 destination connections

- 4203 distinct ASes
- 72% produced fewer than 10 connections
- 34% had only a single connection

Tor Client AS Distribution

Rank	#	CC	Description
1	2238	DE	Deutsche Telekom AG
2	701	CN	ChinaNet
3	672	EU	Arcor
4	576	IT	Telecom Italia
5	566	DE	HanseNet Telekommunikation
6	429	DE	Telefonia Deutschland
7	280	FR	Proxad
8	279	US	AT&T Internet Services
9	276	CN	CNC Group Backbone
10	272	TR	TTNet

Tor Destination AS Distribution

Rank	#	CC	Description
1	5203	CN	ChinaNet
2	**4960**	**US**	**Google Inc.**
3	3527	NL	NForce Entertainment
4	2824	TW	HiNet
5	**2085**	**US**	**AOL**
6	2029	US	ThePlanet.com
7	1530	CN	CNC Group Backbone
8	1104	CN	CNC Group Beijing Province
9	1083	US	Level3 Communications
10	1011	NL	LeaseWeb

Link attacks overview

Background

AS Path Inference

Analysis of Tor network growth

Tor AS statistics

AS-aware path selection algorithms

Tor Path Selection Changes

- Weighted node selection
  - Relay bandwidth
  - Uptime
- Entry guards
- Distinct /16 subnets

Tor Path Selection Changes

Effectiveness of Distinct /16 Subnets

- Using mid-September Tor consensus
  - 876/1238 (≈70%) relays in same AS as at least one other relay, but in distinct /16 subnets
  - 850/1238 (≈68.7%) in same AS but distinct /8 subnet
- Generated 15,000 paths using Tor's algorithm
  - 1 out of every 133 paths contained entry and exit node in same AS but distinct /16 subnet
  - All but four also in distinct /8 subnets

Proposed Path Selection Algorithms

Unique Relay Countries (Unique-CC)

- Do not permit multiple relays from the same country in a single circuit
- Easy to implement with current Tor software
- Has been informally suggested or requested on Tor mailing list

Proposed Path Selection Algorithms

Unique Relay ASes (Unique-AS)

- Do not permit multiple relays from the same AS in a single circuit
- Requires clients or directory authorities to map a relay to an origin AS
- Tor Proposal #144

Proposed Path Selection Algorithms

Approximate AS Paths

- Directory authorities generate and distribute AS graph snapshot and prefix table files

Prior to building a circuit, clients can

1. Map self, entry node, exit node, destination to ASes in the topology
2. Compute shortest length *valley-free* paths from
   Client to entry node (and reverse)
   Exit node to destination (and reverse)
3. Sort in descending order by frequency value
4. Compare the top *n* paths for intersections

Testing AS-aware routing
Results Summary

Used same 3 consensus snapshots from Sept. 2008

Generated 5,000 Tor circuits per snapshot per algorithm

	Forward	Reverse	**Total**
Uniform	12.79%	13.23%	20.49%
Weighted (Tor)	10.92%	11.14%	17.81%
Unique-CC	10.41%	11.24%	17.61%
Unique-AS	10.07%	10.14%	16.73%
Approx. AS Path ($n = 1$)	*6.29%*	*6.01%*	*11.09%*
Approx. AS Path ($n = 3$)	*3.17%*	*3.34%*	*6.23%*

Questions raised today

How do we know how to choose entry nodes in Tor paths (to avoid correlation, predecessor and other attacks)?

We just looked at avoiding a single common link (AS) on both sides of a Tor connection. But, what if an adversary is able to observe some links but not others? What if he can observe multiple links?

These suggest an idea of using trust values in the nodes and links to reduce the threat of correlation from both nodes and links?

Adding trust to onion routing

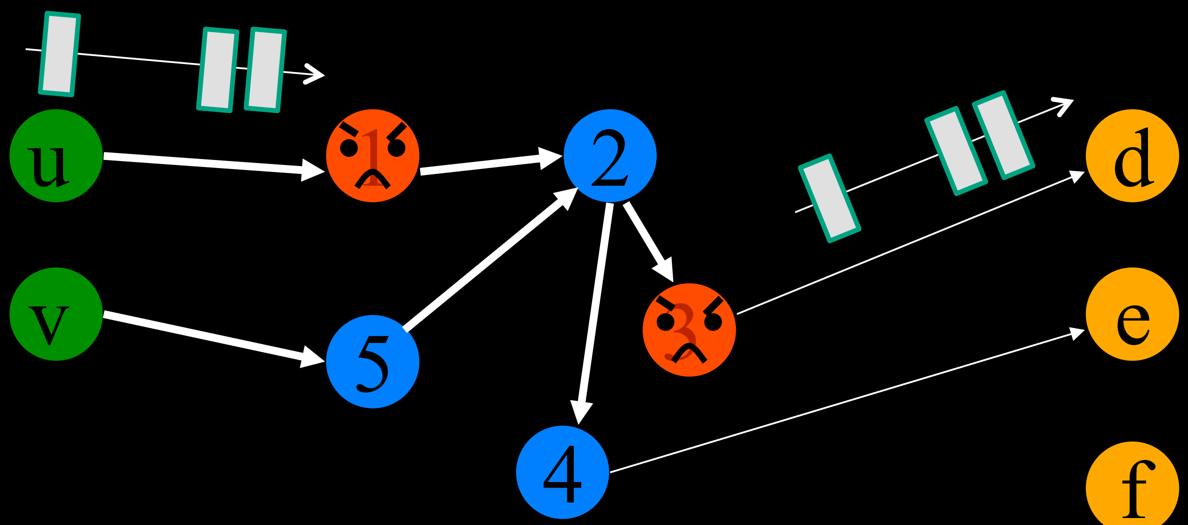
Assume that nodes are trusted to different degrees.

Simplest question to ask first: How can we choose the first and last node in an onion routing circuit to minimize the chance of a correlation attack?

- i.e. minimize the chance that they are both compromised

Adding trust in links, association of a user with the nodes he trust... can come later, but are pointless if we cannot handle this most basic question.

Use trust to minimize risk of end-to-end correlation attack



Some adversarial routers

User doesn't know where the adversary is.

User may have some idea of which routers are likely to be adversarial.

Model

Router r_i has **trust** t_i. An attempt to compromise a router succeeds with probability $c_i = 1-t_i$.

User will choose circuits using a known distribution.

Adversary attempts to compromise at most k routers, $K \subseteq R$.

After attempts, users actually choose circuits.

Model

For anonymity, minimize correlation attack

Probability of compromise:

$$c(p,K) = \Sigma_{r,s \in K}\, p_{rs}\, c_r\, c_s$$

Problem:

Input: Trust values $t_1,\ldots,t_n$

Output: Distribution p^* on router pairs such that

$$p^* \in \text{argmin}_p \max_{K \subseteq R:|K|=k} c(p,K)$$

Algorithm

Turn into a linear program

Variables: p_{rs} $\forall r,s \in R$

t (slack variable)

Constraints:

Probability distribution:

$0 \leq p_{rs} \leq 1$

$\Sigma_{r,s \in R}\, p_{rs} = 1$

Minimax:

$\mathrm{t} - c(p,K) \geq 0 \;\; \forall\, K \subseteq R{:}|K|=k$

Objective function : t

Algorithm

Turn into a linear program

Variables: p_{rs} $\forall r,s \in R$

t (slack variable)

Constraints:

Probability distribution:

$0 \le p_{rs} \le 1$

$\Sigma_{r,s \in R}\, p_{rs} = 1$

Minimax:

$\mathrm{t} - c(p,K) \ge 0 \quad \forall\, K \subseteq R : |K| = k$

Objective function : t

Problem: Exponential-size linear program

Next Attempt: Use Independent-Choice Approximation (instead of pairs)

Let $c(p) = \max_{K \subseteq R : |K| = k} \Sigma_{r \in K}\, p_r\, c_r$.

Choose routers independently using

$$p^* \in \operatorname{argmin}_p c(p)$$

Independent-Choice Approximation

Let $c(p) = \max_{K \subseteq R:|K|=k} \Sigma_{r \in K}\, p_r\, c_r$.

Choose routers independently using

$$p^* \in \text{argmin}_p\, c(p)$$

Let $\mu = \text{argmin}_i\, c_i$.

Let $p^1(r_\mu) = 1$.

Let $p^2(r_i) = \alpha / c_i$, where $\alpha = (\Sigma_i\, 1/c_i)^{-1}$.

Theorem:

$$c(p^*) = \begin{cases} c(p^1) \text{ if } c_\mu \leq k\alpha \\ c(p^2) \text{ otherwise} \end{cases}$$

Independent-Choice Approximation

Question: How close an approximation to choosing nodes that minimize first-last pair compromise is it to choose the first and last nodes independently minimizing the chance that each is compromised?

Answer: Not very. Approximation error is arbitrarily bad.

Theorem: The approximation ratio of independent selection is $\Omega(\sqrt{n})$.

Next try, limit the number of trust levels.

Most users unlikely to have a meaningful arbitrarily fine gradation of trust in all nodes in the network.

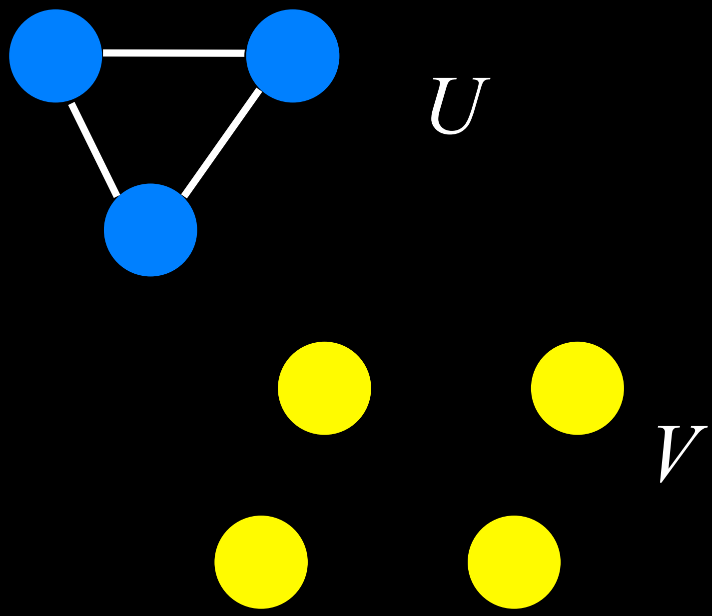
Suppose users have just two levels of trust reflecting essentially

- Those nodes they have particular reason to trust (e.g., part of a coalition)
- Those they don't

Trust Model

Two trust levels: $t_1 \geq t_2$

$U = \{r_i \mid t_i = t_1\}$, $V = \{r_i \mid t_i = t_2\}$

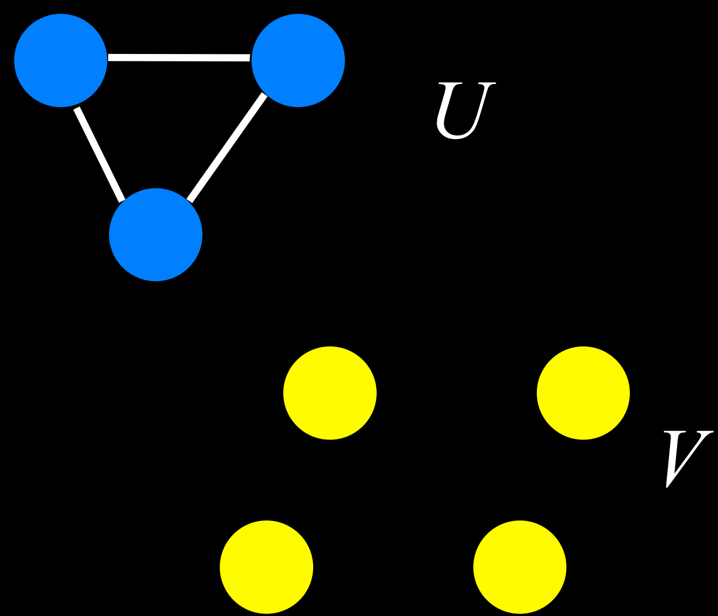


Trust Model

Two trust levels: $t_1 \geq t_2$

U = $\{r_i \mid t_i = t_1\}$, V = $\{r_i \mid t_i = t_2\}$

Theorem: Three distributions can be optimal:



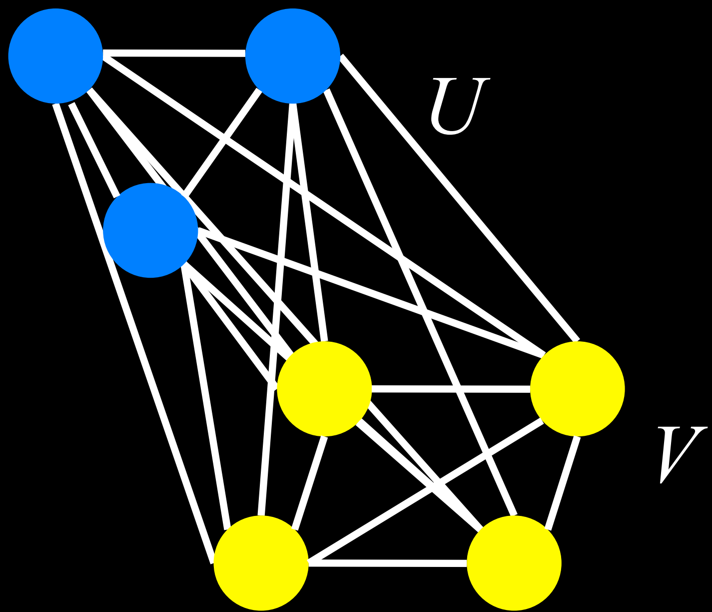
Trust Model

Two trust levels: $t_1 \geq t_2$

U = $\{r_i \mid t_i = t_1\}$, V = $\{r_i \mid t_i = t_2\}$

Theorem: Three distributions can be optimal:

1. $p(r,s) \propto c_r c_s$ for $r,s \in R$



Trust Model

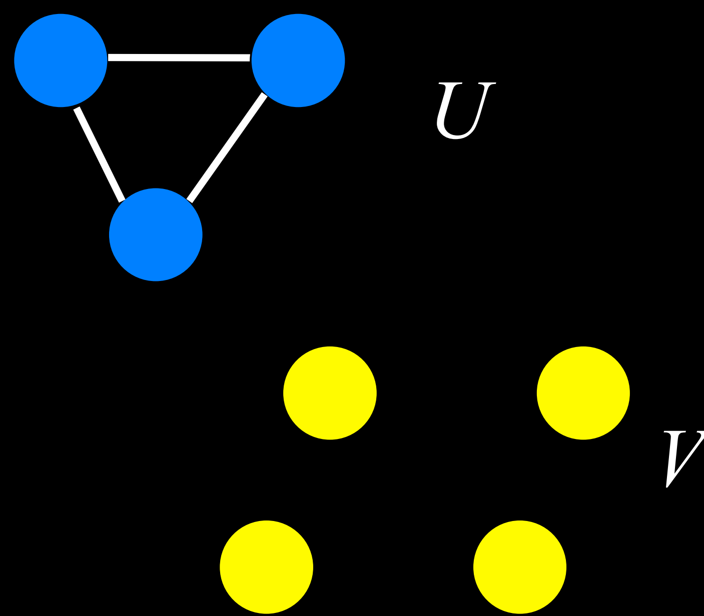
Two trust levels: $t_1 \geq t_2$

$U = \{r_i \mid t_i = t_1\}$, $V = \{r_i \mid t_i = t_2\}$

Theorem: Three distributions can be optimal:

1. $p(r,s) \propto c_r c_s$ for $r,s \in R$

2. $p(r,s) \propto \begin{array}{ll} c_1^{\,2} & \text{if } r,s \in U \\ 0 & \text{otherwise} \end{array}$



Trust Model

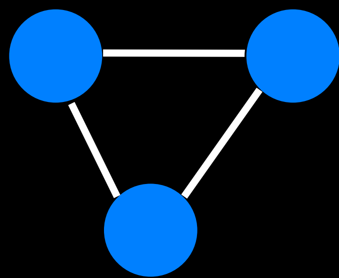
Two trust levels: $t_1 \geq t_2$

$U = \{r_i \mid t_i = t_1\}$, $V = \{r_i \mid t_i = t_2\}$

Theorem: Three distributions can be optimal:

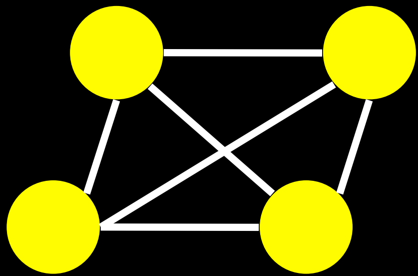
1. $p(r,s) \propto c_r c_s$ for $r,s \in R$

2. $p(r,s) \propto \begin{cases} c_1^2 & \text{if } r,s \in U \\ 0 & \text{otherwise} \end{cases}$



U

3. $p(r,s) \propto \begin{cases} c_1^2(n(n-1)-v_0(v_0-1)) & \text{if } r,s \in U \\ c_2^2(m(m-1)-v_1(v_1-1)) & \text{if } r,s \in V \\ 0 & \text{otherwise} \end{cases}$



V

Generalization and Other Applications

Pick a subset of size *j*

Minimize the chance that all are compromised

Examples:

- Heterogenous sensor networks
- Distributed computation (*e.g.* SETI@home)
- Data integrity in routing

Future Work

- Generalization to other problems
- Heterogeneous trust
  - Users choose paths differently
  - User profiling
  - Adversary may not know trust values
- Roving adversary

Next steps

- Expand adversary model of diverse trust in routing security beyond above correlating adversary
  - Fingerprinting, Trust learning, Adversary learning
- Devise routing strategies for new model
- Incorporate links into adversary model
- Design trust aware network info distribution
- Analysis and simulations of performance/security tradeoffs
- ....

Questions?

Practice saying this while you think of some:
Donna Compagna mangia banane con pane e con panna in compagnia di campane in capanna nelle campagne della Campania.