Anonymized Networks, Hidden Patterns, and Privacy Breaches

Jon Kleinberg

Cornell University

Joint work with Lars Backstrom (Cornell) and Cynthia Dwork (Microsoft)

The Perils of Anonymized Data

Research on social networks: public vs. sensitive data

- Public data: Web pages, blogs, discussion boards, Wikipedia, open social networking sites.
- Sensitive: E-mail, IM, voice, physical proximity.
  E.g. nodes are e-mail or IM accounts;
  edge (v, w) if v communicates with w.

Anonymization of sensitive data:

- Consider research focused on structure and dynamics, not node identities.
- To anonymize: replace node names with random IDs.
- After doing this, is it safe to release?

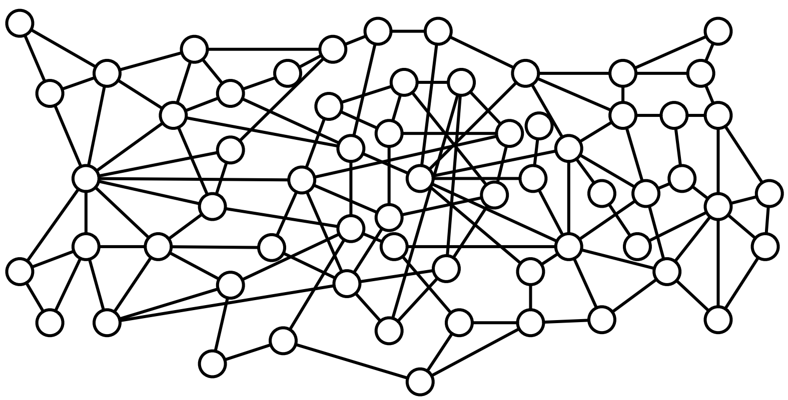
An Attack

With more detailed data, anonymization has run into trouble:

- Identifying on-line pseudonyms by textual analysis [Novak-Raghavan-Tomkins 2004]
- De-anonymizing Netflix ratings via time series [Narayanan-Shmatikov 2006]
- The AOL query logs [“This was a screw-up, and we’re angry and upset about it.” —AOL press release, 7 August 2006]

Our setting is much starker:

- No text, time-stamps, or node attributes
- Just a graph with nodes numbered $1, 2, 3, \ldots, n$.



Attacks on Anonymized Data

Analogy with passive vs. active attacks in cryptography

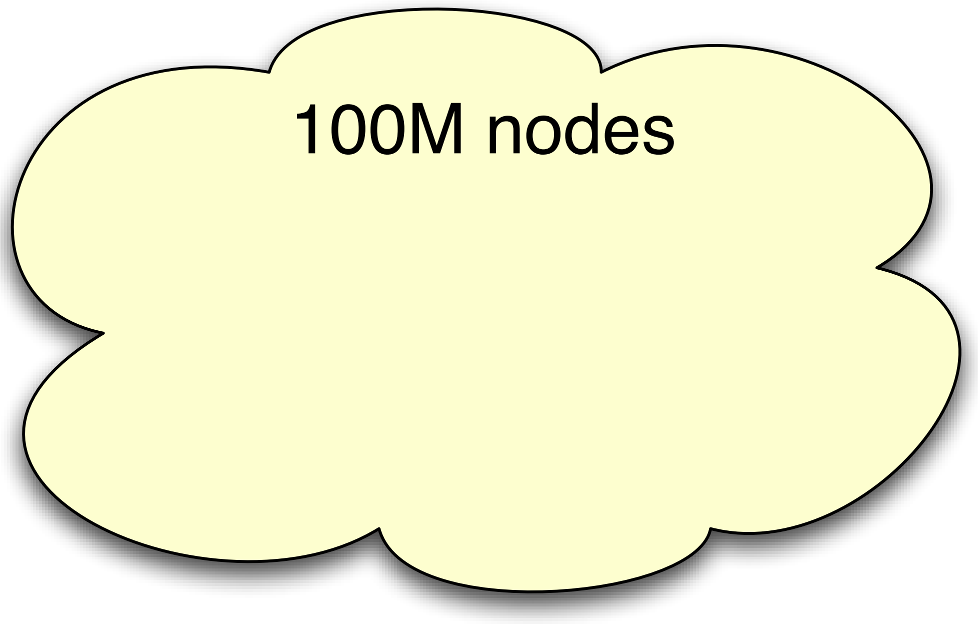
- Passive attack: observe data as it is presented.
- Active attack: insert yourself into the process, potentially causing additional data to be generated.

Template for an active attack on an anonymized network [Backstrom-Dwork-Kleinberg 2007]

- Attacker can create (before the data is released)
  - nodes (e.g. by registering an e-mail account)
  - edges incident to these nodes (by sending mail)
- Privacy breach: learning whether there is an edge between two existing nodes in the network.
- Note: attacker's actions are completely "innocuous."

Main result: active attacks can easily compromise privacy by creating very few additional nodes.

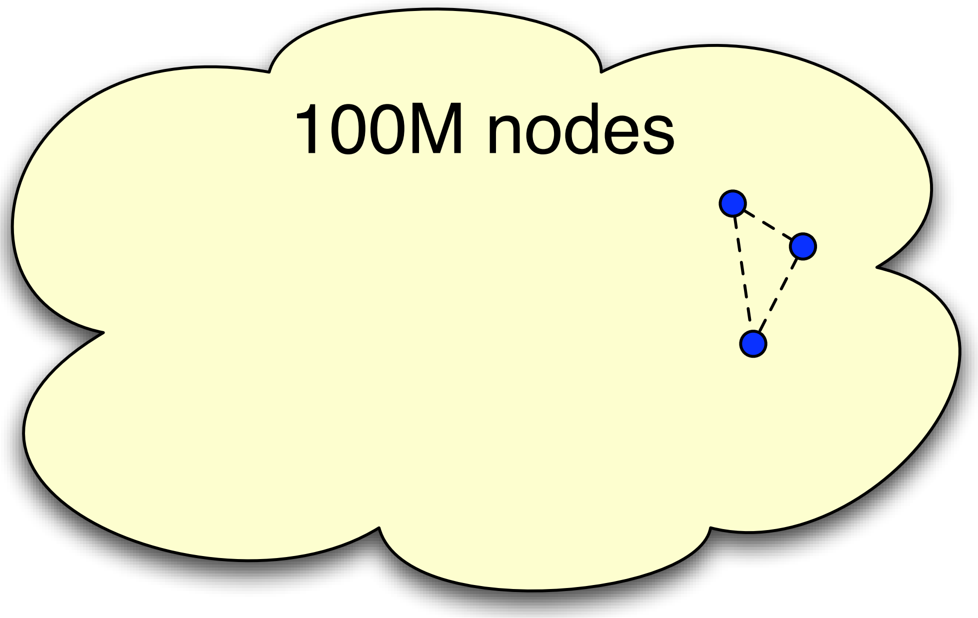
An Attack



Scenario:

Suppose a big company were going to release an anonymized communication graph on 100 million users.

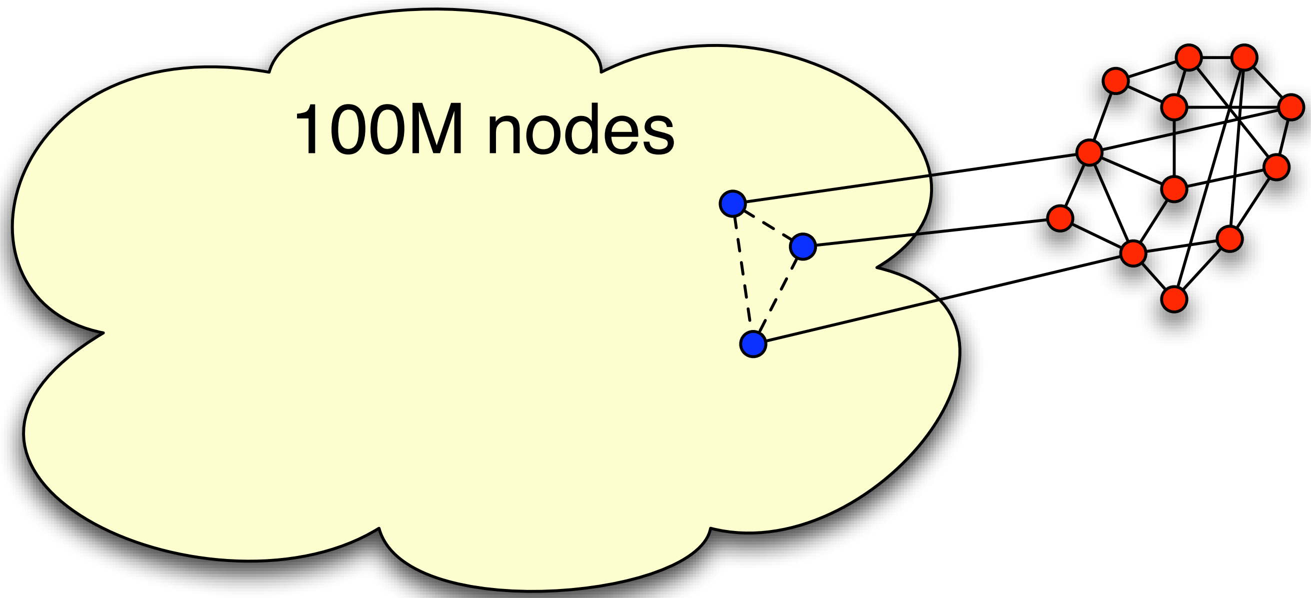
An Attack



An attacker chooses a small set of b user accounts to "target":

Goal is to learn edge relations among them.

An Attack

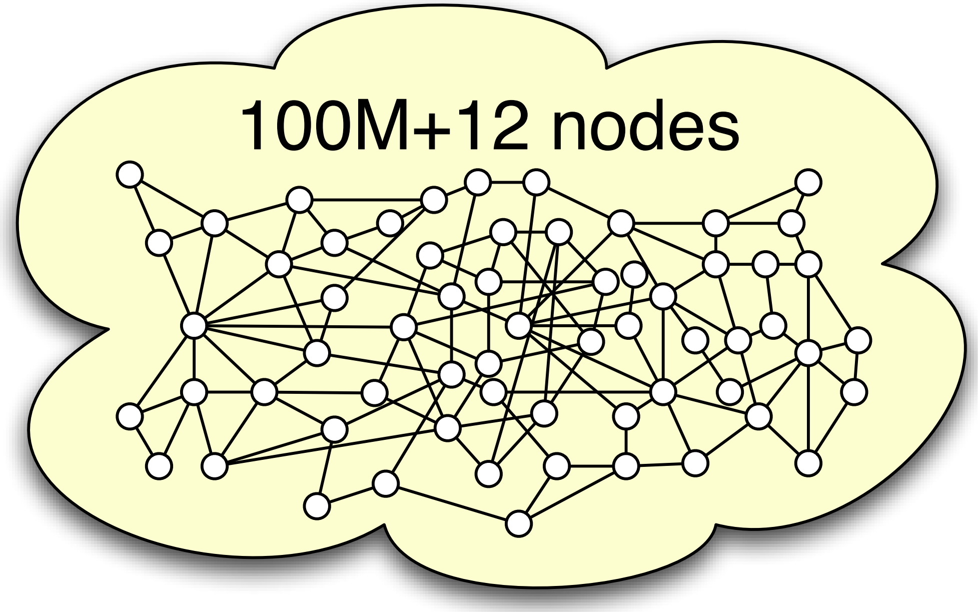


Before dataset is released:

Create a small set of k new accounts, with links among them, forming a subgraph H.

Attach this new subgraph H to targeted accounts.

An Attack

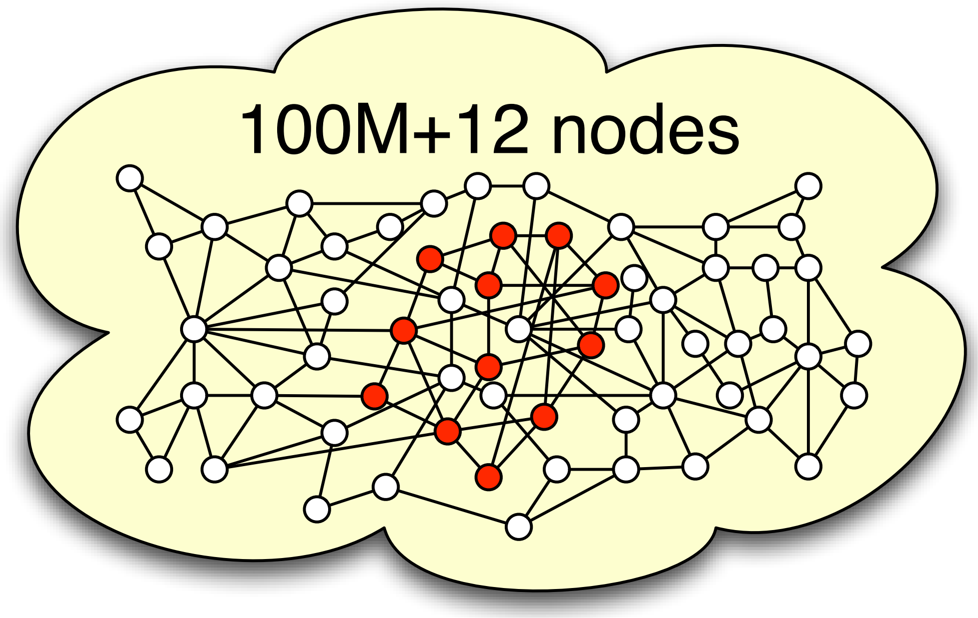


When anonymized dataset is released, need to find H.

Why couldn't there be many copies of H in the dataset?
(We don't even know what the network will look like ...)

Why wouldn't it be computationally hard to find H?

An Attack

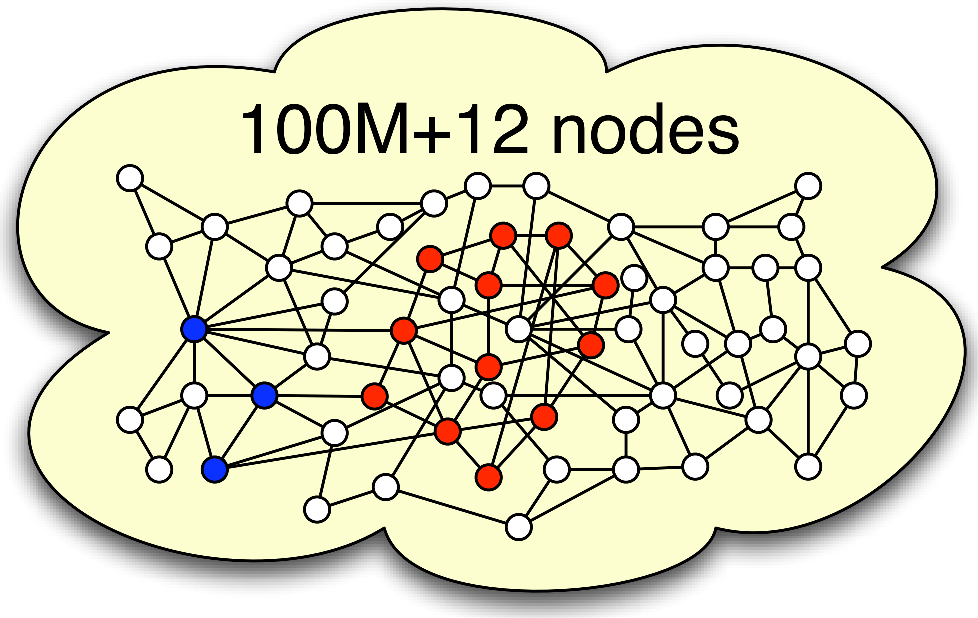


In fact,

Theorem: small random graphs H will likely be unique and efficiently findable.

Erdös-Rényi construction; each edge present with prob. $1/2$.

An Attack



Once H is found:

Can easily find the targeted nodes by following edges from H.

Specifics of the Attack

First version of the attack:

- Create random H on $(2 + \varepsilon) \log n$ nodes.
  Can compromise $\sim (\log n)^2$ targeted nodes.
- In experiments on 4.4 million-node LiveJournal graph, 7-node graph H can compromise 70 targeted nodes (and hence $\sim$ 2400 edge relations).

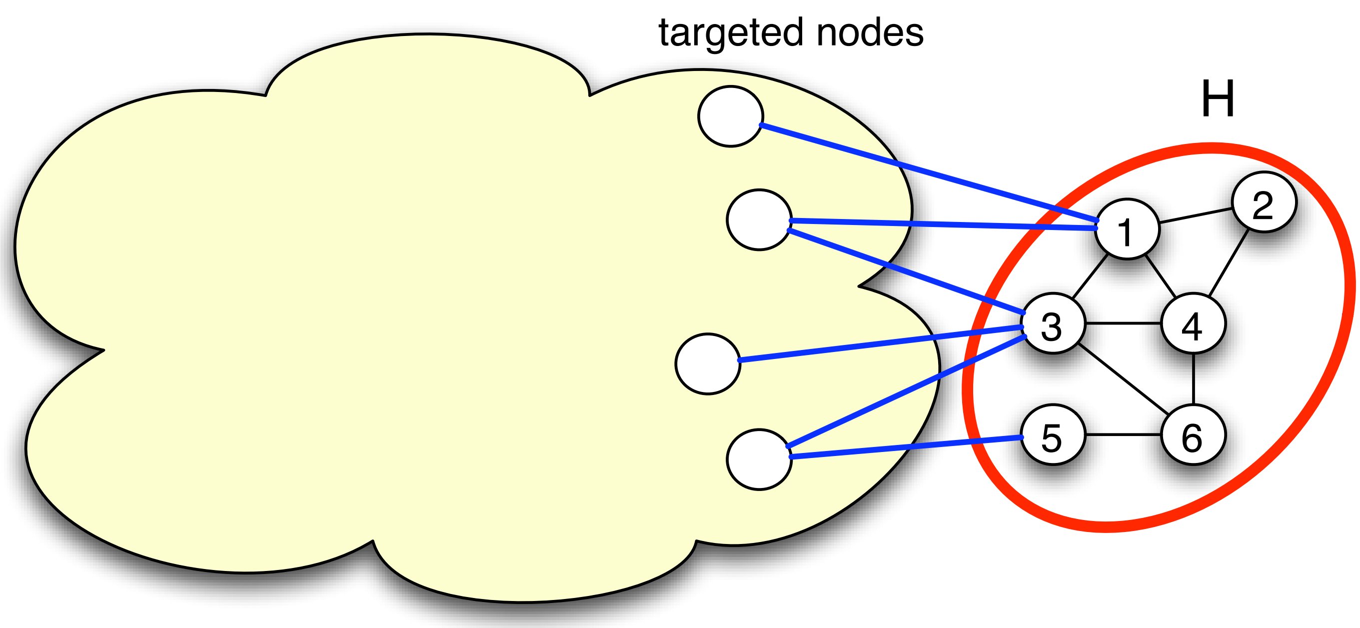
Second version of the attack:

- Logarithmic size is not optimal.
- Can begin breaching privacy with H of size $\sim \sqrt{\log n}$.

Passive attacks:

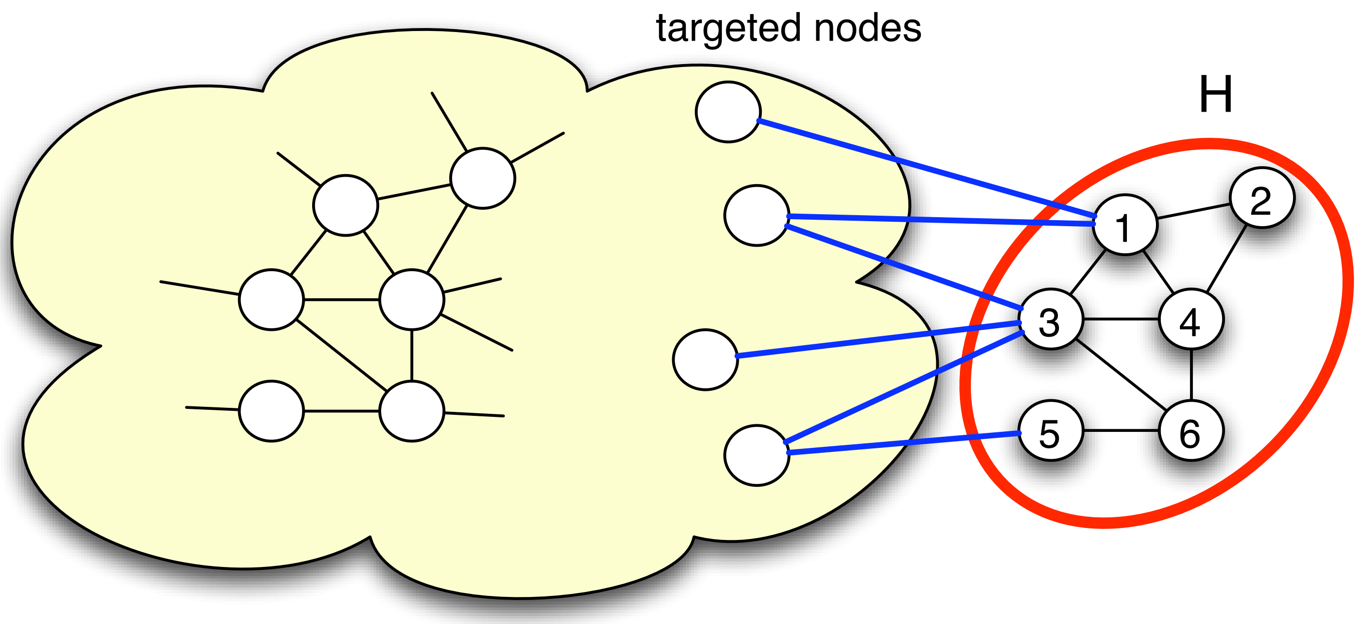
- In LiveJournal graph: with reasonable probability, you and 6 of your friends chosen at random can carry out the first attack, compromising about 10 users.

Specifics of the Attack



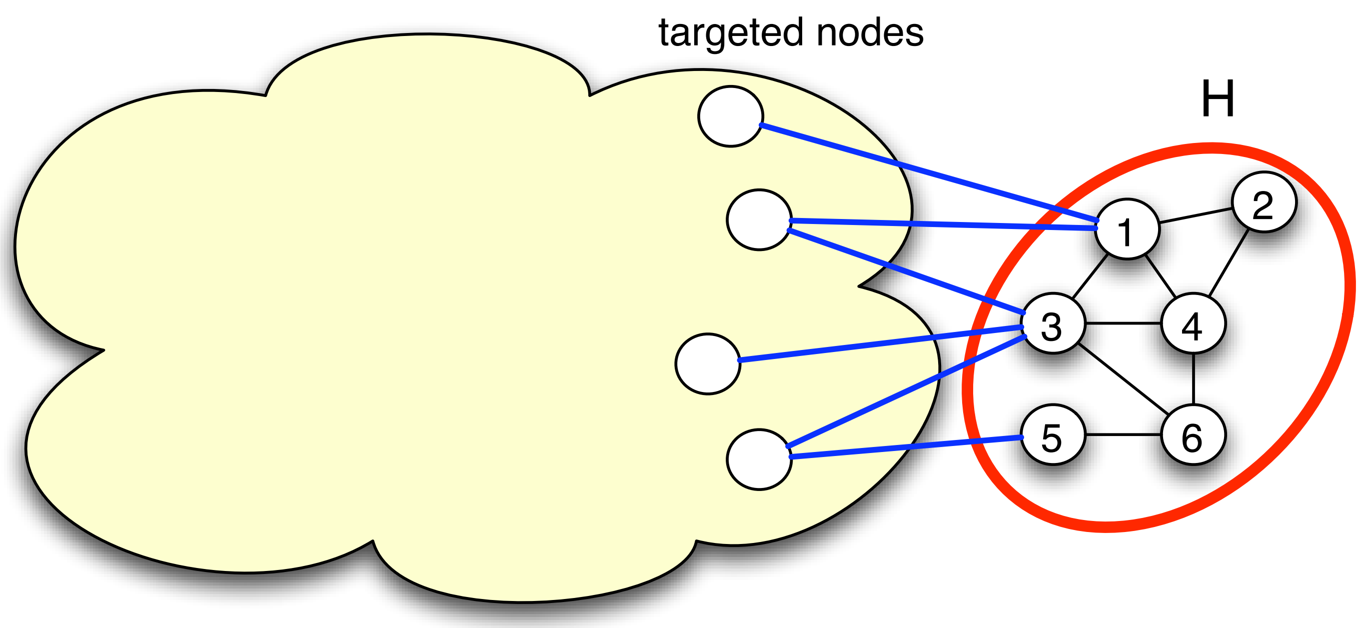
- Random subgraph H (each edge with prob. $\frac{1}{2}$).
- Link each targeted node to distinct subset of nodes in H.
- Must show
  - H is unique up to isomorphism (even after plugging it into rest of graph).
  - H is efficiently findable in unlabeled graph.
  - H has no internal symmetries (automorphisms); this is easy.

Specifics of the Attack



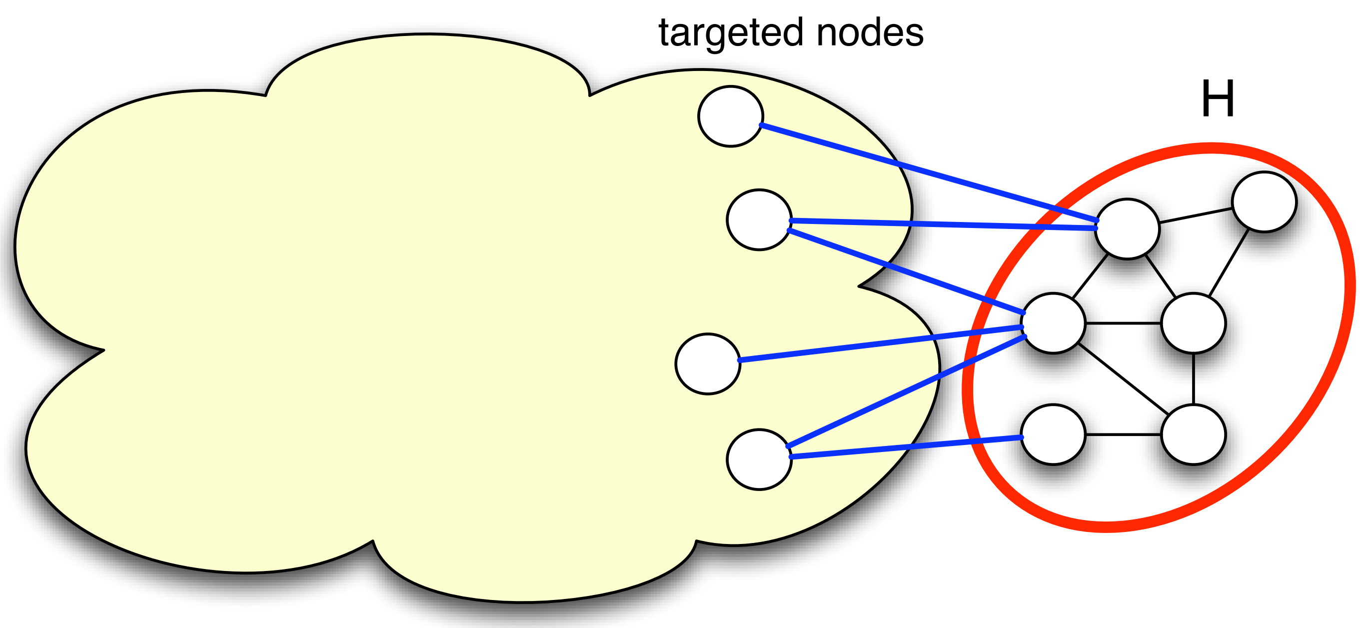
- Random subgraph H (each edge with prob. $\frac{1}{2}$).
- Link each targeted node to distinct subset of nodes in H.
- Must show
  - H is unique up to isomorphism (even after plugging it into rest of graph).
  - H is efficiently findable in unlabeled graph.
  - H has no internal symmetries (automorphisms); this is easy.

Specifics of the Attack



- Random subgraph H (each edge with prob. $\frac{1}{2}$).
- Link each targeted node to distinct subset of nodes in H.
- Must show
  - H is unique up to isomorphism (even after plugging it into rest of graph).
  - H is efficiently findable in unlabeled graph.
  - H has no internal symmetries (automorphisms); this is easy.

Specifics of the Attack

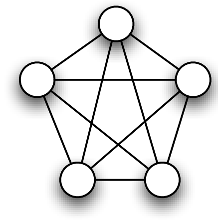


- Random subgraph H (each edge with prob. $\frac{1}{2}$).
- Link each targeted node to distinct subset of nodes in H.
- Must show
  - H is unique up to isomorphism (even after plugging it into rest of graph).
  - H is efficiently findable in unlabeled graph.
  - H has no internal symmetries (automorphisms); this is easy.

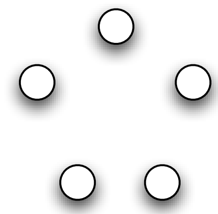
Why is H Unique? Ideas from Ramsey Theory

Basic calculation at the foundation of

- Theorem (Erdös, 1947): There exists an n-node graph with no clique and no independent set of size $> 2 \log n$.
- Quantitative bound for Ramsey's Theorem; one of the earliest uses of random graphs.



clique



independent set

The calculation:

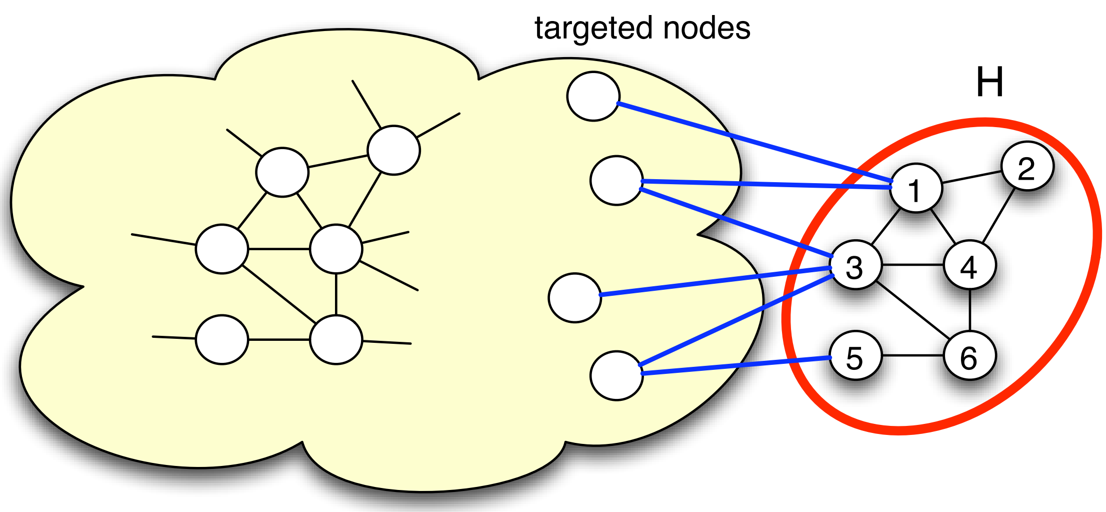
- Build random n-node graph, include each edge with prob. $\frac{1}{2}$.
- There are $< n^k$ sets of k nodes; each is a clique or independent set with probability $\approx 2^{-k^2/2}$.
- Product $n^k \cdot 2^{-k^2/2}$ upper-bounds probability of any clique or indep. set; it drops below 1 once k exceeds $\approx 2 \log n$.

Why is H Unique? Ideas from Ramsey Theory

Erdös: Graph is random, subgraph is non-random.
Our case: Subgraph (H) is random, graph is non-random.

But main calculation starts from same premise:

- Almost correct: there are $< n^k$ subgraphs that could be a second copy of H, and each is isomorphic to H with prob. $\approx 2^{-k^2/2}$.



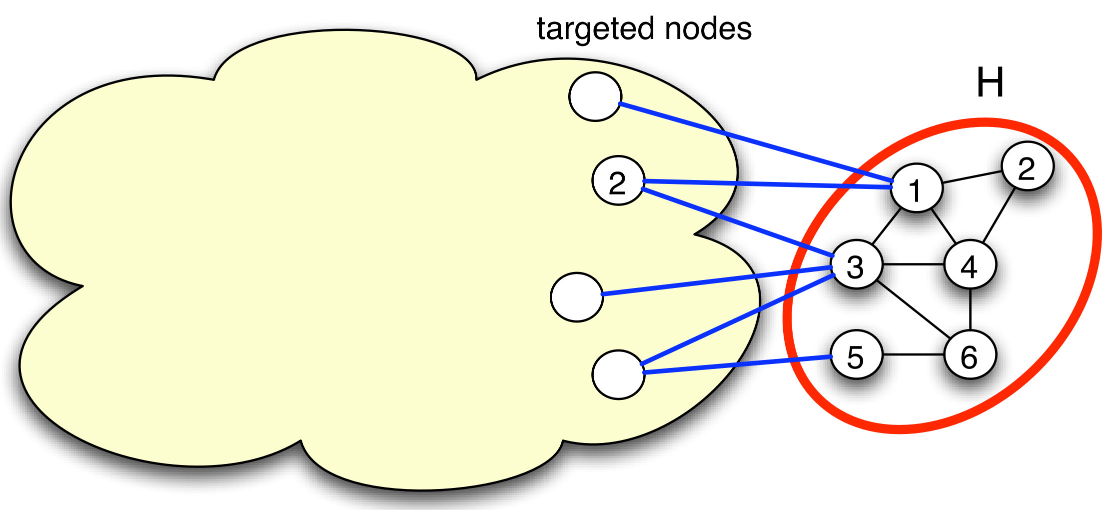
- Analysis is greatly complicated because H is plugged into full graph.
- New copies of H could partly overlap original copy of H.

Why is H Unique? Ideas from Ramsey Theory

Erdös: Graph is random, subgraph is non-random.
Our case: Subgraph (H) is random, graph is non-random.

But main calculation starts from same premise:

- Almost correct: there are $< n^k$ subgraphs that could be a second copy of H, and each is isomorphic to H with prob. $\approx 2^{-k^2/2}$.

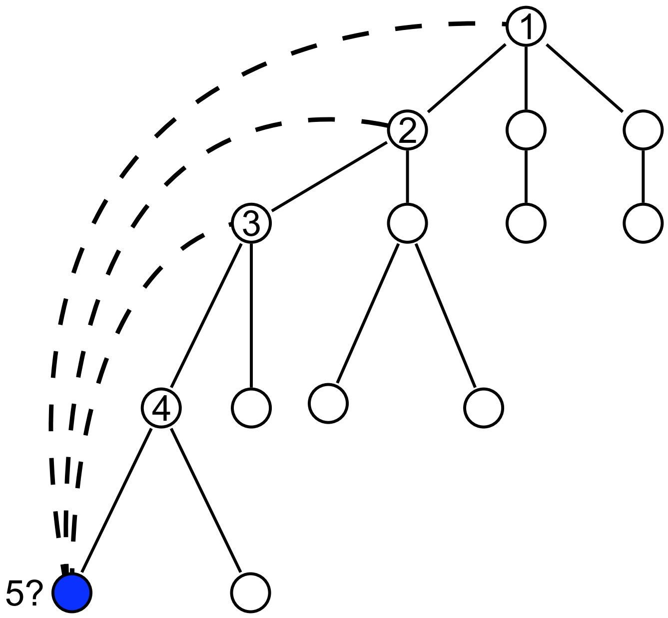


- Analysis is greatly complicated because H is plugged into full graph.
- New copies of H could partly overlap original copy of H.

Finding the subgraph H

To find H:

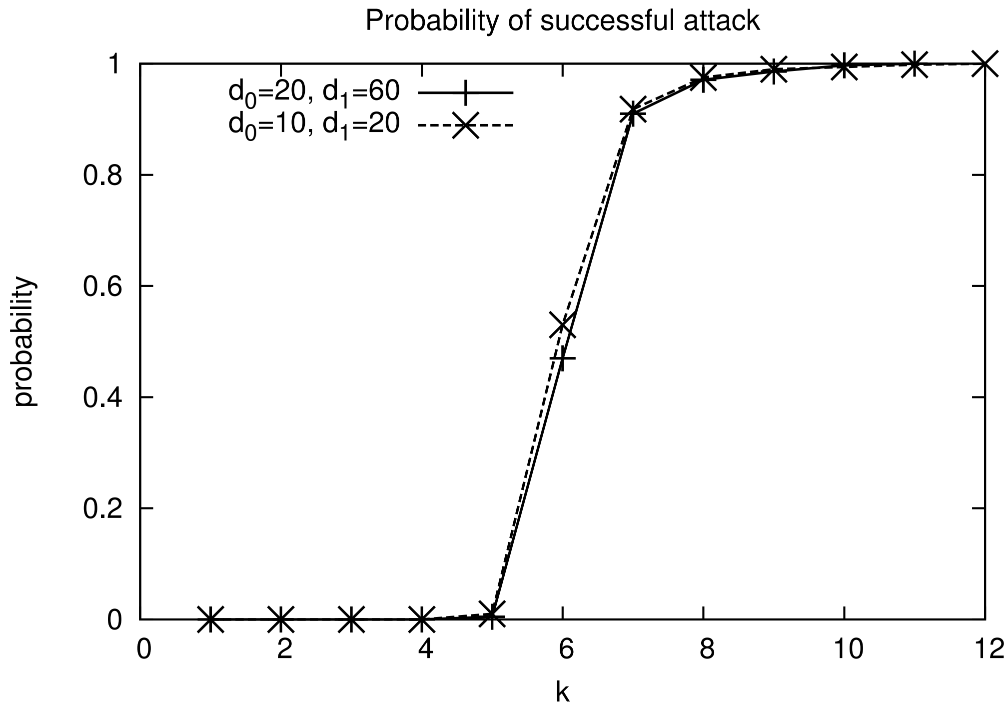
- Can assume there is a path through nodes $1, 2, \ldots, k$.
- Start search at all possible nodes in G.
- Prune search path at depth j if edges back from node j don't match, or if degree of j doesn't match.



- Probability of a spurious path surviving to depth j is $\approx 2^{-j^2/2}$ (modulo overlap worries).
- Overall size of search tree slightly more than linear in n.

Experiments

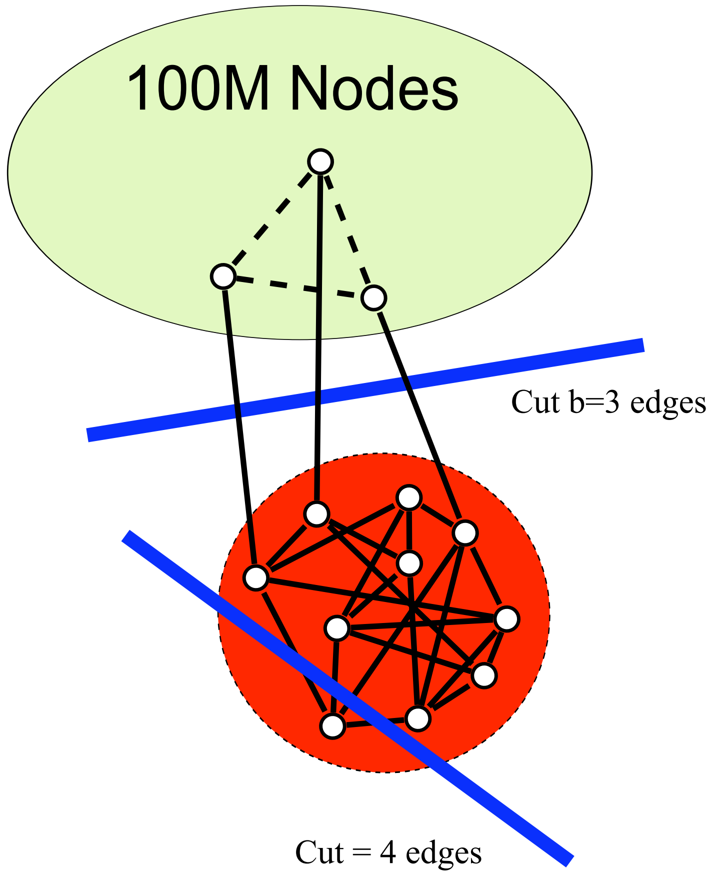
- Simulated the attack on social/blogging network LiveJournal
- 4.4 million nodes, 77 million edges



- With 7 nodes, degrees in $[20, 60]$, success rate $> 90\%$.
- Average of 70 nodes compromised (2415 edges).
- Search tree about 90,000 nodes; recovery time < 1 sec.
- 7 nodes much less than $2 \log n$; randomization of degrees crucial to performance.

Stronger Theoretical Bound

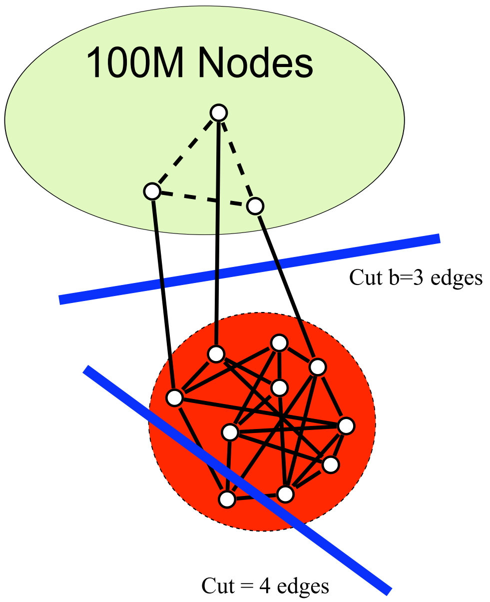
- Variant on construction breaches privacy with H of size $\sim \sqrt{\log n}$: Optimal up to constant factors.
- Construct H as before on k nodes, but connect to $b = \frac{k}{3}$ targeted nodes.
- With high prob., min. internal cut in H exceeds b = cut to rest of graph.



Stronger Theoretical Bound

Recovery:

- Break graph up along cuts of size $\leq b$. Uses Gomory-Hu tree computation (e.g. Flake et al. 2004)
- Can prove that H will be one of the components after this decomposition.

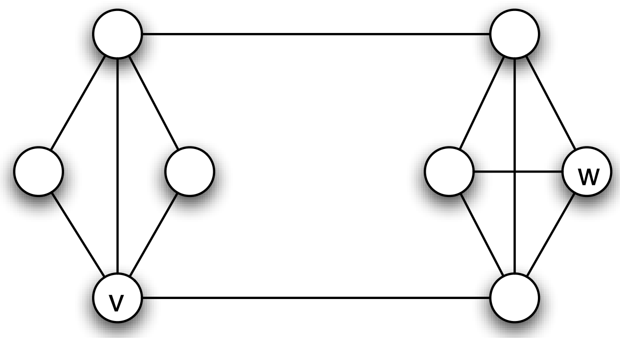


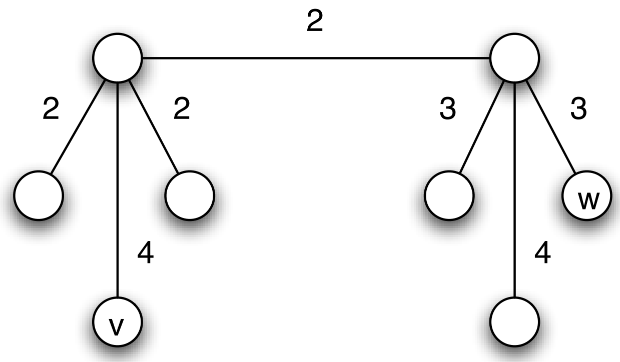
Uniqueness of H:

- After breaking apart the graph, there are $\leq \frac{n}{k}$ size-k components other than H.
- Each is isomorphic to H with probability $\approx 2^{-k^2/2}$.
- Now $2^{-k^2/2}$ only has to cancel $\frac{n}{k}$, not n^k, so $k \approx \sqrt{\log n}$ is enough.

Recovery: Gomory-Hu Trees

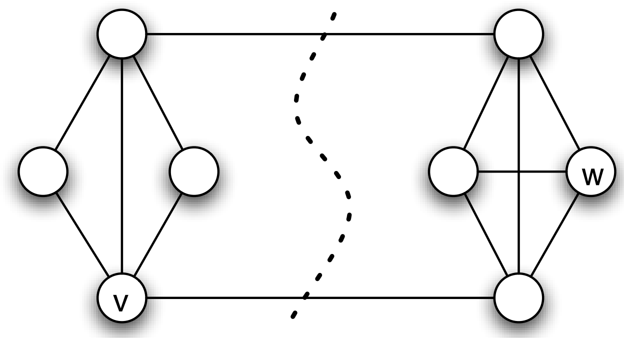
- Recovery: Break graph up along cuts of size $\leq b$.
- To do this, build Gomory-Hu tree:
  - Tree T with same node set as original graph.
  - To find min. v-w cut in graph, delete min-weight edge on v-w path in T.

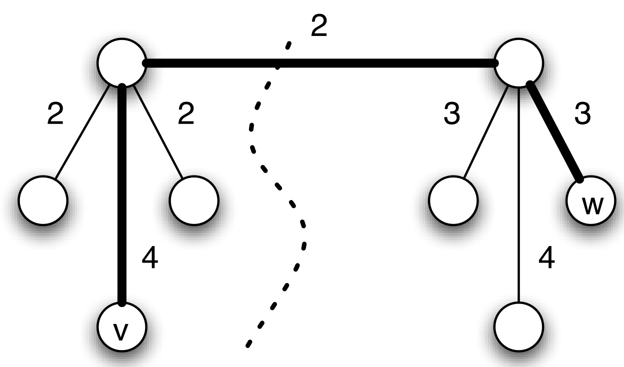




Recovery: Gomory-Hu Trees

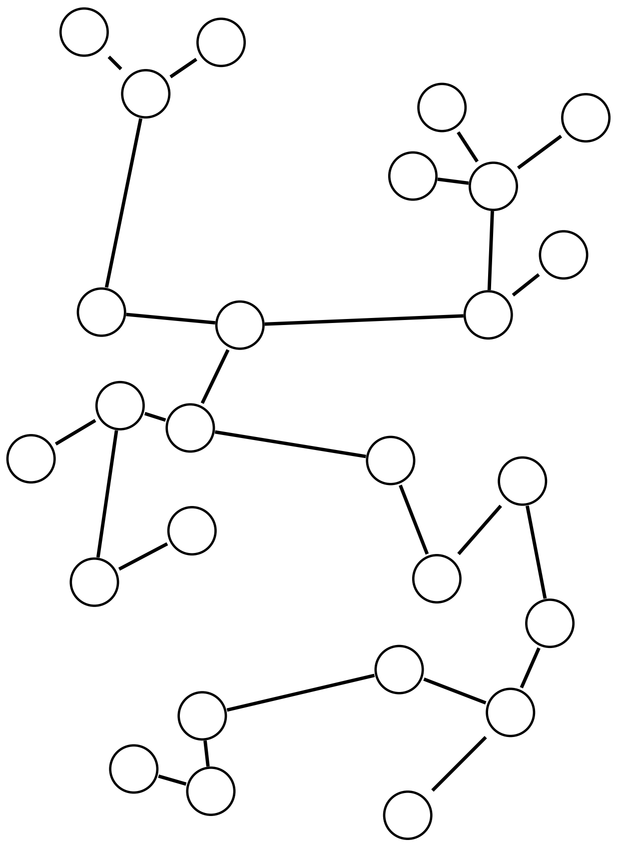
- Recovery: Break graph up along cuts of size $\leq b$.
- To do this, build Gomory-Hu tree:
  - Tree T with same node set as original graph.
  - To find min. v-w cut in graph, delete min-weight edge on v-w path in T.





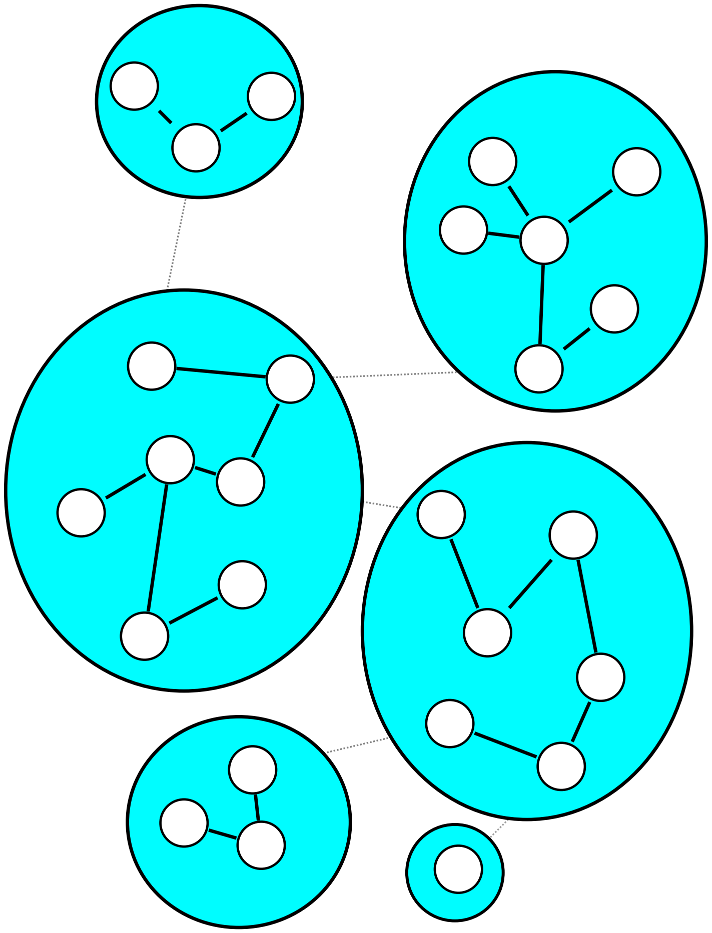
Recovery: Gomory-Hu Trees

- Recovery: Break graph up along cuts of size $\leq b$.
- To do this, build Gomory-Hu tree:
  - Tree T with same node set as original graph.
  - To find min. v-w cut in graph, delete min-weight edge on v-w path in T.



Recovery: Gomory-Hu Trees

- Recovery: Break graph up along cuts of size $\leq b$.
- To do this, build Gomory-Hu tree:
  - Tree T with same node set as original graph.
  - To find min. v-w cut in graph, delete min-weight edge on v-w path in T.

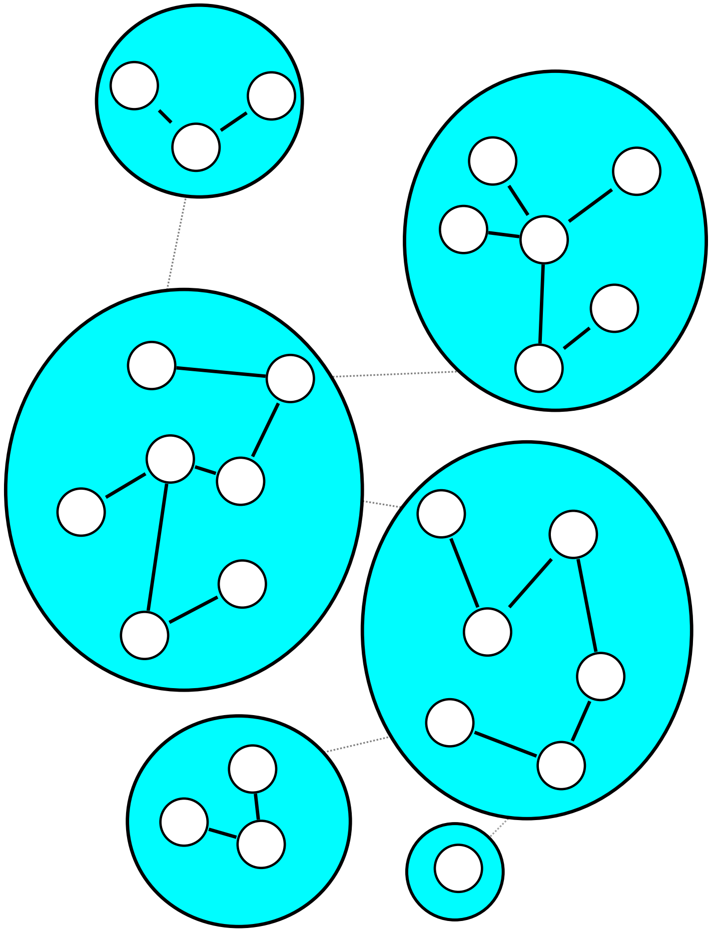


- To find H: delete all edges in T of weight $\leq b$.
- Can prove H will be one of the resulting components.

Recovery: Gomory-Hu Trees

Uniqueness of H:

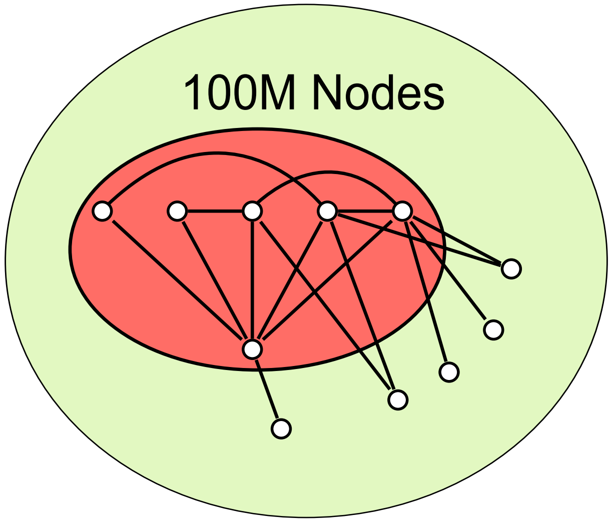
- After breaking apart the graph, there are $\leq \frac{n}{k}$ components of size k, other than H.
- Each is isomorphic to H with probability $\approx 2^{-k^2/2}$.
- Now just need $\frac{n}{k} \cdot 2^{-k^2/2} \ll 1$, so $k \approx \sqrt{\log n}$ is enough.



Passive Attacks

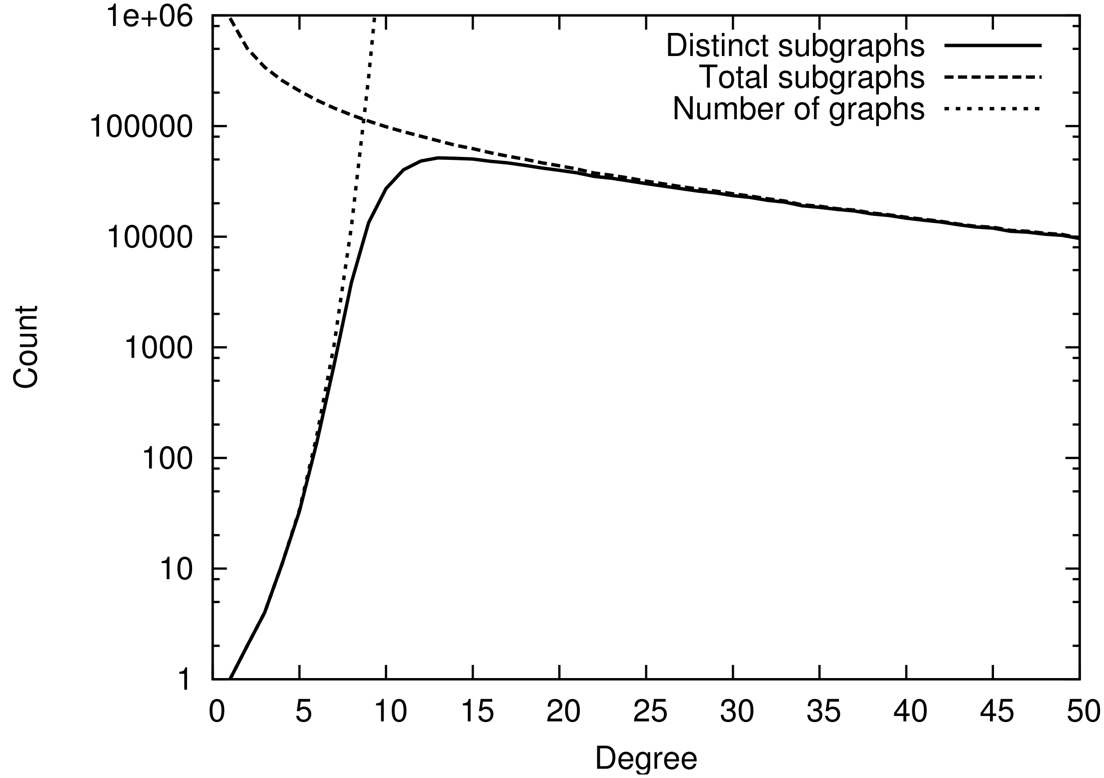
If you're already in the network, can you carry out this attack with no preparation?

- A node v recruits its neighbors.
- Suppose neighborhood subgraph $N(v)$ is unique (and efficiently findable).
- If a node w is the only one to attach to a particular subset of $N(v)$, then w is compromised.



What is the probability $N(v)$ is unique, as a function of its size?

Uniqueness of Neighborhood Subgraphs



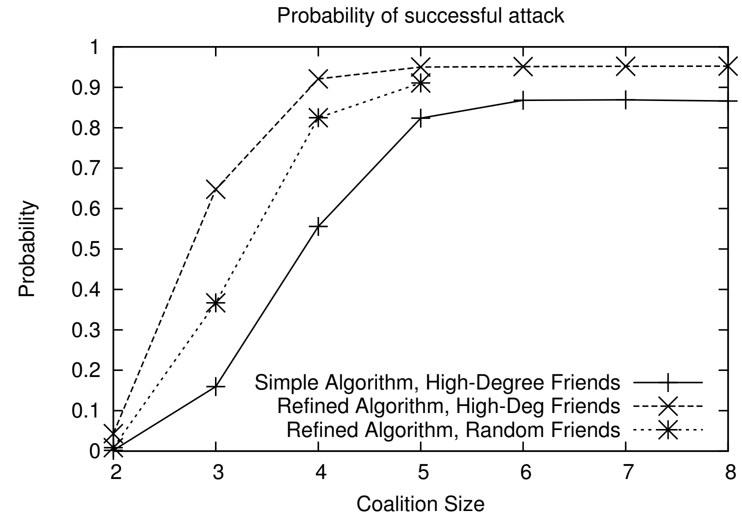
In LiveJournal graph, number of distinct k-node $N(v)$'s:

- Small k: approx. the number of distinct k-node graphs.
- Larger k: approx. the number of nodes of degree k.

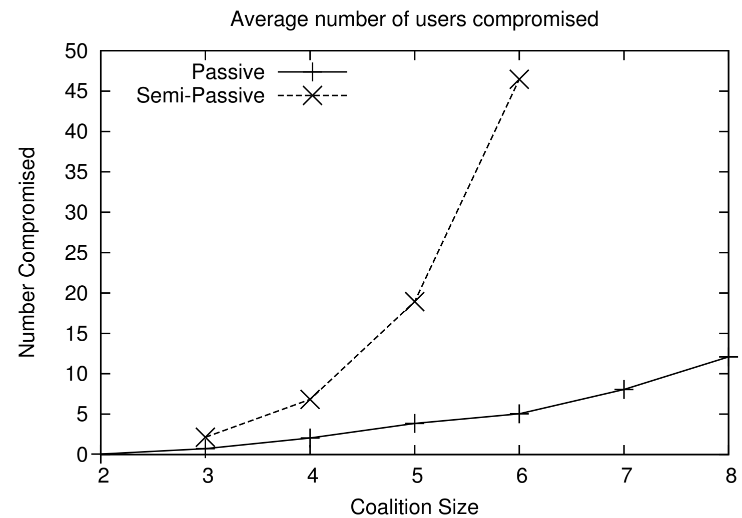
If your degree is reasonably large, your pattern of friends is very likely unique.

Passive Attacks

- Don't need full neighbor subgraph.
- Attack has reasonable chance of success if you just recruit 4-6 of your friends.



- With 6 friends, can compromise about 10 nodes.
- Can compromise many more with some advance linking: a "semi-passive" attack.



The Perils of Anonymized Data

What's the conclusion from all this?

- Doesn't apply to social network data that's already public; orthogonal to issues of legal/contractual safeguards.
- But widespread release of an anonymized social network? Danger: you don't what someone's hidden in there. And passive attacks don't even require advance planning.
- Further directions: privacy-preserving mechanisms for making social network data accessible.
  - May be difficult to obfuscate network effectively (e.g. [Dinur-Nissim 2003, Dwork-McSherry-Talwar 2007])
  - Interactive mechanisms for network data may be possible (e.g. [Dwork-McSherry-Nissim-Smith 2006])