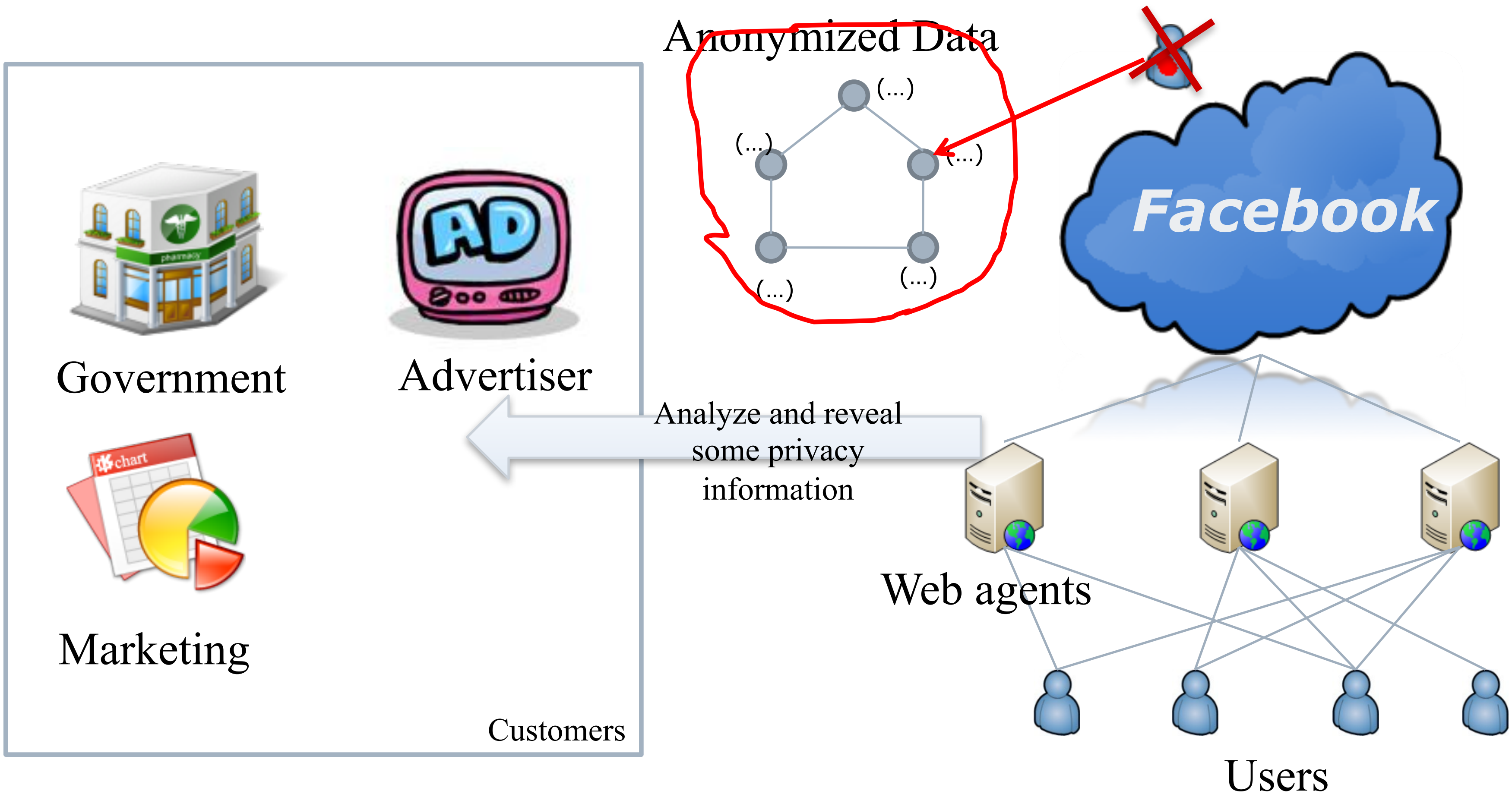
Privacy Preserving Graph Publication

Social Network Benefits



Protection Methods

- Two methods
  - Publishing sanitized graph
  - Publishing noised aggregate information
    - Differential privacy on graph

Publishing sanitized graph

1. Privacy protection and the attack models
2. Preventing passive attacks
3. Preventing active attacks
4. Other works

Publishing sanitized graph

① Privacy protection and the attack models

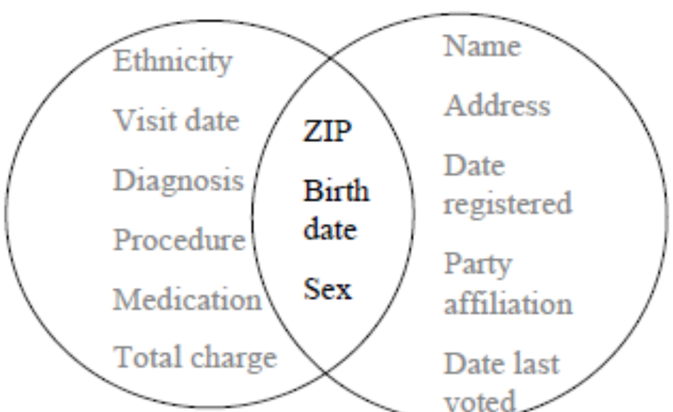
② Preventing passive attacks

③ Preventing active attacks

④ Other works

Attack the Anonymized Data

- An attacker
  - Background knowledge
    - The information he knows about a victim
  - Sensitive information
    - The information that user cares



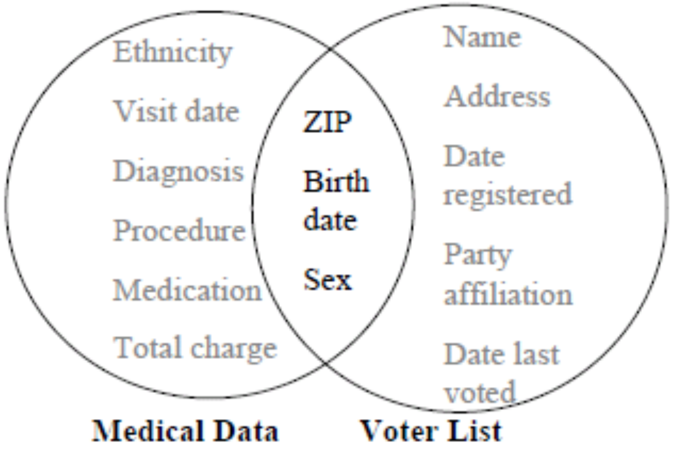
Medical Data **Voter List**

Race	BirthDate	Gender	ZIP	Problem
black	9/20/1965	male	02141	short of breath
black	2/14/1965	male	02141	chest pain
black	10/23/1965	female	02138	painful eye
black	8/24/1965	female	02138	wheezing
black	11/7/1964	female	02138	obesity
black	12/1/1964	female	02138	chest pain
white	10/23/1964	male	02138	short of breath
white	3/15/1965	female	02139	hypertension
white	8/13/1964	male	02139	obesity
white	5/5/1964	male	02139	fever
white	2/13/1967	male	02138	vomiting
white	3/21/1967	male	02138	back pain

Hospital Data

Attack the Anonymized Data

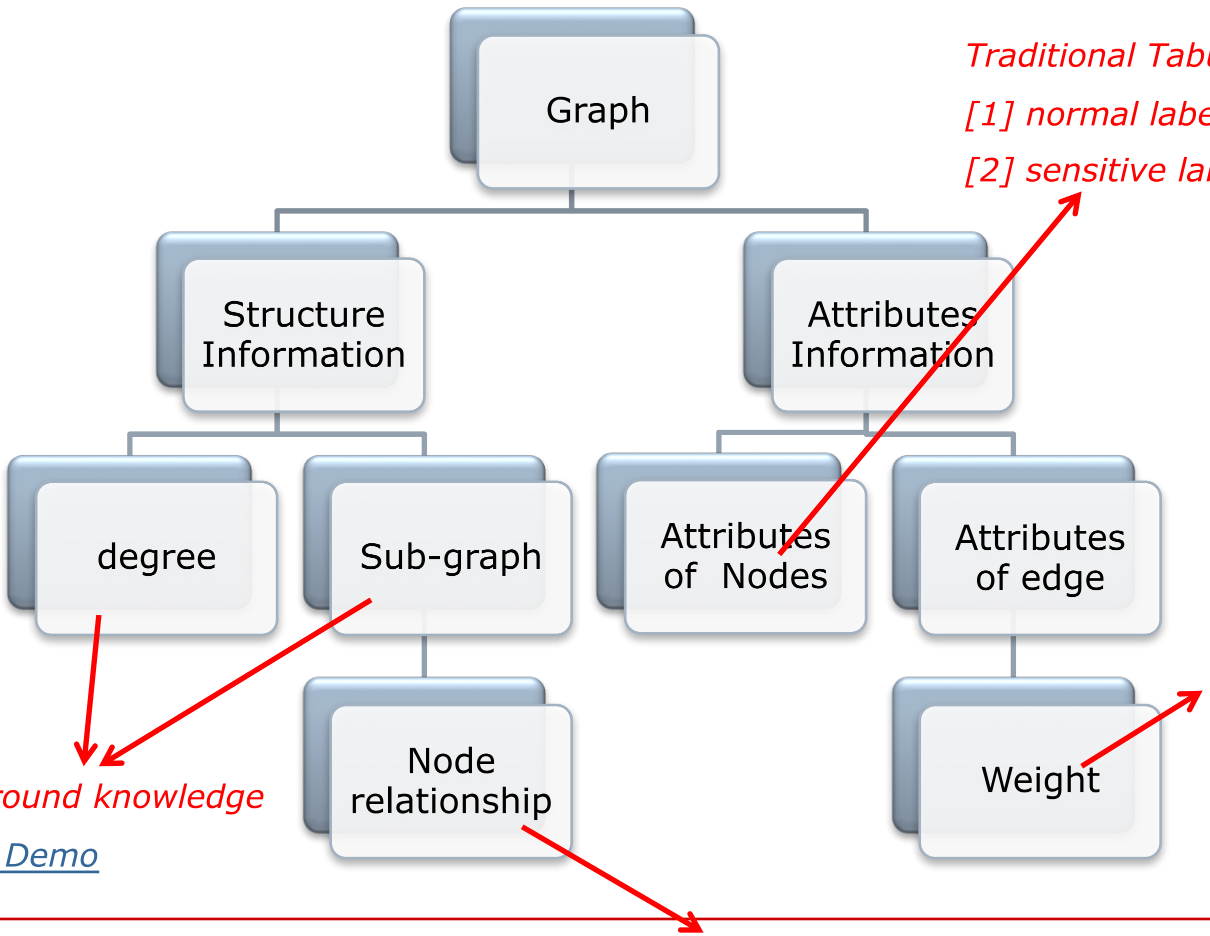
- An attacker
  - Background knowledge
    - The information he knows about a victim
  - Sensitive information
    - The information that user cares



	Race	Birth	Gender	ZIP	Problem
t1	Black	1965	m	0214*	short breath
t2	Black	1965	m	0214*	chest pain
t3	Black	1965	f	0213*	hypertension
t4	Black	1965	f	0213*	hypertension
t5	Black	1964	f	0213*	obesity
t6	Black	1964	f	0213*	chest pain
t7	White	1964	m	0213*	chest pain
t8	White	1964	m	0213*	obesity
t9	White	1964	m	0213*	short breath
t10	White	1967	m	0213*	chest pain
t11	White	1967	m	0213*	chest pain

2-Anonymous Hospital Data

Information in Social Networks



Traditional Tabular data

[1] normal labels   background knowledge

[2] sensitive labels   protected information

[1] Sensitive information

[2] Background knowledge

Background knowledge

Attack Demo

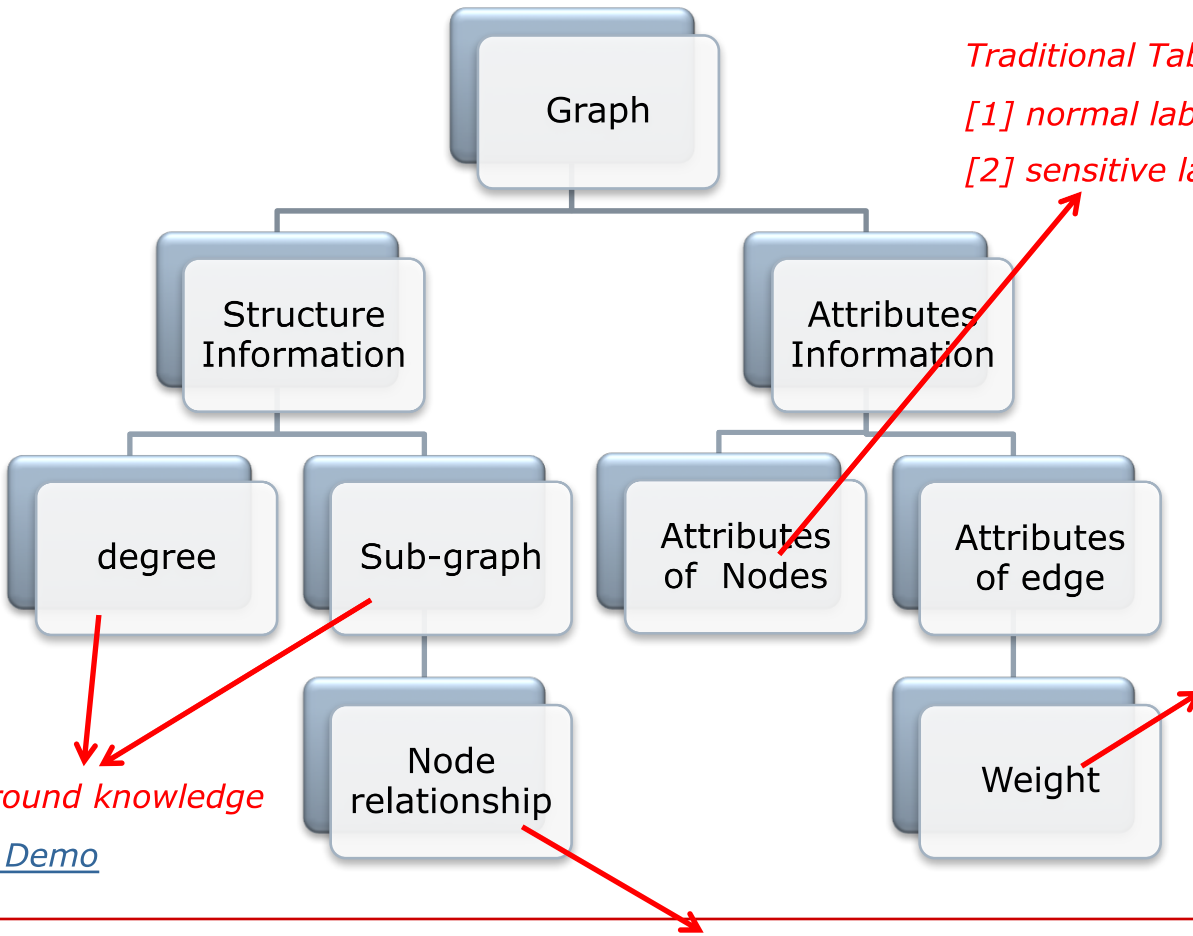
Sensitive information

Information in Social Networks

Traditional Tabular data

[1] normal labels   background knowledge

[2] sensitive labels   protected information



[1] Sensitive information

[2] Background knowledge

Background knowledge

[Attack Demo]

Sensitive information

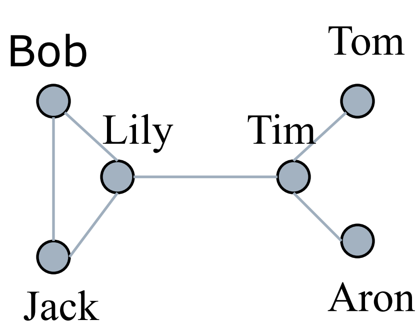
Protection objectives

Graph Model	Protection		Works
Unweighted Graph	Node Protection (Anti Node re-identification)	$\Pr ob(u \Rightarrow n) \le \frac{1}{k}$	[8][12][13][14] [15][21][22][23]
	Link Protection	$\Pr ob(con(u_1, u_2)) \le \frac{1}{k}$	[13][22]
		$\Pr ob(u \in e) \le \frac{1}{k}$	[13]
Weighted Graph	Edge weights	Hide the real edge weights	[17][24]
		Hide the relative order between weights	[24]

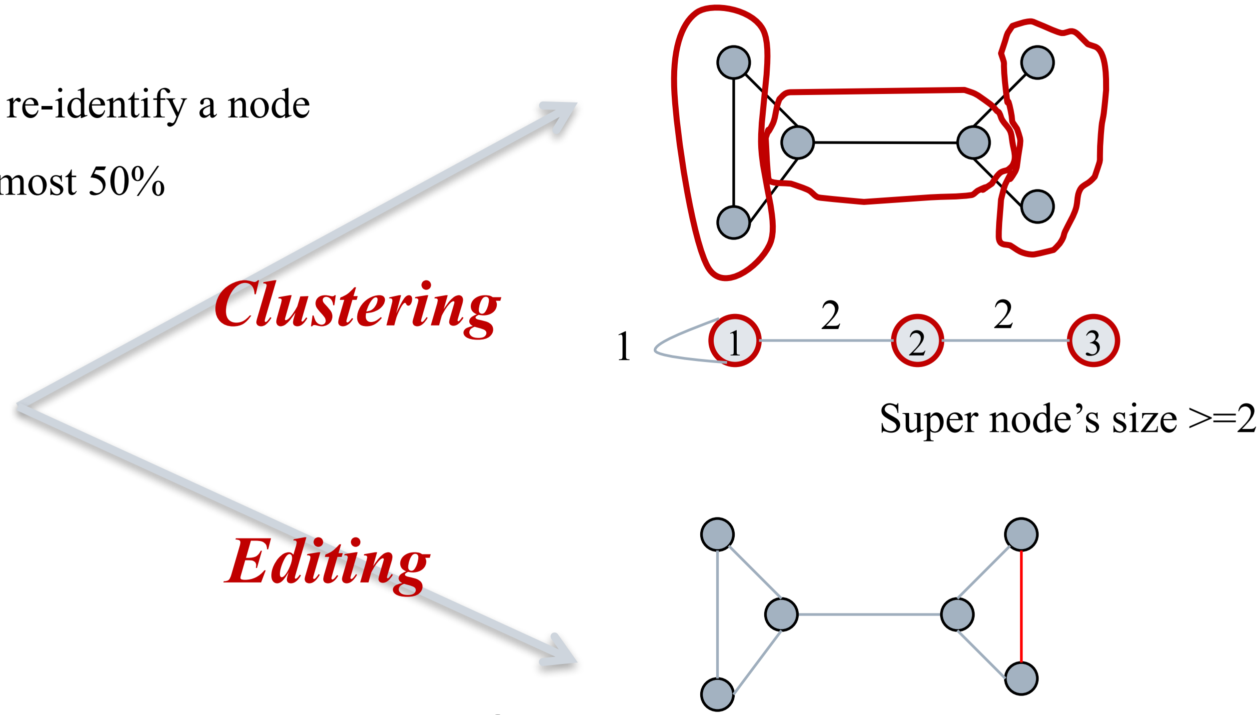
Privacy Protection Method

k=2

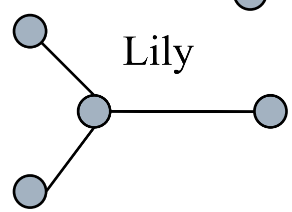
An attack can only correctly re-identify a node with probability at most 50%



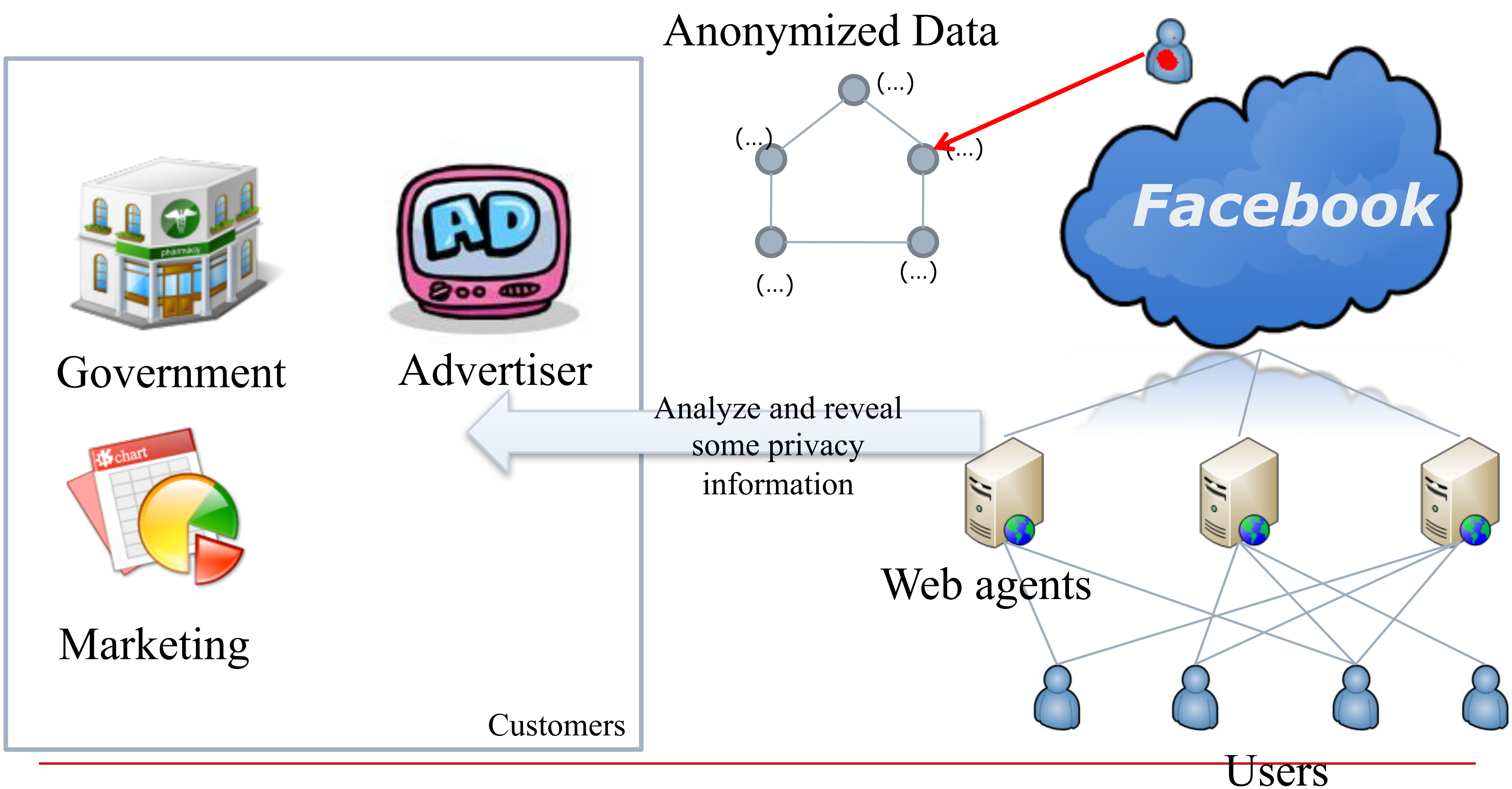
Original Graph



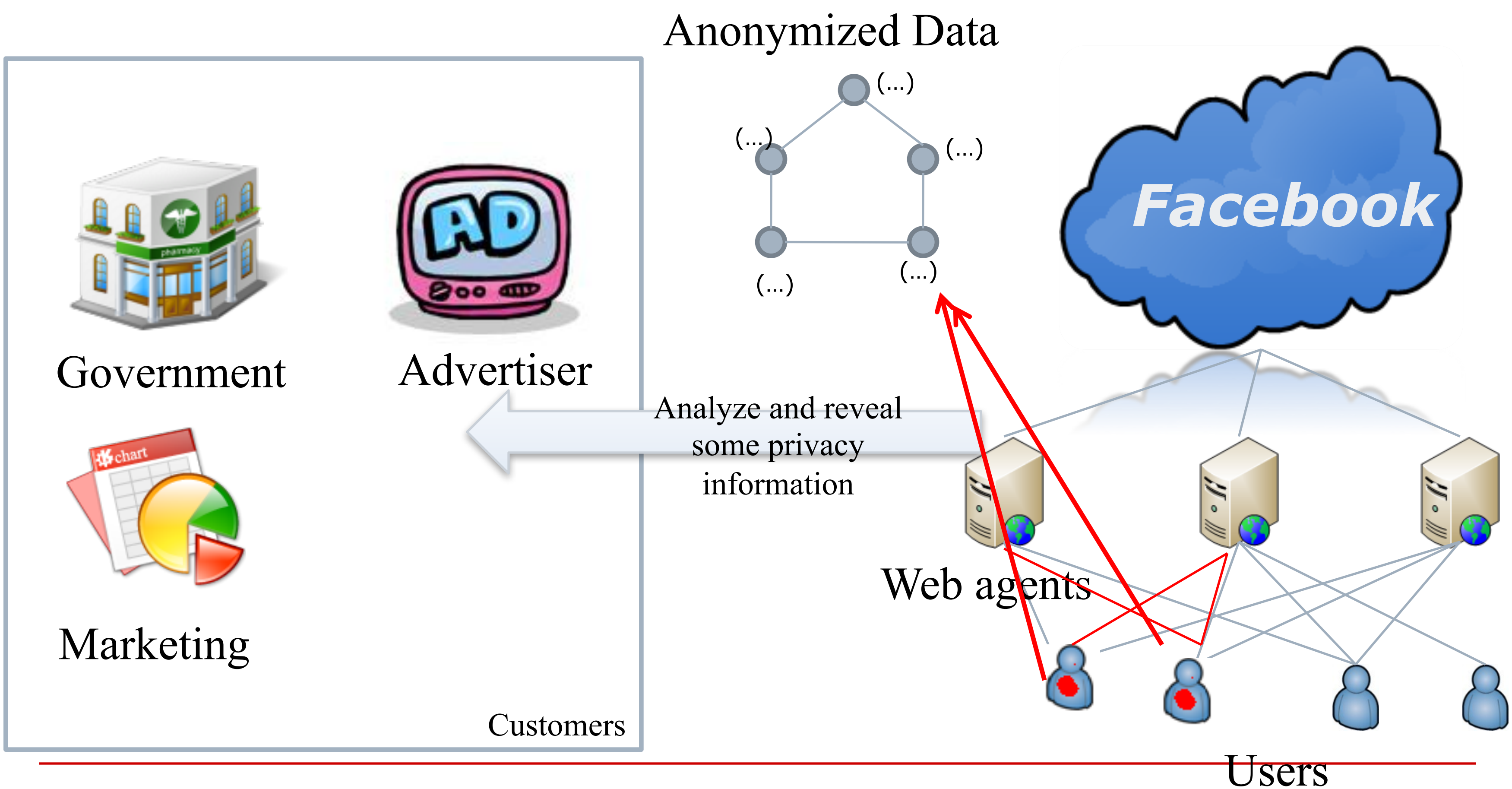
An attacker's knowledge



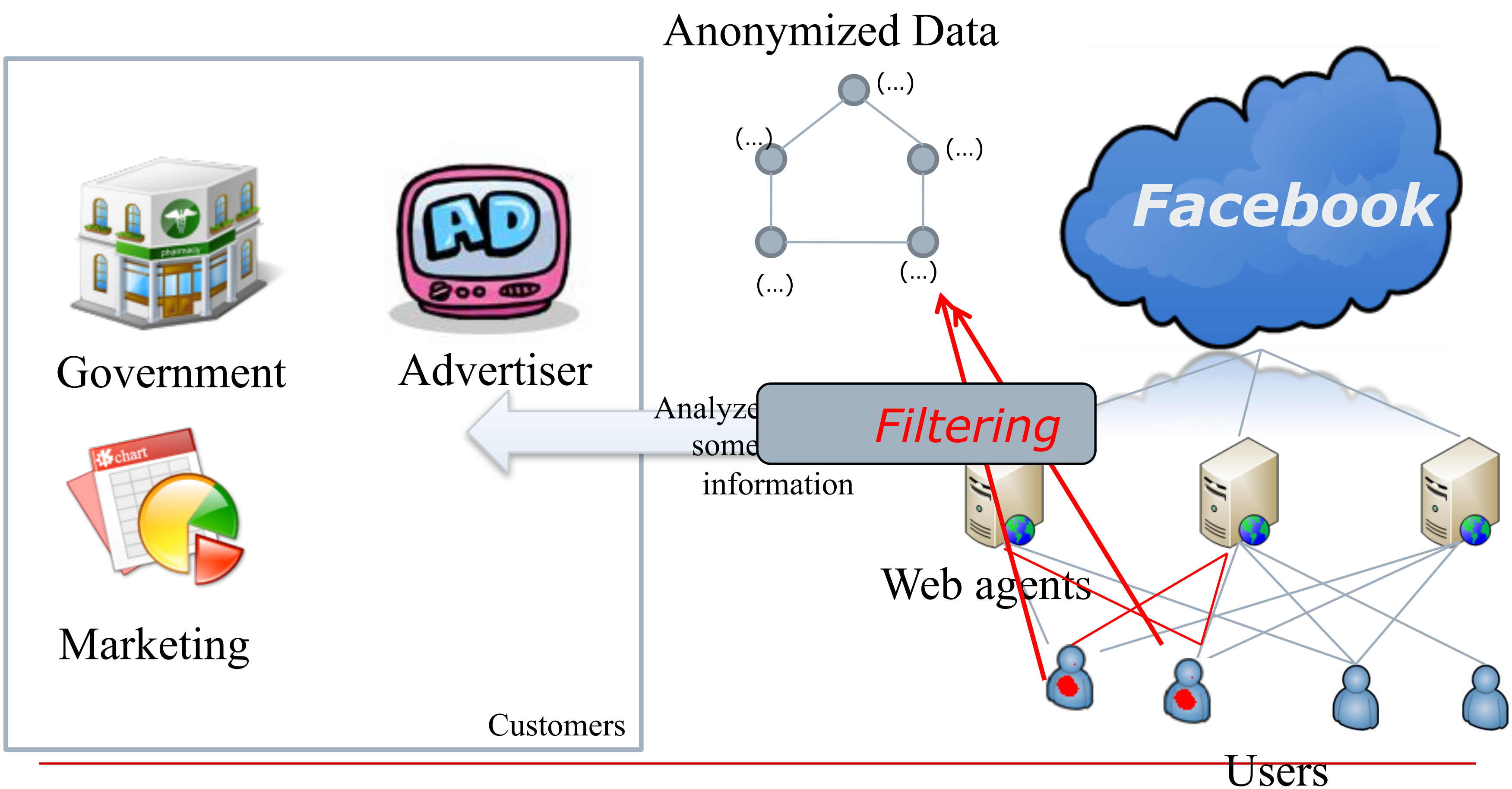
Passive attack and Active attack



Passive attack and Active attack



Anti Active attack



Current works

Prevent Attack Type	Method	Papers
Passive Attack	Clustering	[8][13] [15][16]
	Node/Edge Editing	[10][11][12][14][16][18][21][22][23]
	Protecting edge weights	[17][24]
Active Attack	Fake Nodes Recognition	[11][25]
	Parameter Analysis	[9]

Publishing sanitized graph

① Privacy protection and the attack models

② Preventing passive attacks

  ① Edge editing based models

③ Preventing active attacks

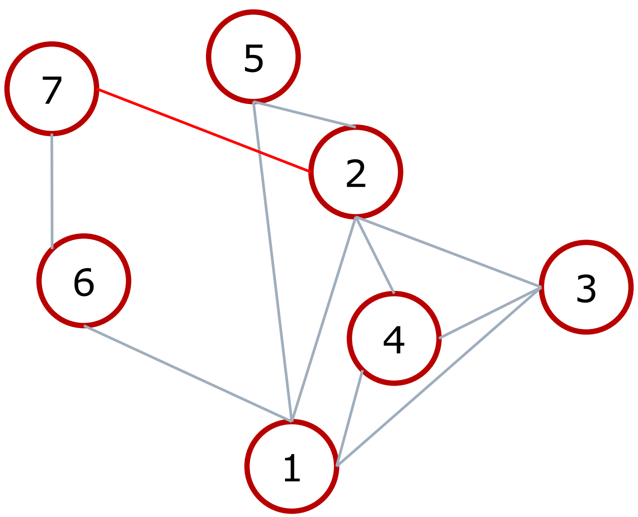
④ Other works

Edge editing based models

Name	Structure knowledge	Protection objective
K-degree anonymous	Node degrees	Avoid Node re-identification
K-neighborhood anonymous	Neighborhood graph	Avoid Node re-identification
K-automorphism anonymous	Any subgraph	Avoid Node re-identification
K-symmetric anonymous	Any subgraph	Avoid Node re-identification
K-isomorphism	Any subgraph	Avoid Node re-identification Avoid Edge discovery
Random change edge model	Neighborhood graph	Avoid Node re-identification Avoid Edge discovery

K-degree anonymous[12]

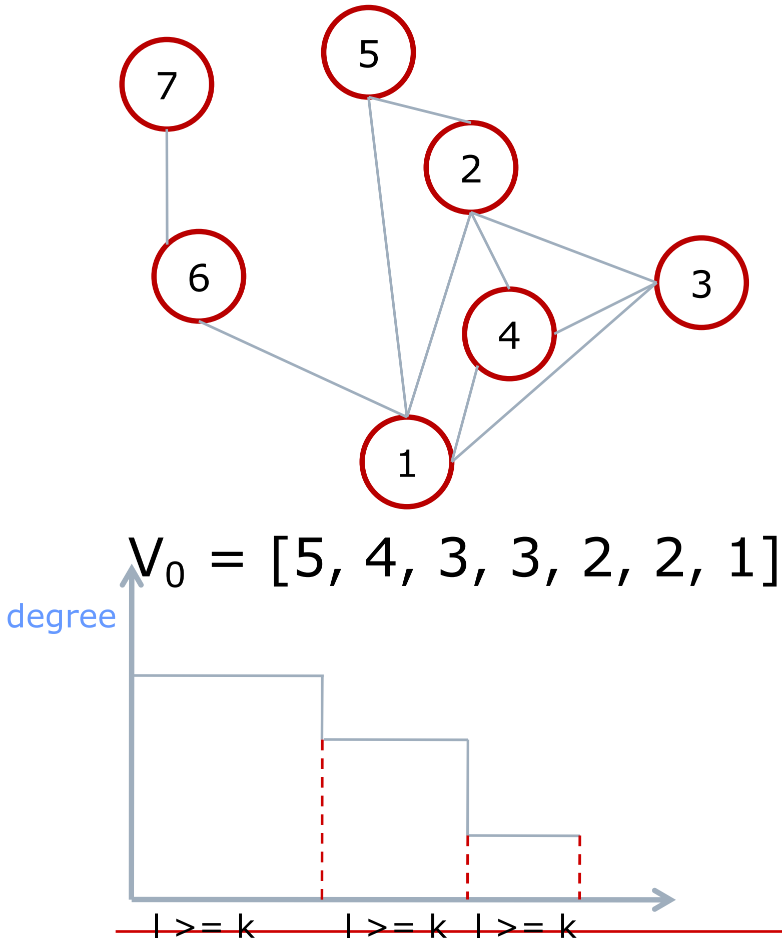
- K-degree anonymous
  - For every node v, there exist at least k-1 other nodes in the graph with the same degree as
    - No single node class is identified at H_0 vertex refinement queries

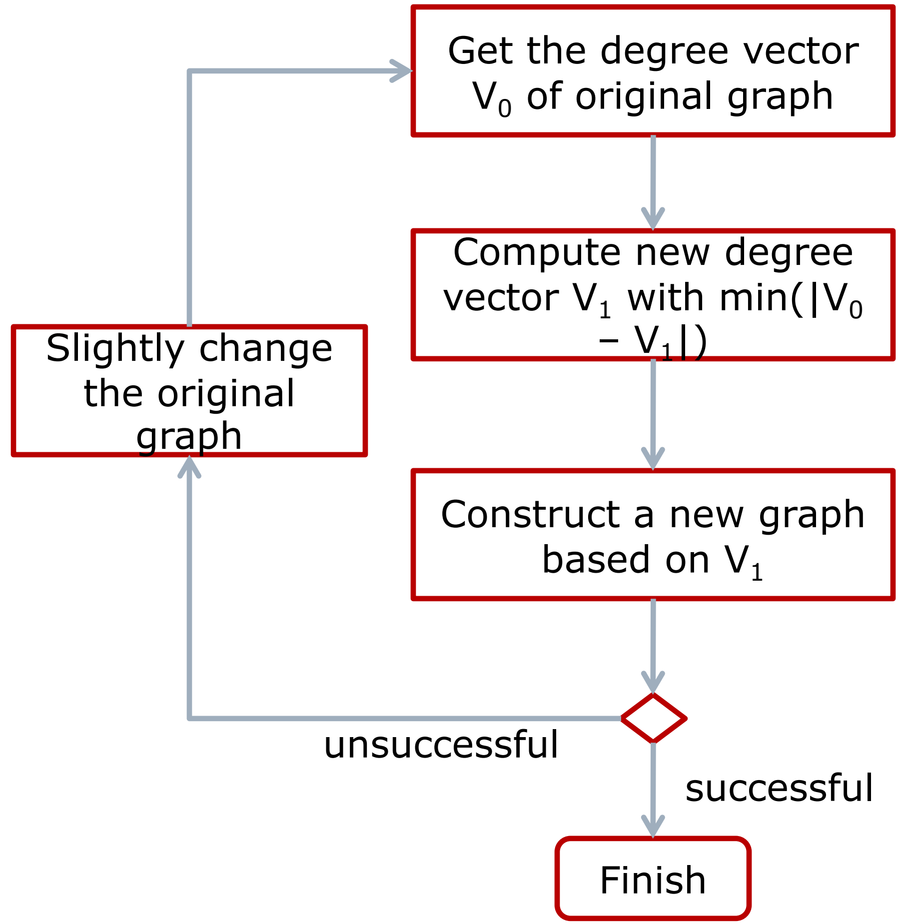


Node	Degree
1	5
2	5
3	3
4	3
5	2
6	2
7	2

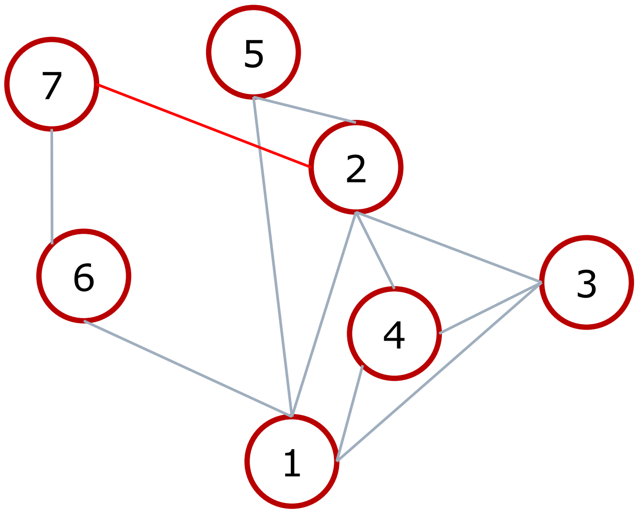
Achieve k-degree anonymous by adding/deleting edges

K-degree algorithm skeleton



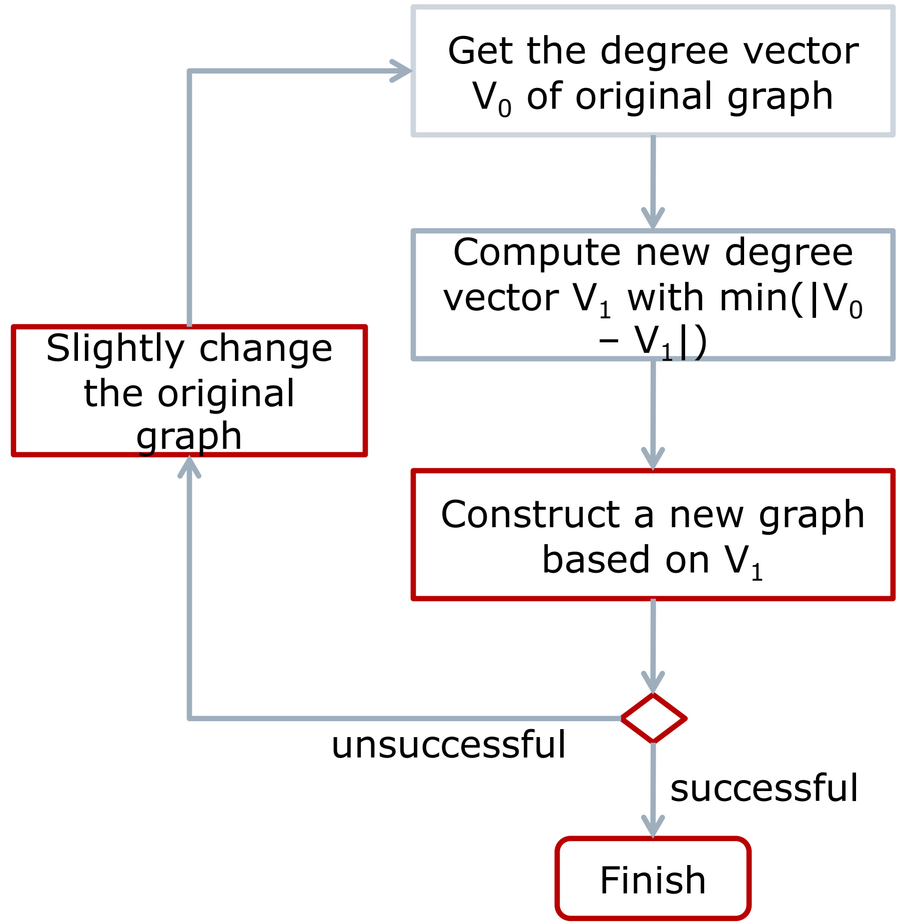


K-degree algorithm skeleton



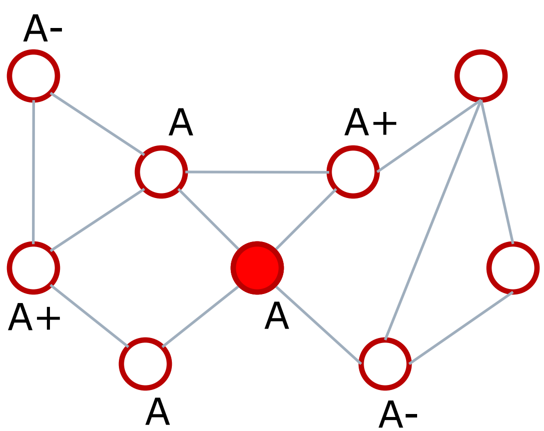
$$V_0 = [5, 4, 3, 3, 2, 2, 1]$$

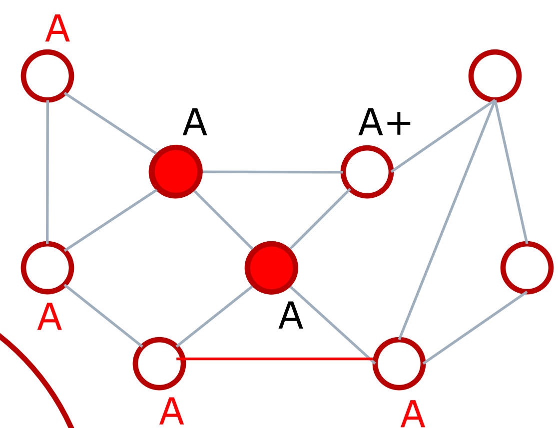
$$V_1 = [5, 5, 3, 3, 2, 2, 2]$$

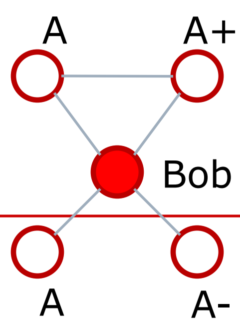


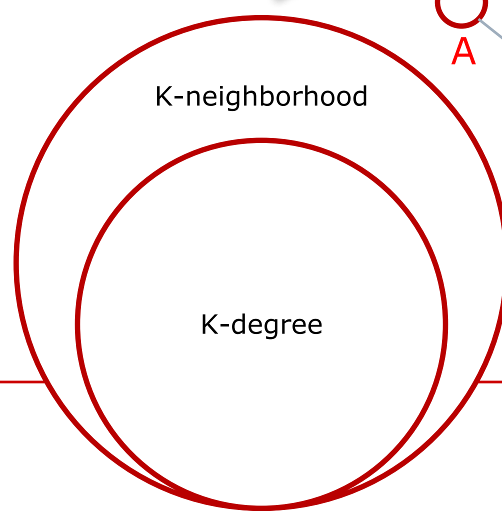
K-neighborhood[14]

- K-neighborhood anonymous
  - For every node v, there exist at least k-1 other nodes in the graph with the same m-hop neighborhood sub-graph
    - No single node class is identified by sub-graph queries





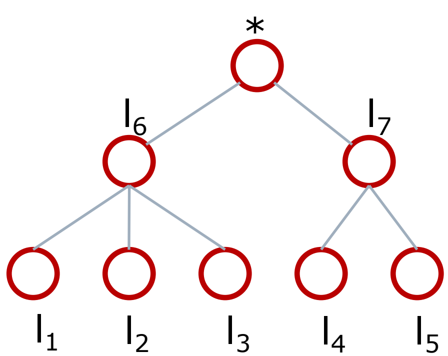


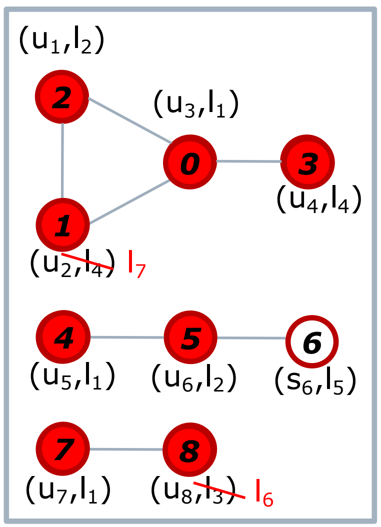


K-neighborhood algorithm skeleton

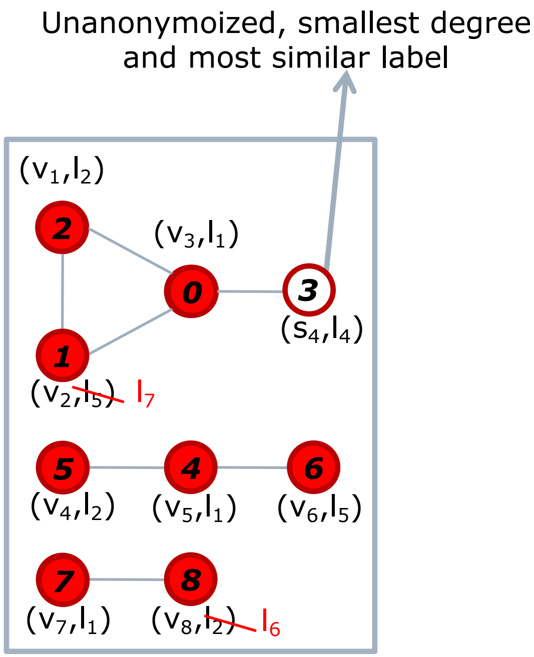
- Neighborhood representation problem
  - Minimum DFS code is unique
- Two nodes’ neighborhood anonymous problem

Label Category Tree





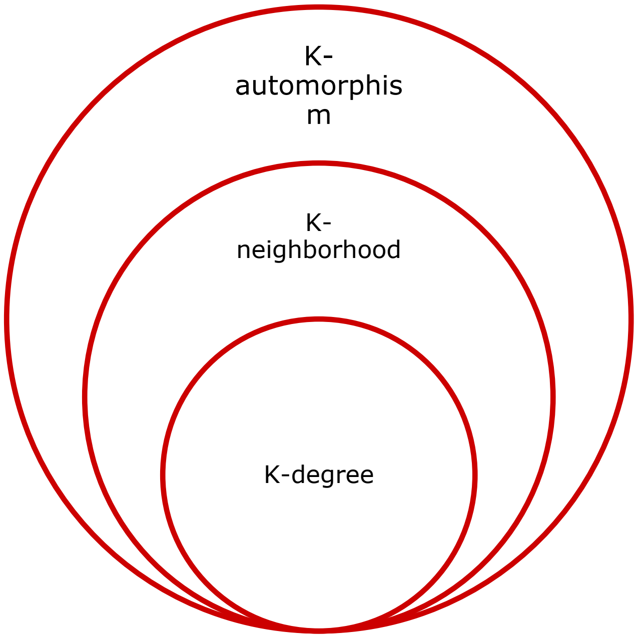
u’s neighborhood graph



v’s neighborhood graph

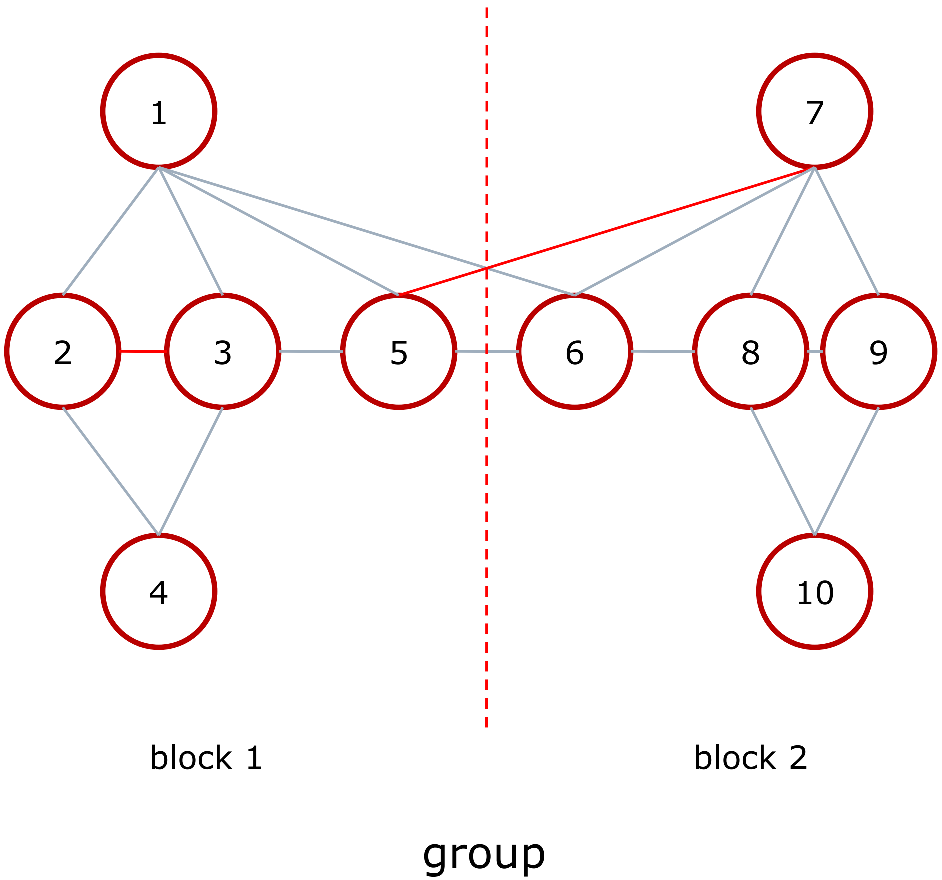
K-automorphism[21]

- K-automorphism [21] (k-symmetric [23]) anonymous
  - For every node v, there exist at least k-1 other nodes in the graph that are same on the
  - The graph should be k-symmetry
  - No single node class is identified by any kind of structure queries



When k-neighborhood consider the neighborhood of nodes in l step, l = the longest path in graph, k-neighborhood = k-automorphism

K-automorphism Algorithm Skeleton



Partition the graph to groups, each group has at least k blocks

↓

For each block, perform graph alignment

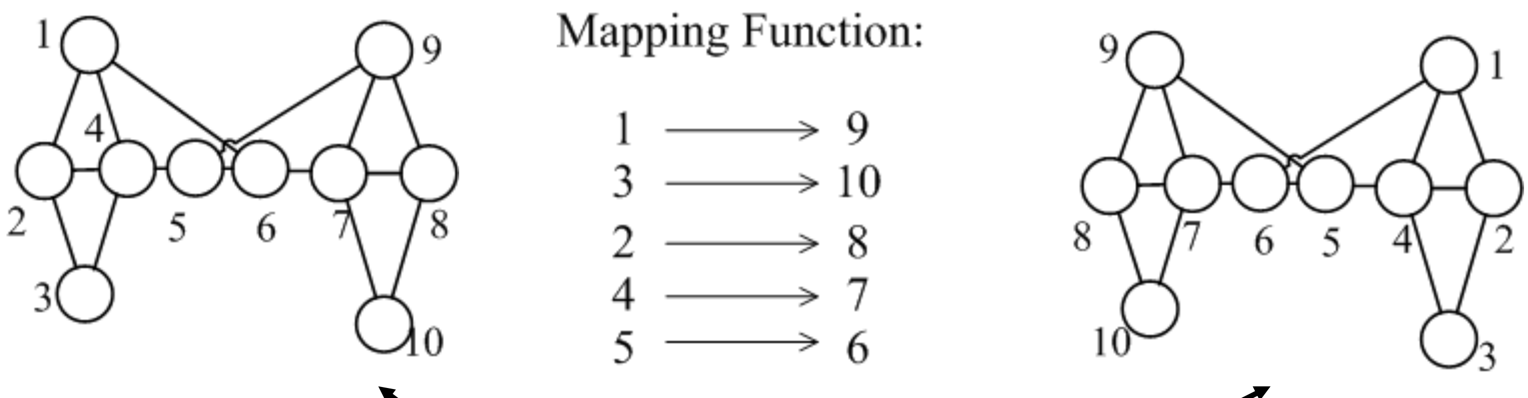
↓

For all crossing edges of blocks, perform edge copy

↓

Finish

K-Automorphism Network



Graph isomorphism

The Motivation

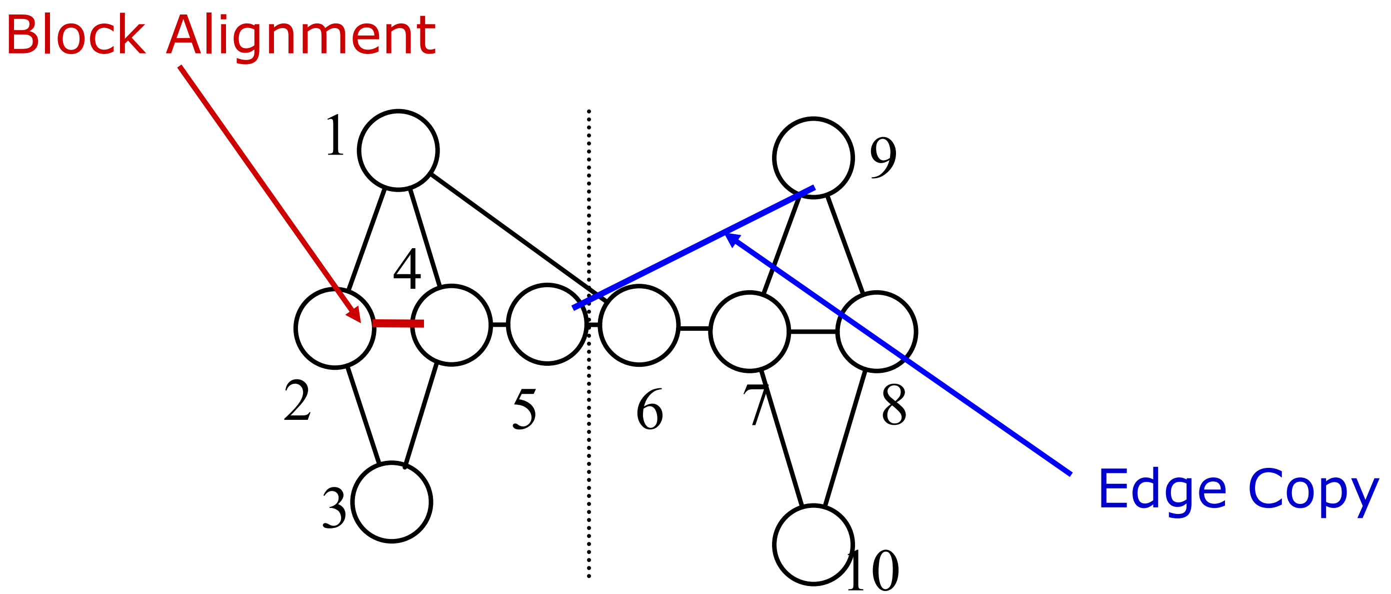
If the released graph is a k-automorphism network,

It can resist any attack.

Problem Definition:

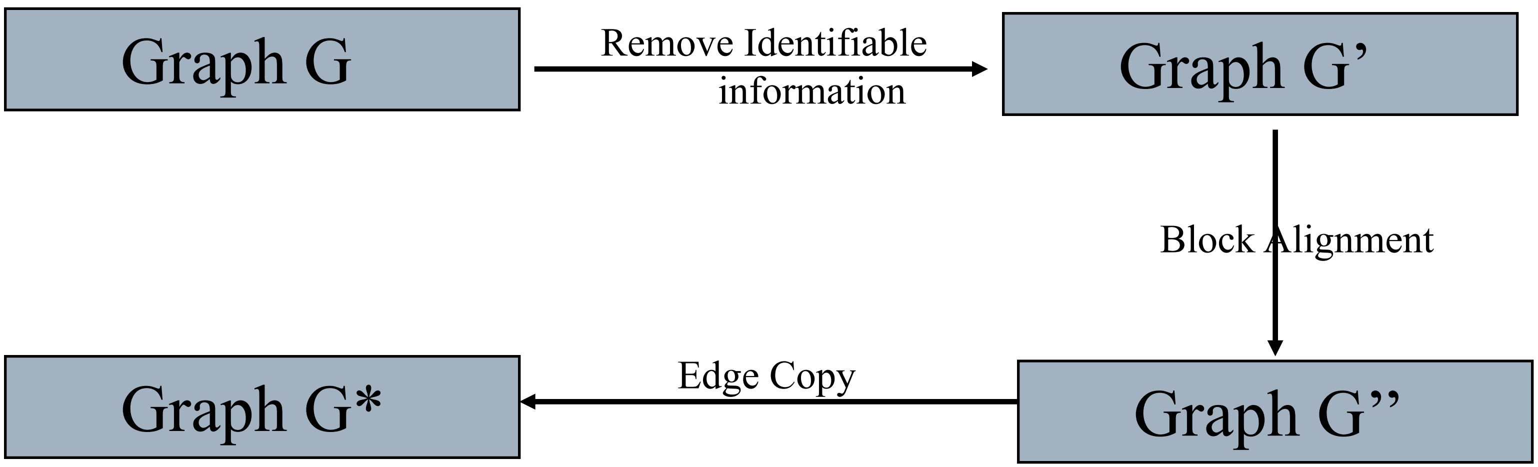
Given an original network *G*, find a network *G**, where *G* is a sub-graph of *G**, and *G** is a k-automorphic network. *G** is published as *G*'s anonymized version. Furthermore, we require that Cost(G,G*) is minimized.

KM Algorithm-(Overview)

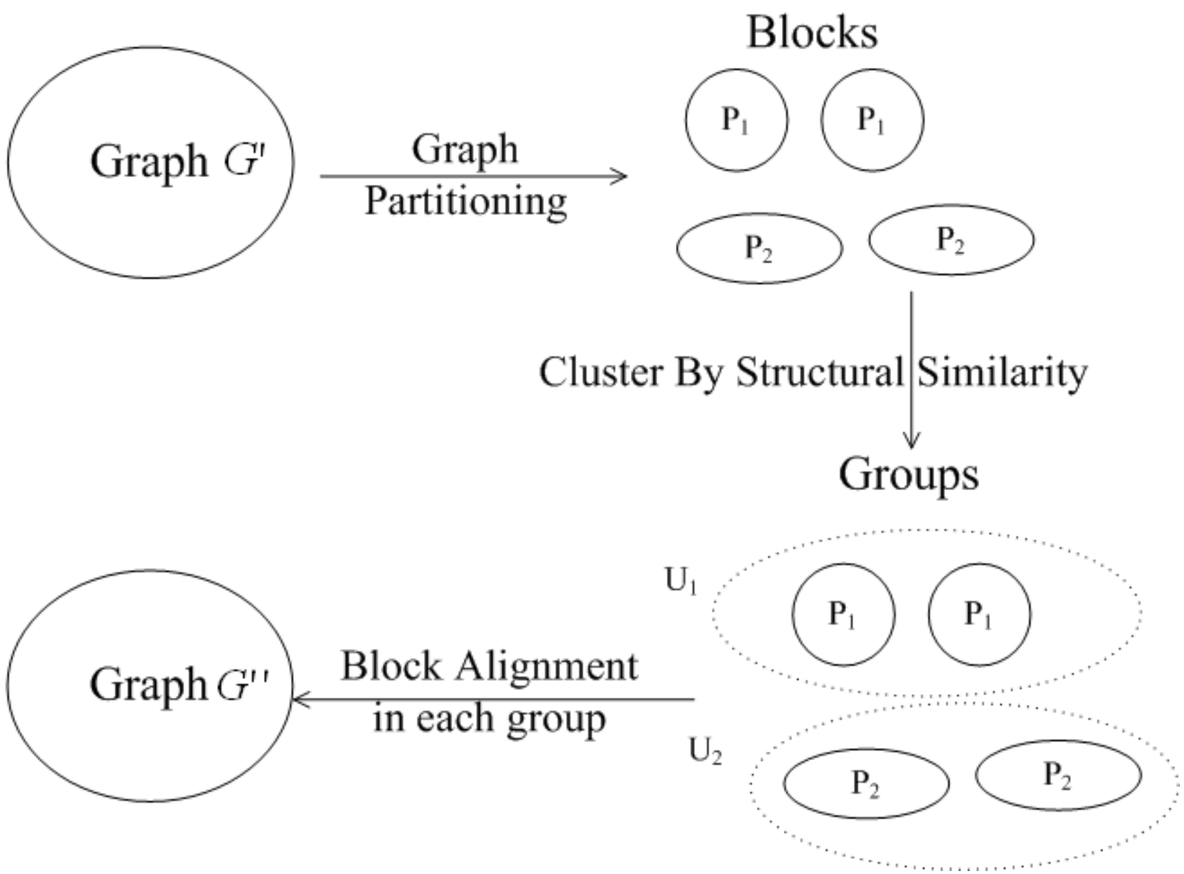


(a) Naïve anonymization Network G'

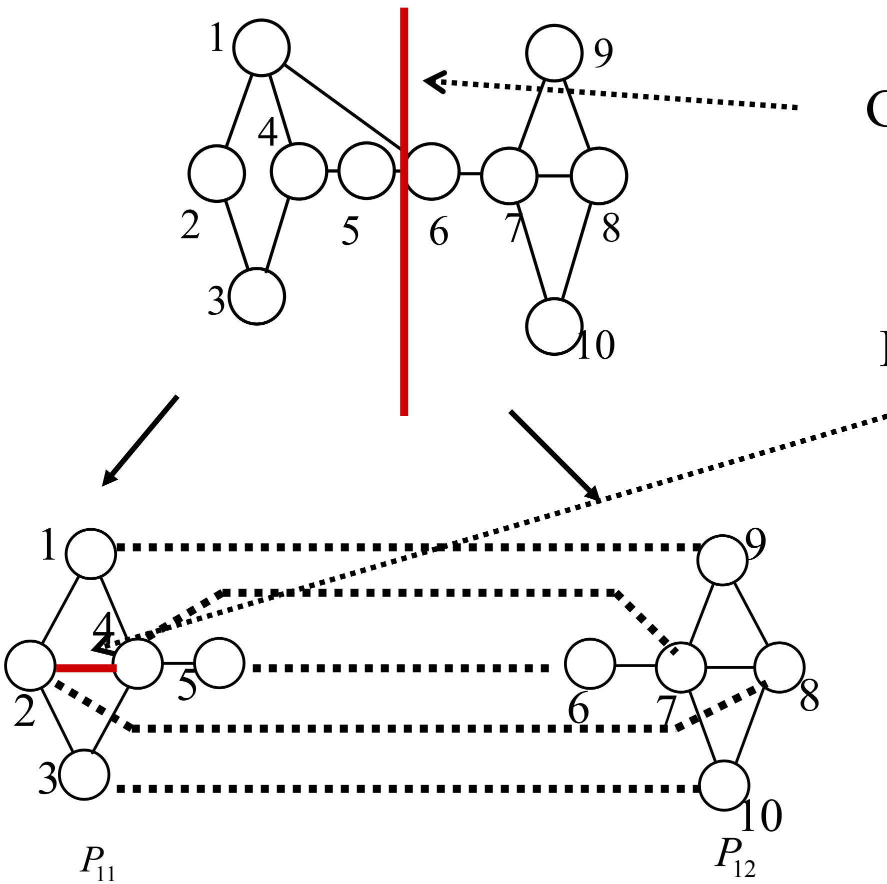
Framework


Graph G
Remove Identifiable information
Graph G'
Block Alignment
Graph G''
Edge Copy
Graph G*

Block Alignment



Block Alignment



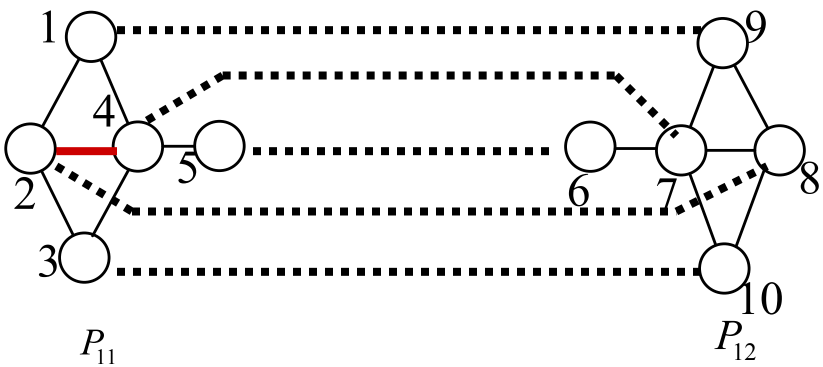
Graph Partition

Block Alignment

1	9
4	7
5	6
2	8
3	10

Alignment Vertex Table (AVT)

An Optimal Block Alignment



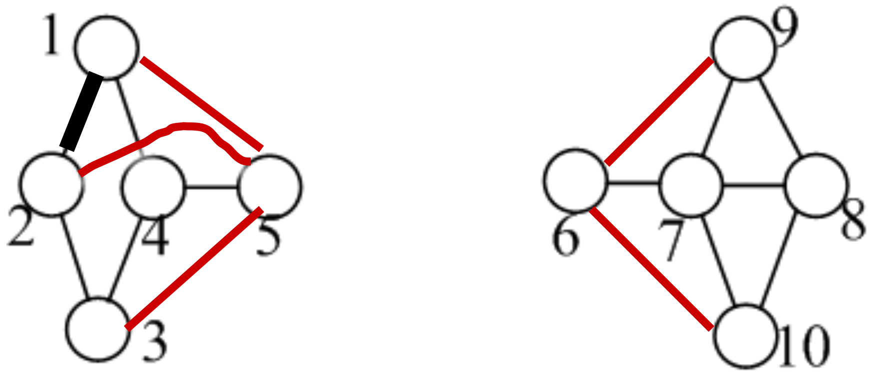
1	9
4	7
5	6
2	8
3	10

Alignment Vertex Table (AVT)

We prove the optimal block alignment is NP-hard

Degree-Based Alignment

The largest same
degree

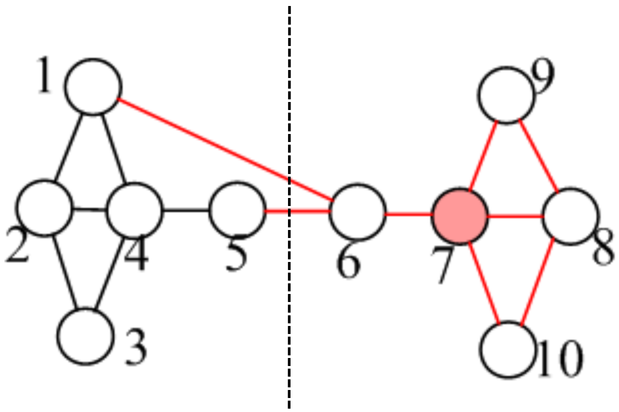


BF-search
Pair-up vertices with the same
or similar degrees

Vertex Alignment Table

4	8
1	9
3	10
5	7
2	6

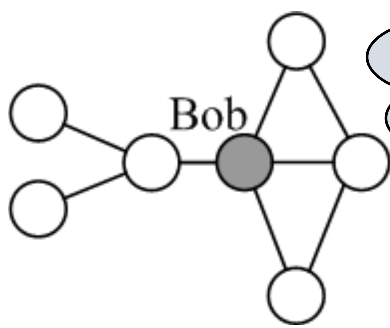
After Block Alignment



Alignment Vertex Table
(AVT)

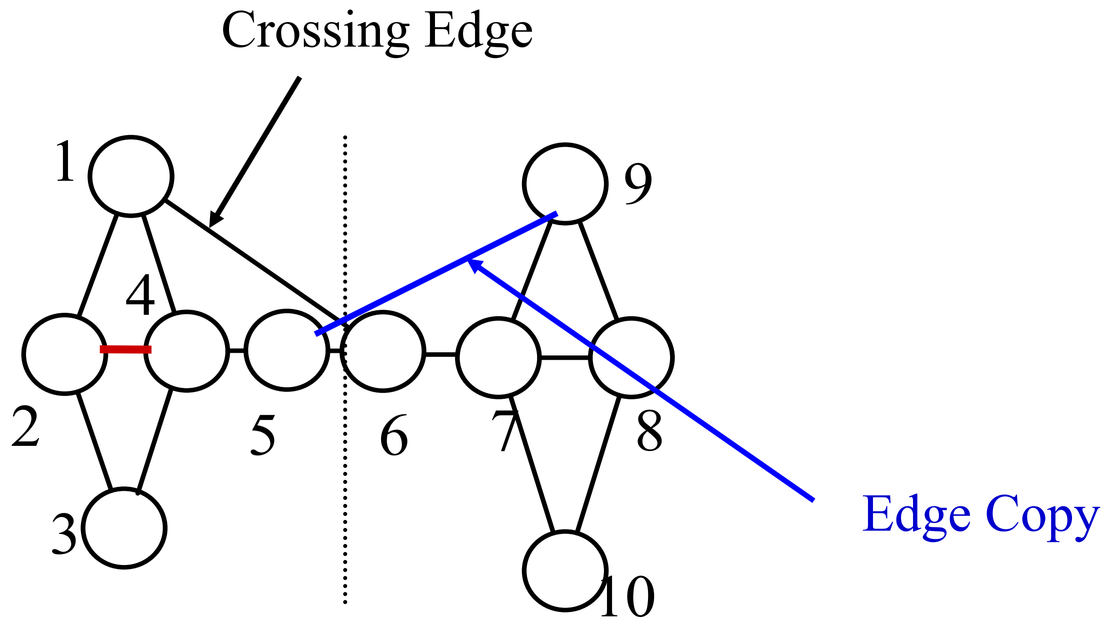
1	9
4	7
5	6
2	8
3	10

Query:



The privacy of Bob is
still compromised

Edge Copy



(a) Naïve anonymization
Network G'

Edge Copy

According to Automorphic Function, duplicate all crossing edges.

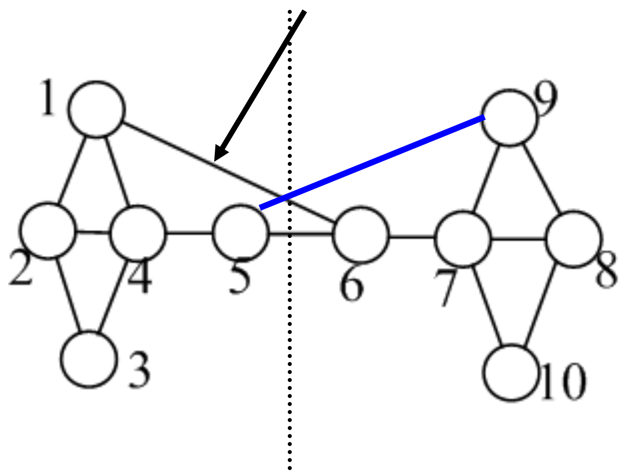
Alignment Vertex Table (AVT)

1	9
4	7
5	6
2	8
3	10

Automorphic Function:

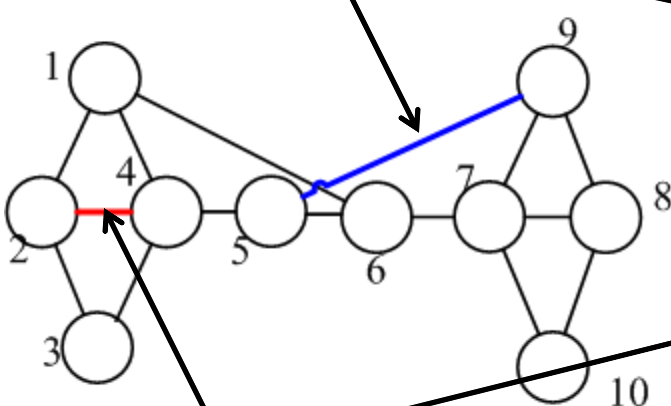
F(1)=9; F(9)=1;
F(4)=7; F(7)=4;
F(5)=6; F(6)=5;
F(2)=8; F(8)=2;
F(3)=10; F(10)=3;

Crossing Edge



Cost

Edges introduced during edge copy



Edges introduced during alignment

Given a group U_i of blocks P_{ij}, $j=1,\ldots,k$,
the anonymization cost of group U_i is defined as follows:

$$\mathbf{Cost(U_i)=} \boxed{\mathbf{AlCost(U_i)}}+\boxed{\mathbf{0.5*(k-1)*\textstyle\sum_j|CrossEdge(P_{ij})|}}$$

The total cost is the sum of all group costs.

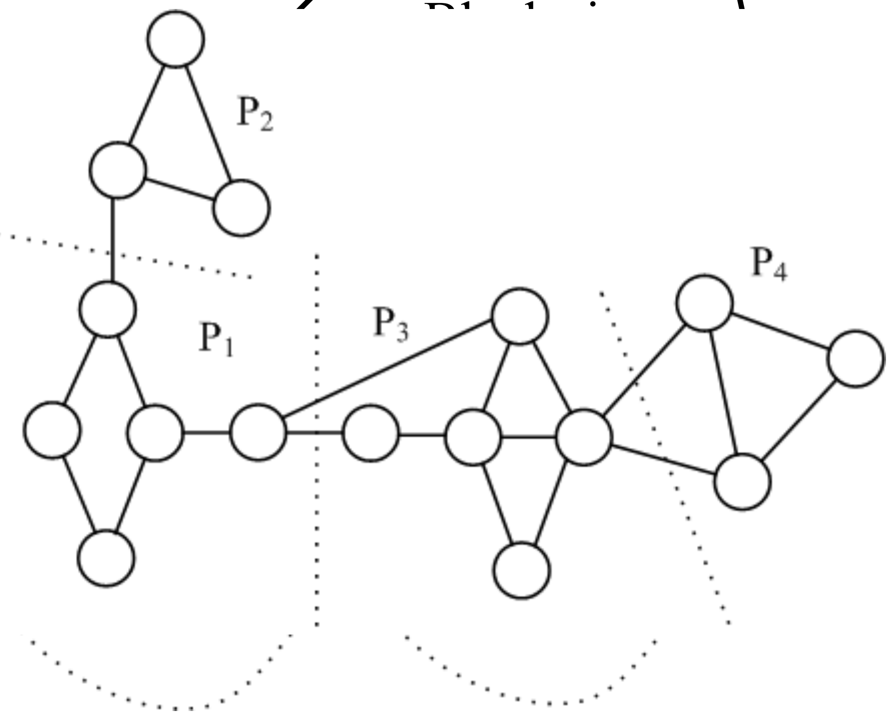
Graph Partition

Objection of this step:

Partition graph G' into *n* blocks, and cluster these blocks into *m* groups U_i. Each group U_i has no less than *k* blocks.

Graph Partition

Blocks in another group



1) Set Min_sup= *k* (i.e. k=2)

2) Find the matches of the largest frequent subgraphs (non-overlapping) as the initial group U of blocks.

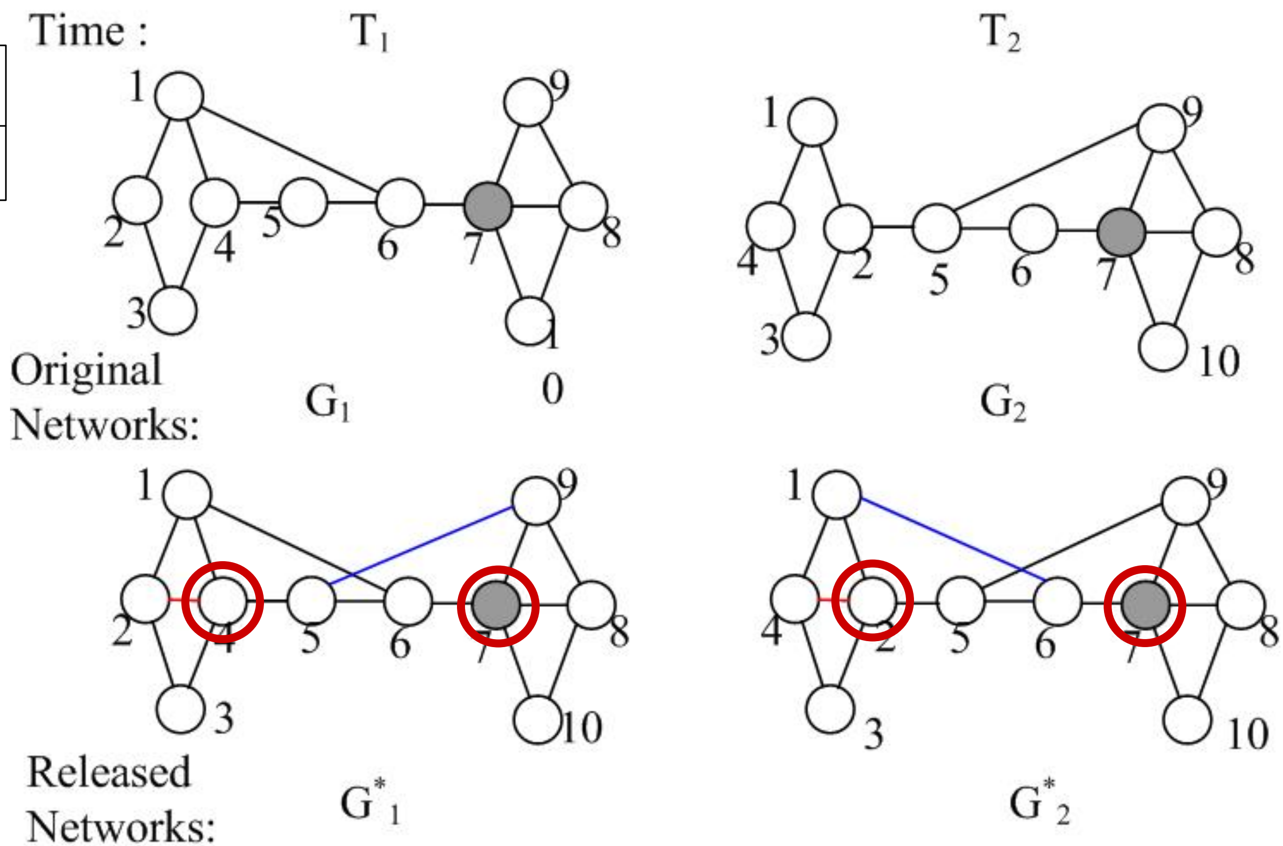
3) Expand and alignment all blocks in the group U, until Cost(U) is increased.

4) Extract all blocks in group U from the original graph G.

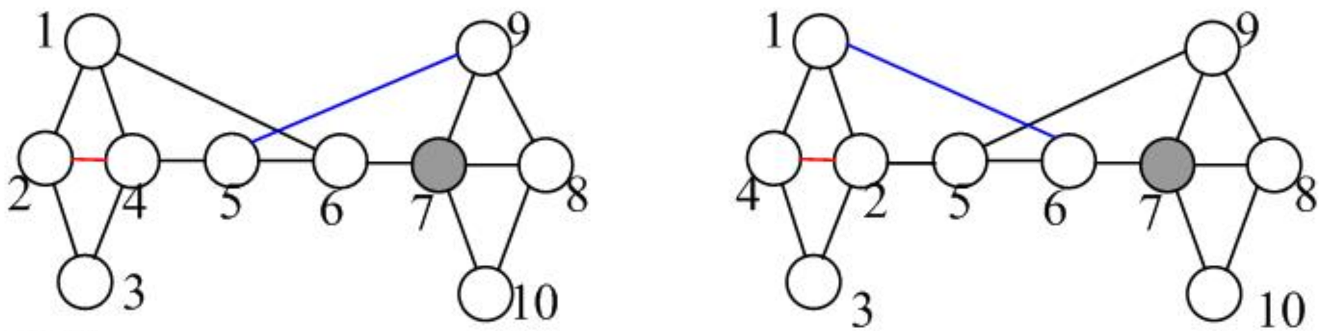
Iterate Steps 1-3 until no vertices in Graph left.

Dynamic Releases

T_1	{4, 7}
T_2	{2, 7}



Vertex ID Generation



Released Networks: G^*_1 G^*_2

T_1

AVT	A_1
1	9
2	8
4	7
5	6
3	10

T_2

AVT	A_2
1	9
4	8
2	7
5	6
3	10

$$\mathrm{Re}\,s(7, G_1^*) = \{4,7\}$$

$$\mathrm{Re}\,s(7, G_2^*) = \{2,7\}$$

$$\mathrm{Re}\,s(7, G_1^*) \bigcap \mathrm{Re}\,s(7, G_2^*) = 7$$

Vertex ID Generation

T_1

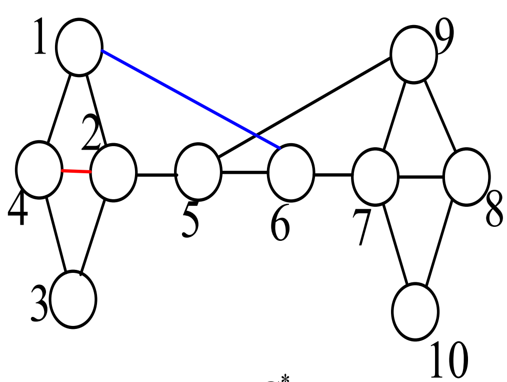
AVT	A_1
1	9
2	8
4	7
5	6
3	10

T_2

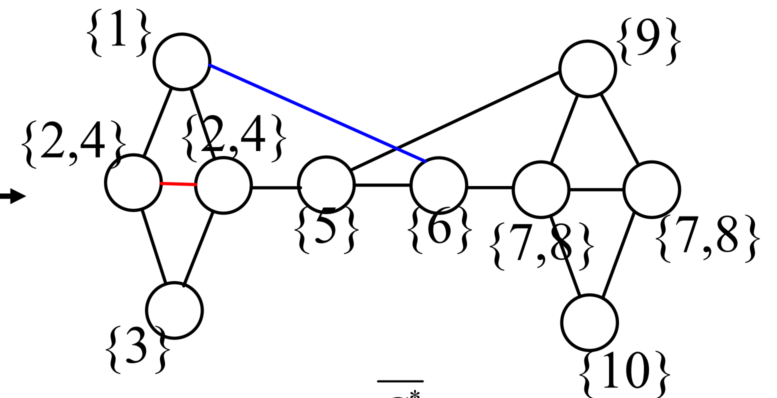
AVT	A_1
1	9
4	8
2	7
5	6
3	10

Generalized vertex ID table

OriID	GenID
1	{1}
2	{2,4}
3	{3}
4	{2,4}
5	{5}
6	{6}
7	{7,8}
8	{7,8}



G_2^*



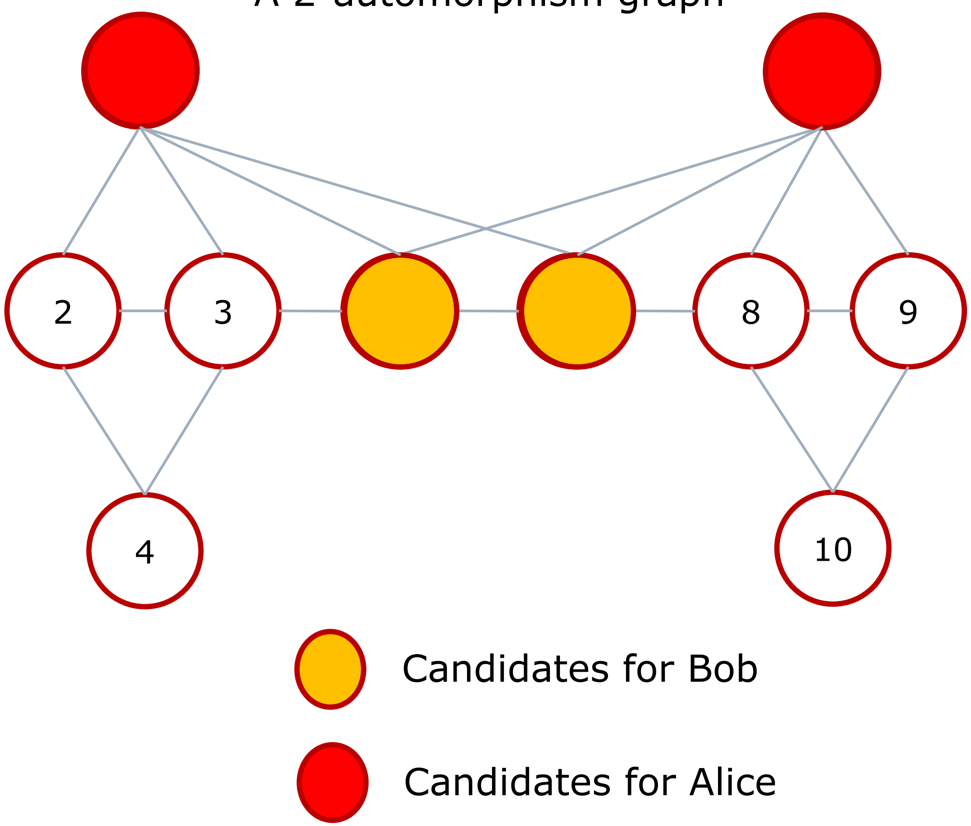
$\overline{G_2^*}$

Link Protection

- Models
  - K-degree
  - K-neighborhood
  - K-automorphism (K-symmetric)
- Protection objective
  - Preventing node re-identification
- Link Protection?

Link Leakage in k-automorphism

A 2-automorphism graph

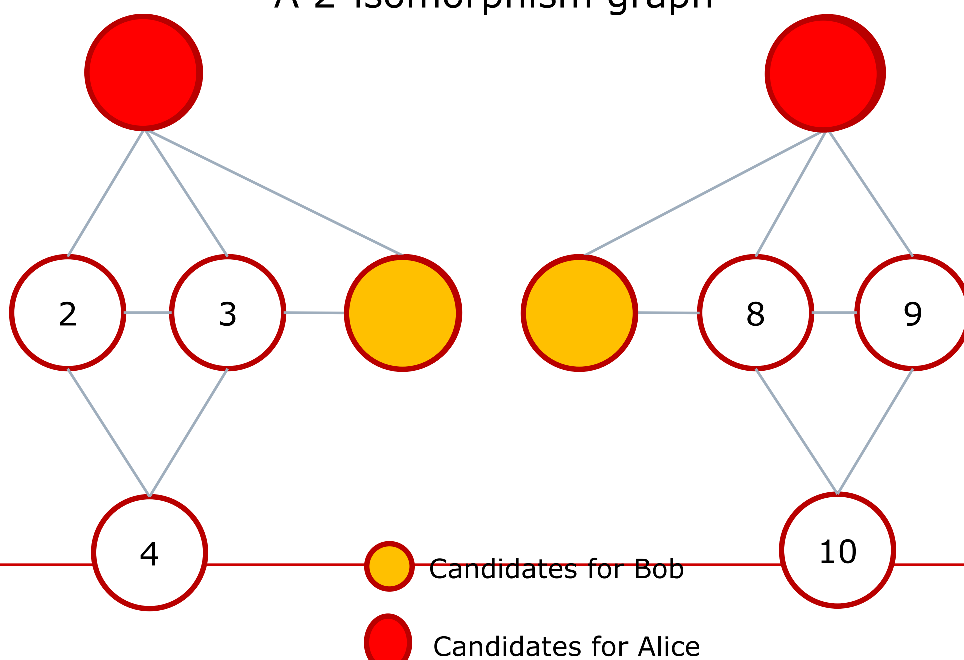


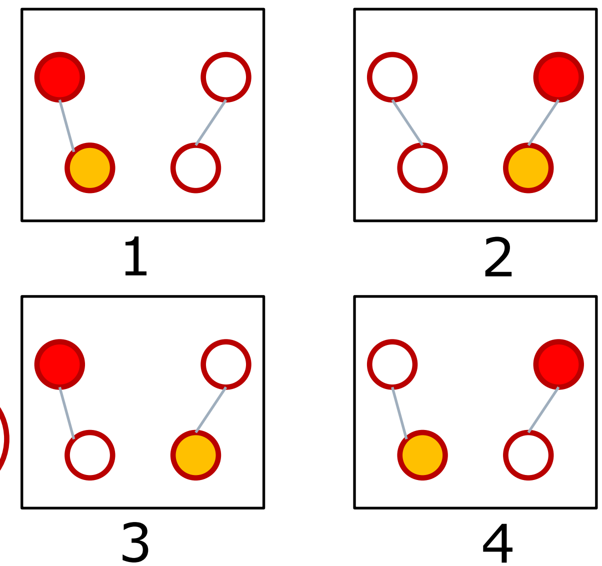
Prob(con(Bob, Alice)) = 100%

K-isomorphism [22]

- K-isomorphism anonymous
  - The graph contains at least k disjoint isomorphism subgraphs

A 2-isomorphism graph

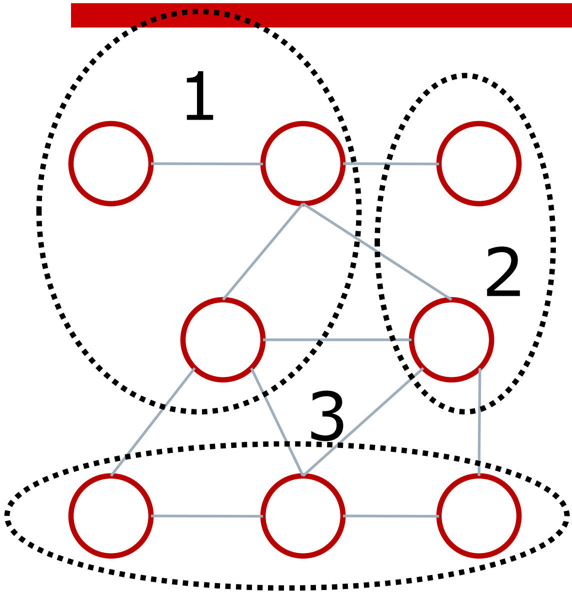


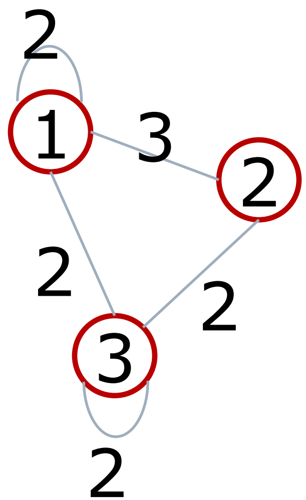


Publishing sanitized graph

1. Privacy protection and the attack models
2. Preventing passive attacks
   1. Edge editing based models
   2. Clustering based models
3. Preventing active attacks
4. Other works

Resist neighborhood attack through graph clustering[8]





Step1: Partition the graph, each partition contains at least k nodes

Step2: For each partition, generate a super node

Step3: Draw the edges between partitions, the weight is the edge number

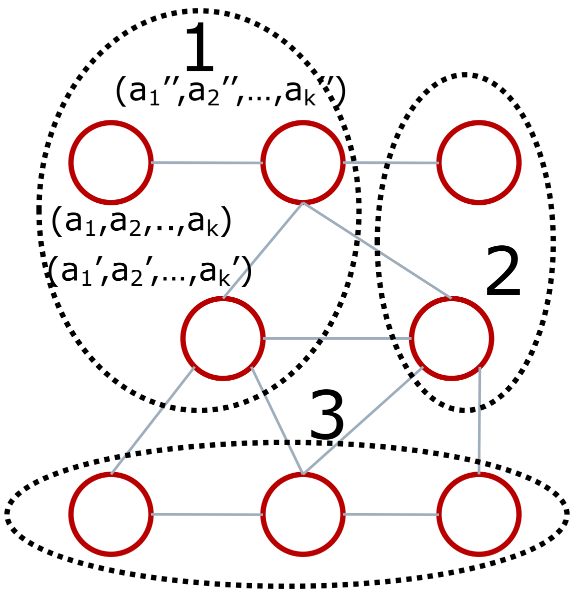
Step3: Draw the sel-edges for each partition, the weight is the edge number with it

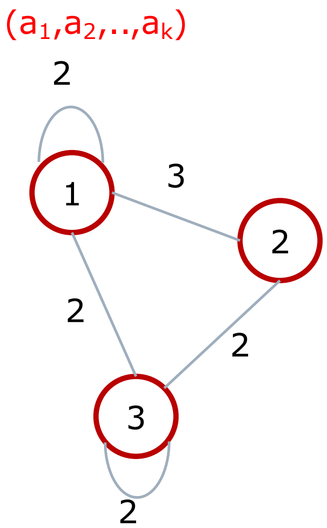
This paper used Simulated Annealing to minimize the number of sampling graphs:

$$|W(G)| = \prod_{X \in V} \binom{\frac{1}{2}|X|(|X|-1)}{d(X,X)} \prod_{X,Y \in V} \binom{|X||Y|}{d(X,Y)}$$

$d(X,Y): No.\ of\ edges\ between\ X\ and\ Y$

K-anonymous masked[15]





A = Generalization Information Lost

B = Structural Information Lost

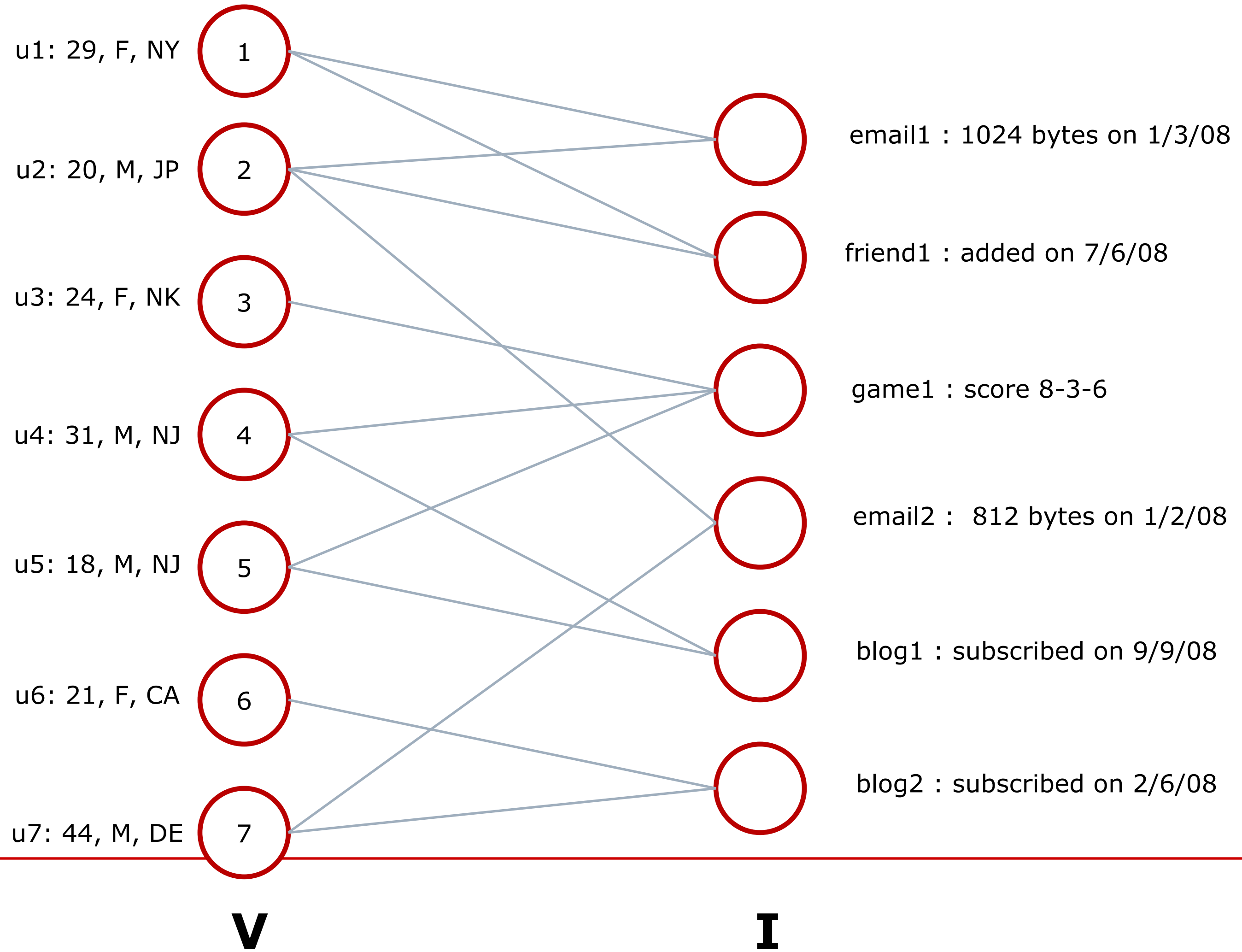
Cost = a*A + b*B

The algorithm is partition the graph into clusters bigger than k by minimizing this cost

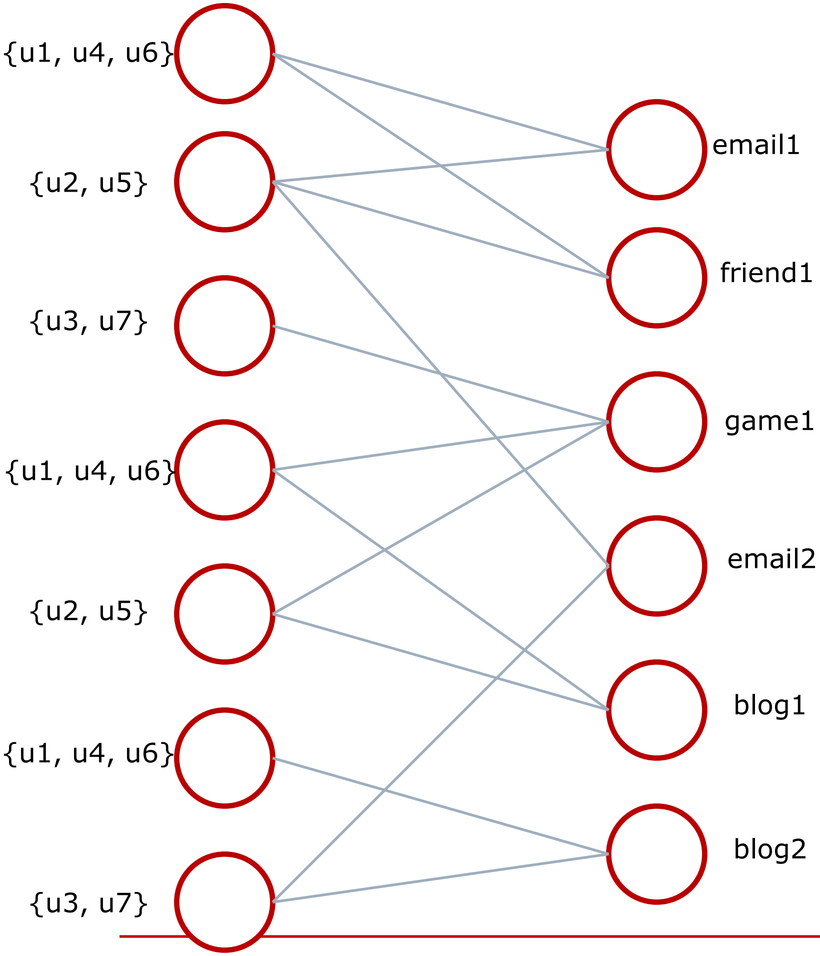
Clustering model for link protection [13]

- Graph Model
  - Undirected bipartite graph (V, I, E)
  - V is a set of users
    - Each user has a group of attributes
  - I is a set of interactions
    - Each interaction can contain more than two users
  - Edge(v, i) means user v is involved in interaction I
- Protect Objectives
  - Node protection: $\Pr ob(u \Rightarrow n) \leq \frac{1}{k}$
  - Link protection 1: $\Pr ob(e(u_1, u_2)) \leq \frac{1}{k}$
    - user x and y are in any interaction together
  - Link protection 2: $\Pr ob(u \in e) \leq \frac{1}{k}$
    - user x is involved in interaction i

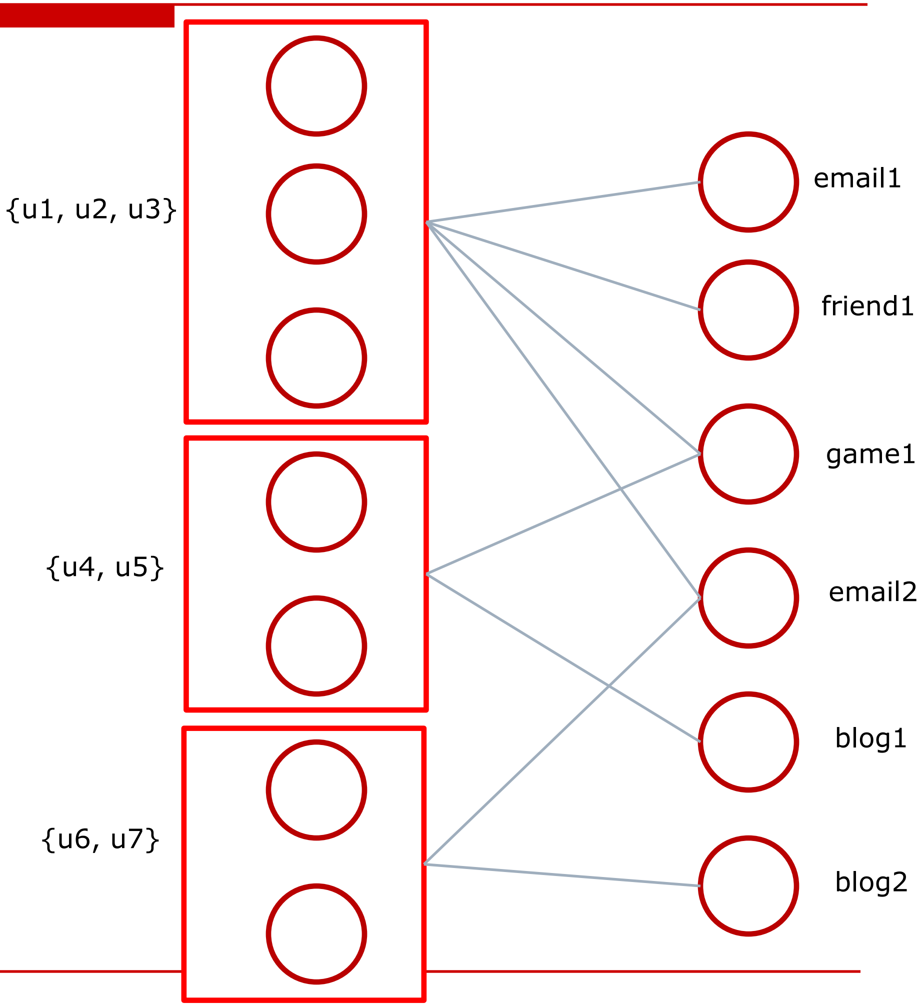
Graph Model Demo



Clustering graphs



Attacks using node attributes



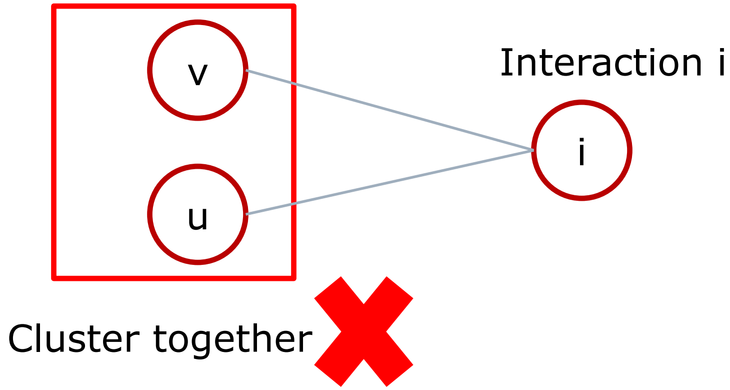
Attacks using node attributes + structure information

Safety Clustering Condition

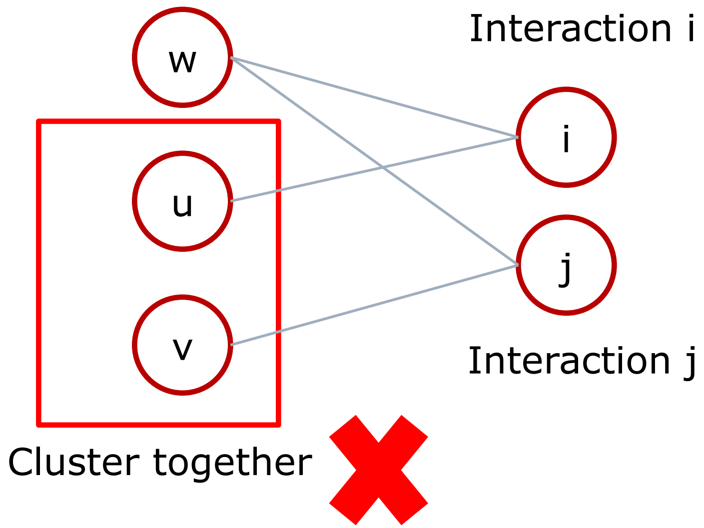
Safety Clustering Condition: $\forall \{u,i\},\{v,i\} \in E : friends(u,v)$

$$\forall u \in S, v \in S \Rightarrow (\neg friends(u,v)) \wedge (\neg \exists w(friends(u,w) \wedge friends(v,w)))$$

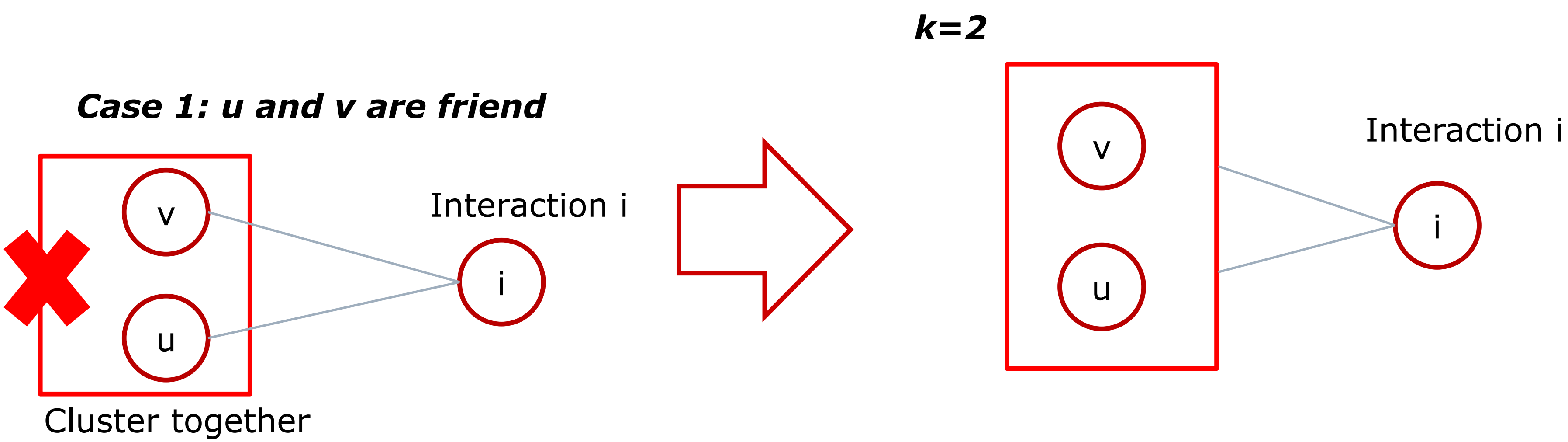
Case 1: u and v are friend



Case 2: u and v are friend



Safety Clustering Condition Cont.

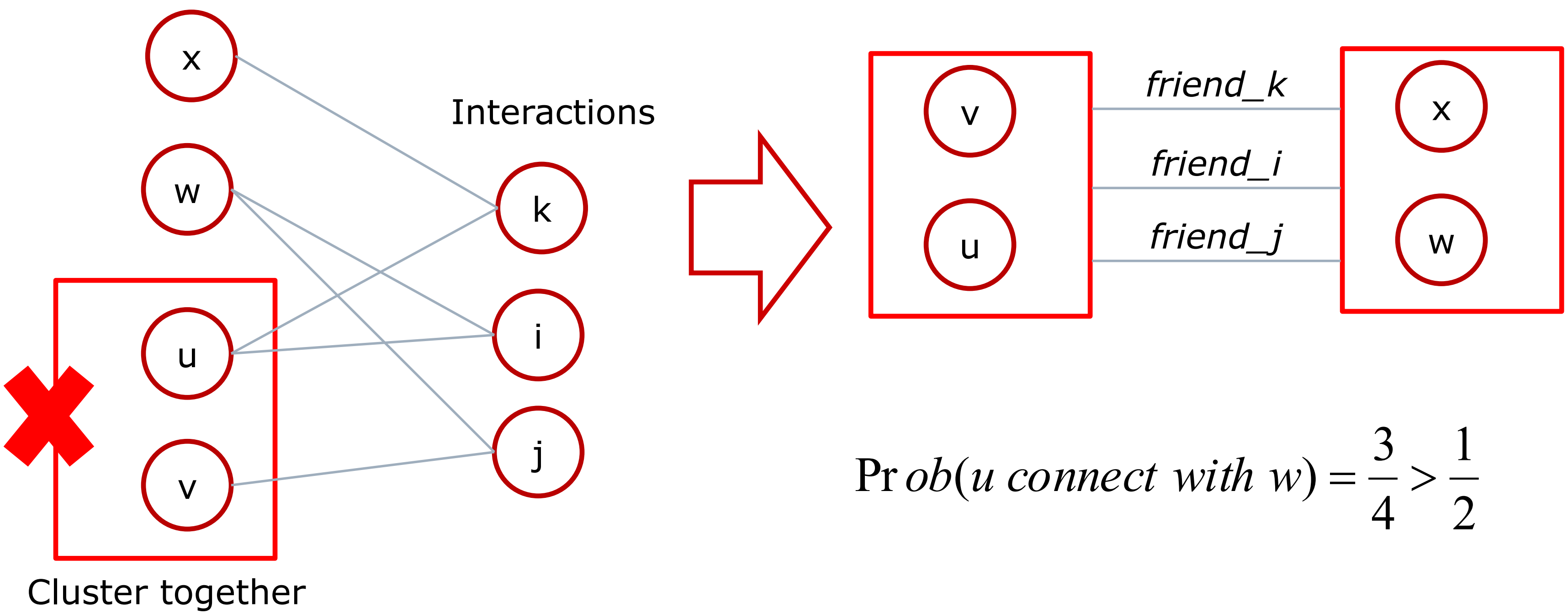


$$\Pr ob(u\ in\ i) = \frac{2}{2} > \frac{1}{2}$$

Safety Clustering Condition Cont.

Case 2: u and v are friend

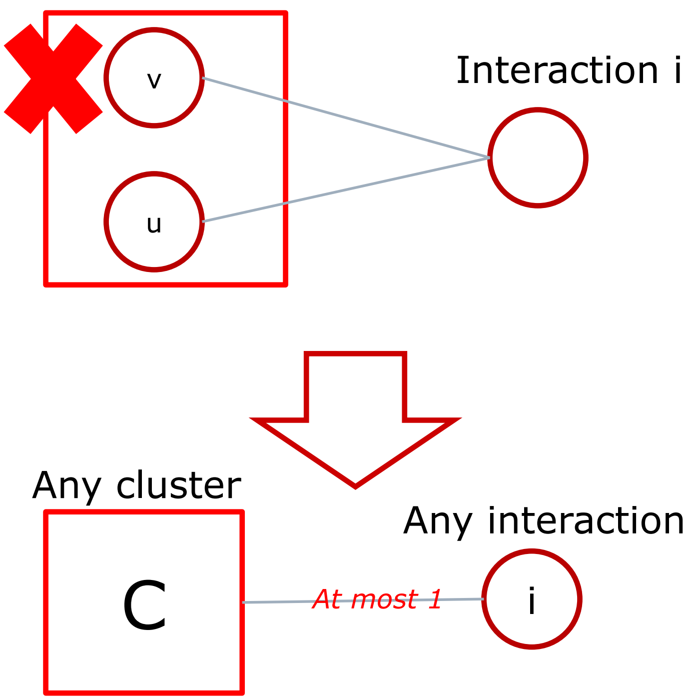
k=2



$$\Pr ob(u\ connect\ with\ w) = \frac{3}{4} > \frac{1}{2}$$

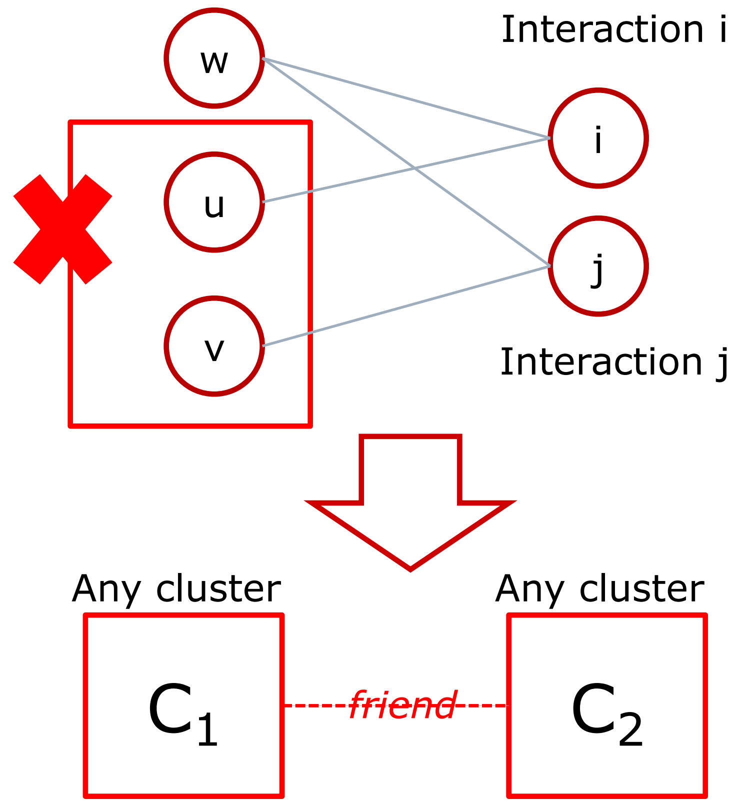
Safety Clustering Condition cont.

Case 1: u and v are friend



$$|C| \geq k \Rightarrow \forall u \in C, \Pr ob(u\ in\ i) \leq \frac{1}{k}$$

Case 2: u and v are friend



Publishing sanitized graph

1. Privacy protection and the attack models
2. Preventing passive attacks
   1. Edge editing based models
   2. Clustering based models
   3. Protecting edge weights
3. Preventing active attacks
4. Other works

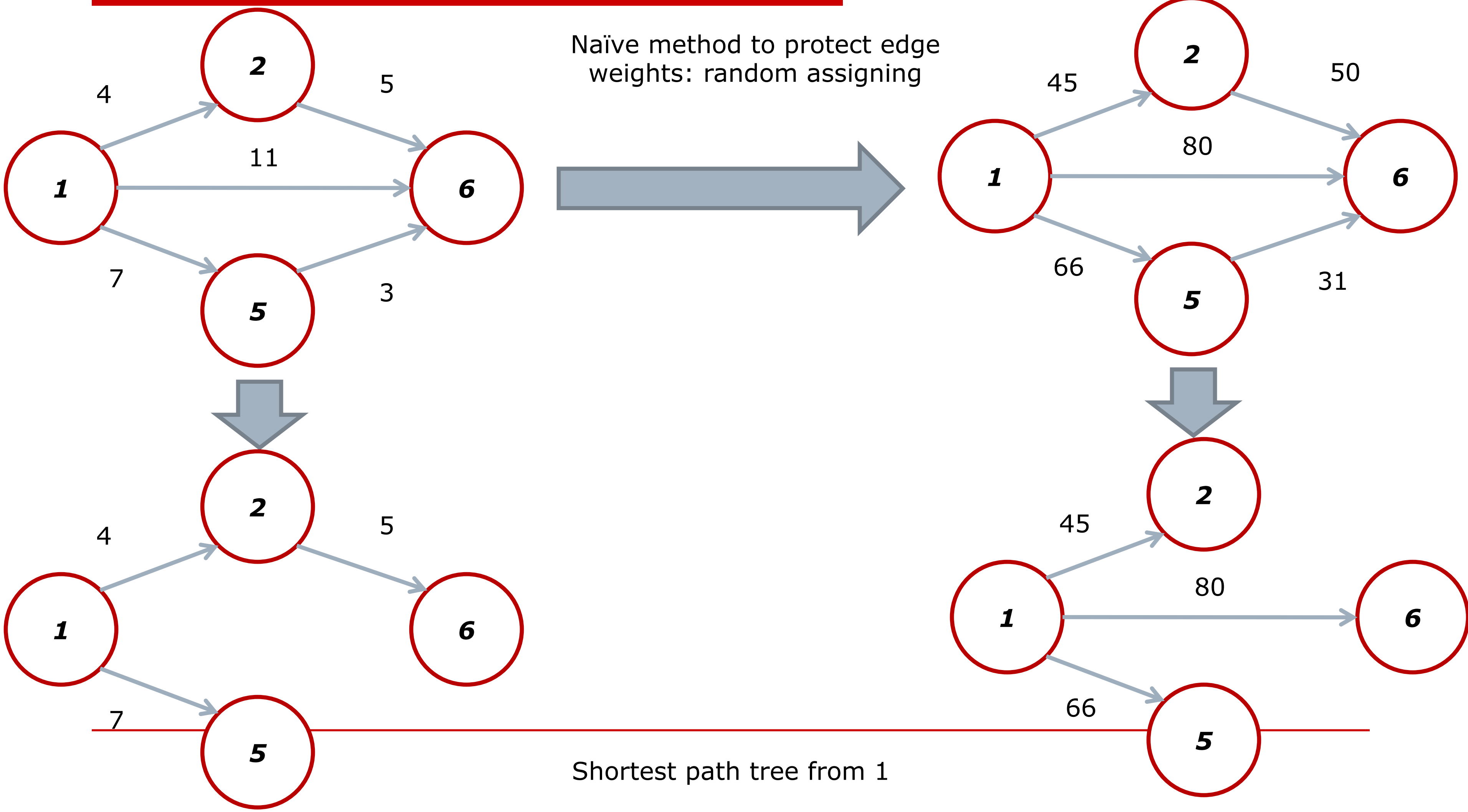
Noised edge weights [17]

- Graph model: weighted graph
- Protection objective
  - Hide the real value of edge weights
- An attacker's background knowledge
  - The published graph
    - Using the edge weights he saw to guess the original weights
- Utility
  - Length of shortest paths
- Method
  - Add gaussian randomization multiplication noise to edge weights
    - Has high probability to preserve the length of shortest paths

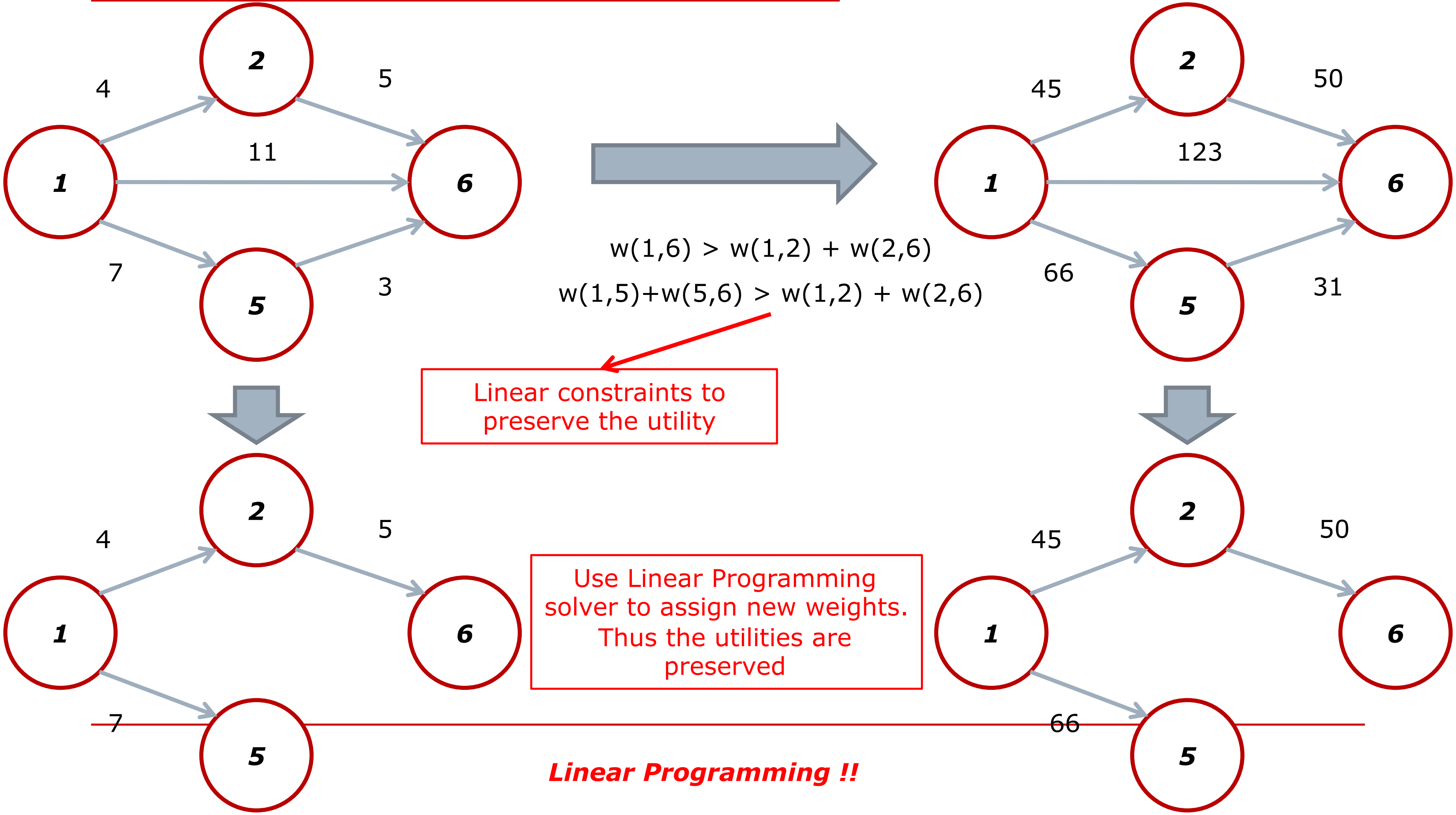
ICDE 10: Anonymous Weighted Graph [24]

- Graph model: weighted graph
- Protection objectives
    - Hide the weights or the orders of the weights
- An attacker’s background knowledge
    - The published graph
- Utility
    - Certain graph metrics (Can be modeled as the linear inequations between edges weights)
        - Single source shortest path tree
        - Some shortest paths

Motivation Example



Motivation Example cont.



Linear Programming !!

Solution Skeleton

Edge weights are variables

LP model

Utility request:

shortest path tree etc.

Linear Constraints

From the construction of this tree

Variables:

w(1,2), w(1,5), w(1,6), w(2,6), w(5,6)

Constraints:

w(1,6) > w(1,2) + w(2,6)

w(1,5)+w(5,6) > w(1,2) + w(2,6)

Variables:

w(1,2) = 45

w(1,5) = 66

w(1,6) = 123

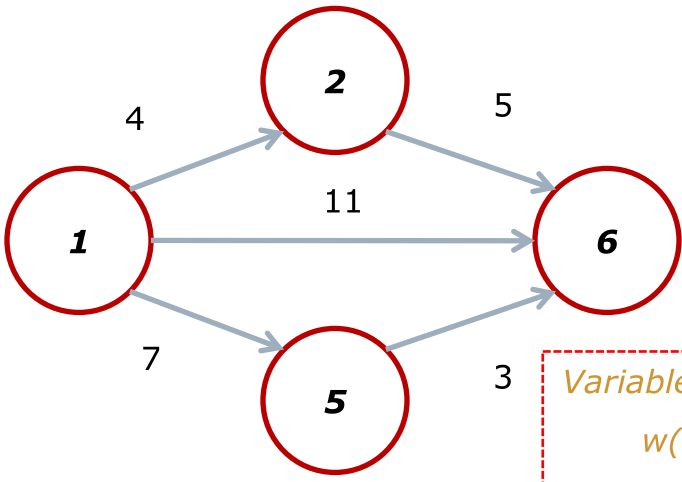
w(2,6) = 50

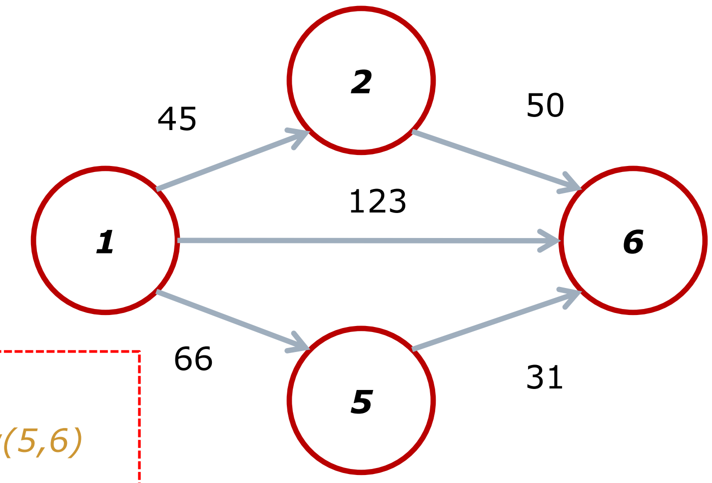
w(5,6) = 31

Solver

All the edge weights

Motivation Example cont.





Variables:

w(1,2), w(1,5), w(1,6), w(2,6), w(5,6)

Constraints:

w(1,6) > w(1,2) + w(2,6)

w(1,5)+w(5,6) > w(1,2) + w(2,6)

LP Solver

Variables:

w(1,2) = 45

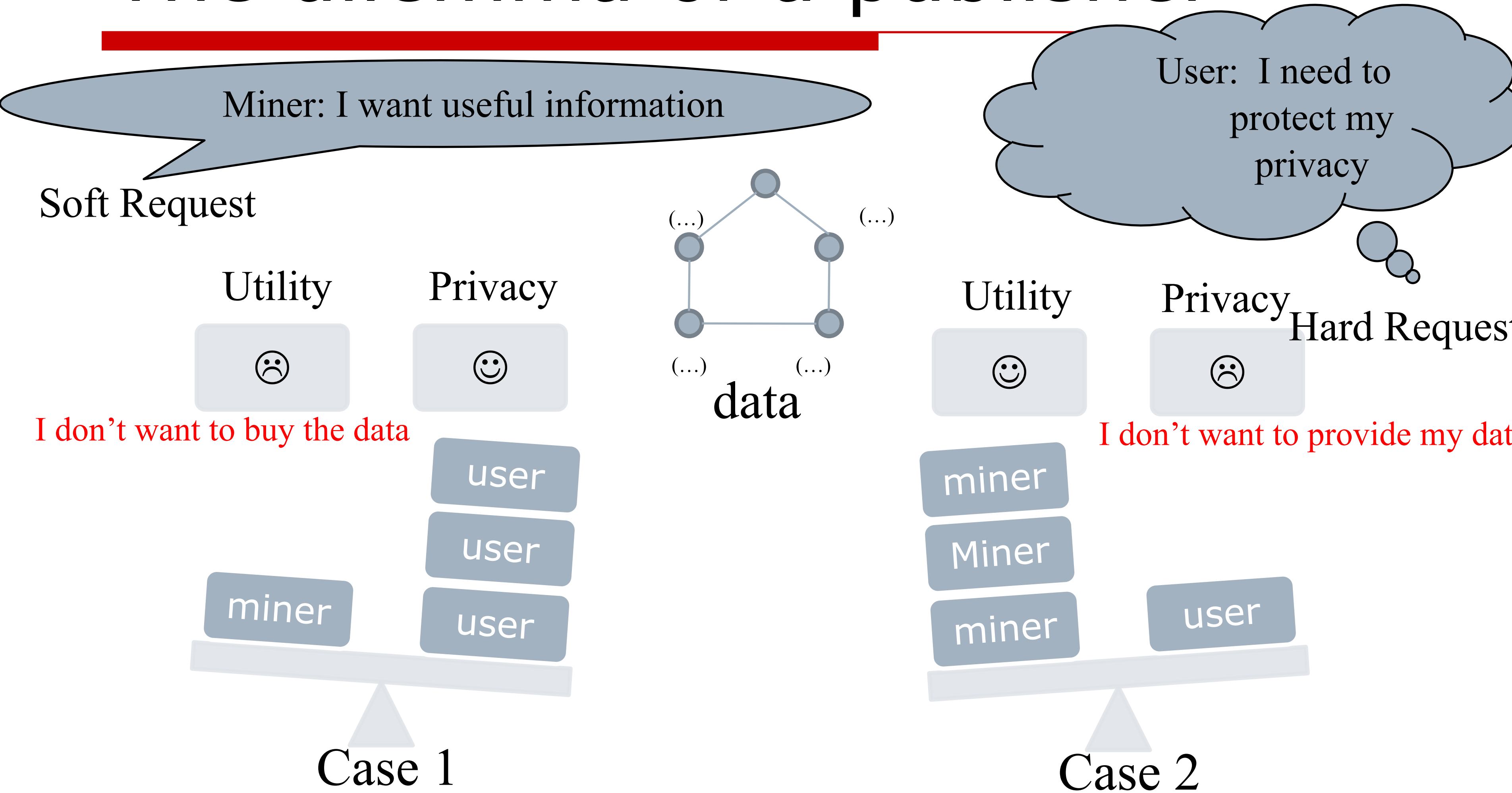
w(1,5) = 66

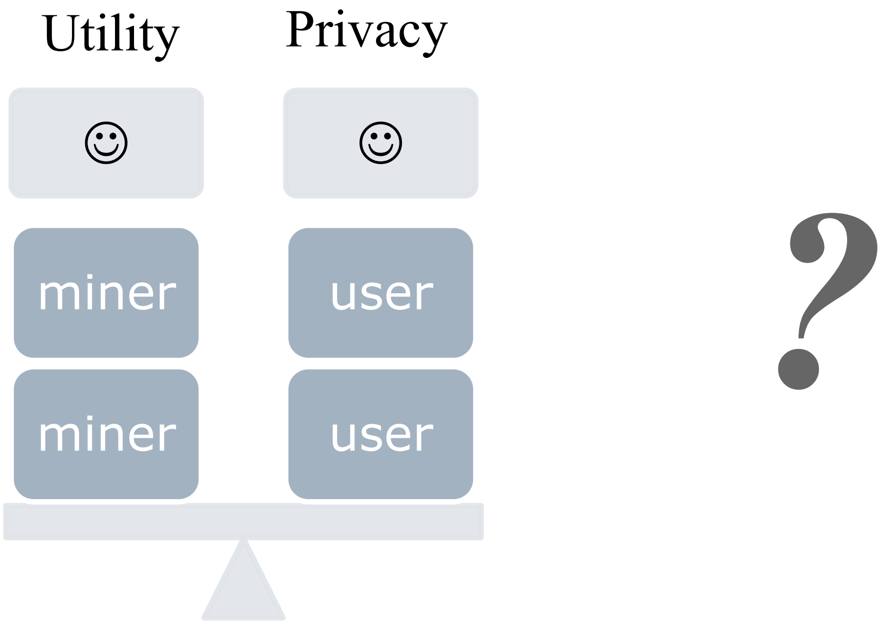
w(1,6) = 123

w(2,6) = 50

w(5,6) = 31

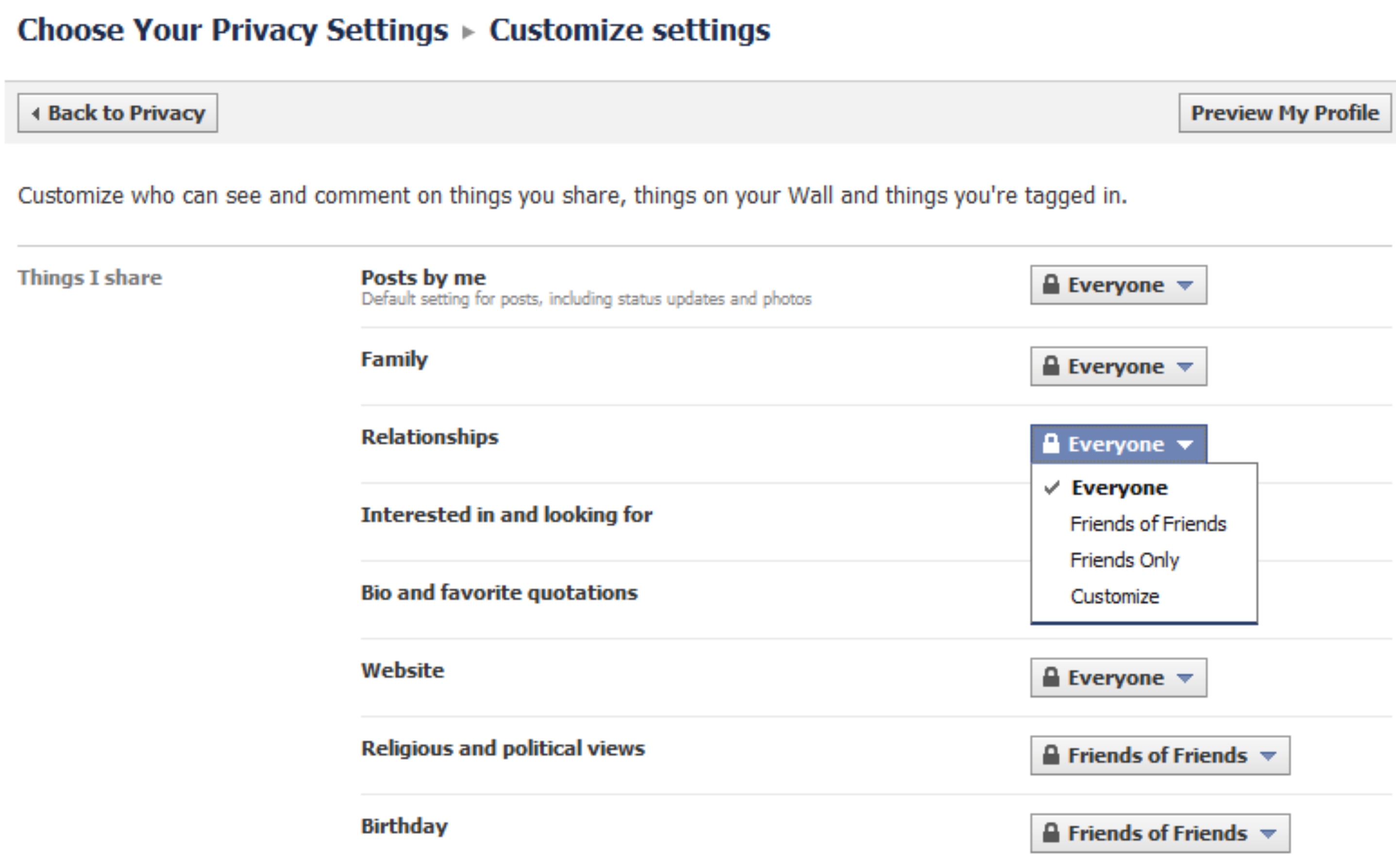
The dilemma of a publisher





Utility
Privacy
☺
☺
miner
user
miner
user
?

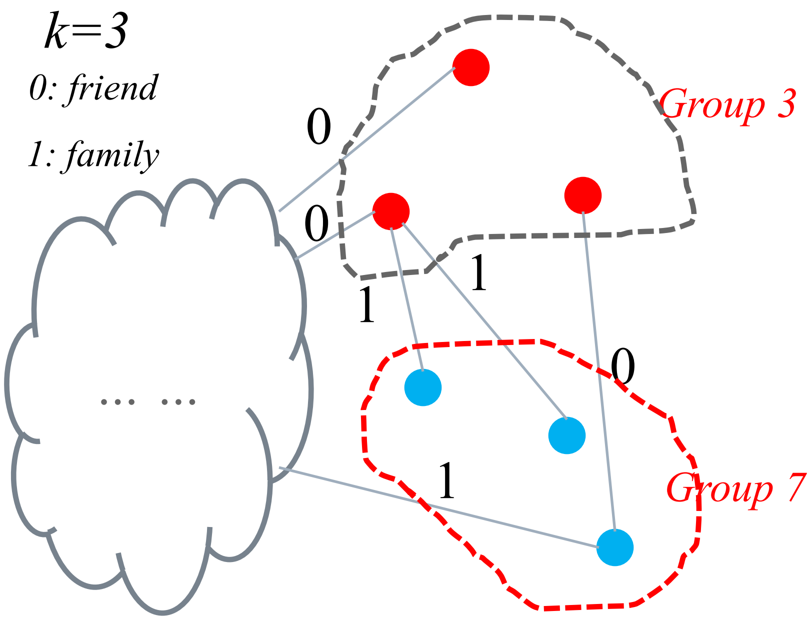
Social Network Websites Support ..



Avoid Attacks Using Knowledge 1

Method: Node protection

Grouping + Node attributes permutation



Group ID	Node attributes
…	…
3	[1] Asian, 33, Phd [2] American, 26, master [3] African, 27, master
…	…
7	[1] European, 29, Phd [2] American, 40, bachelor [3] Australian, 35, bachelor
…	…
M	…

Avoid Attacks Using Knowledge 1

Method: Node protection

Grouping + Node attributes permutation

Edge protection (Two safety conditions)

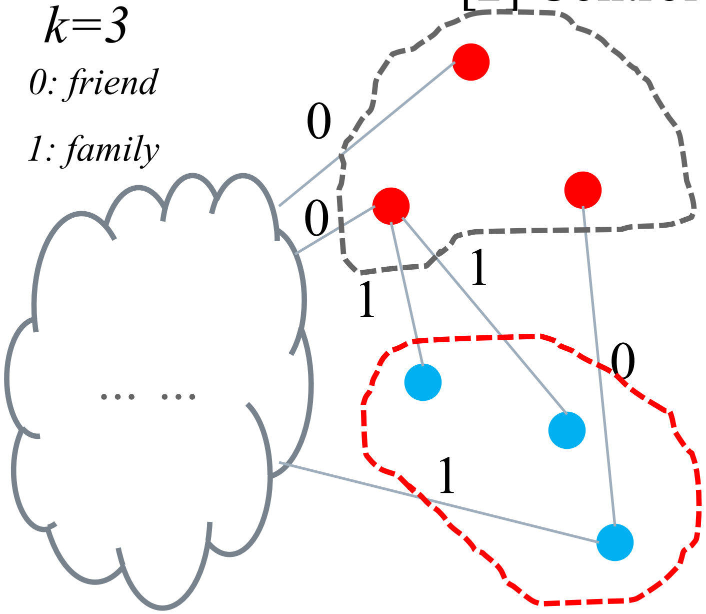
[1] Make sure No edge within a group

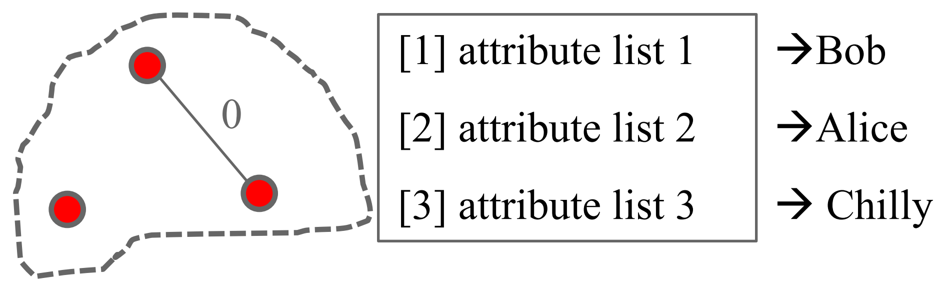
[2] Control the number of edges between groups $\frac{d}{|X||Y|} \leq \frac{1}{k}$

k=3

0: friend

1: family





Each user has 2/k probability to have this edge

Avoid Attacks Using Knowledge 1

Method: Node protection

Grouping + Node attributes permutation

Edge protection (Two safety conditions)

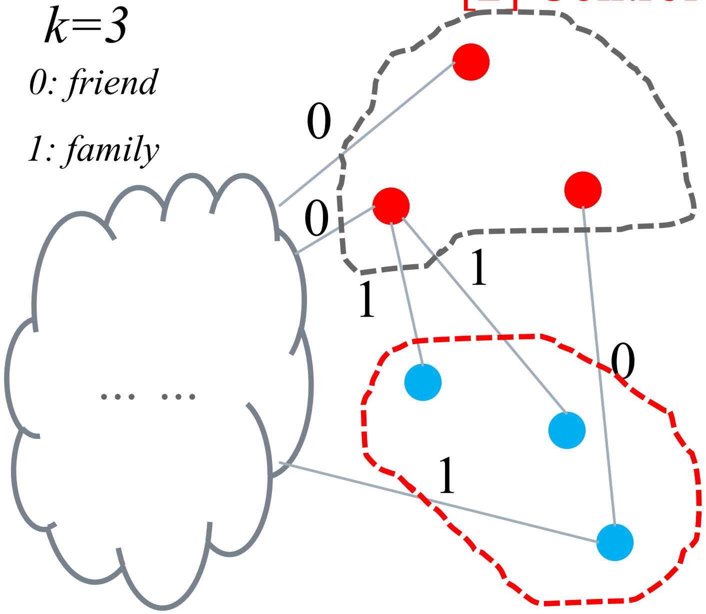
[1] Make sure No edge within a group

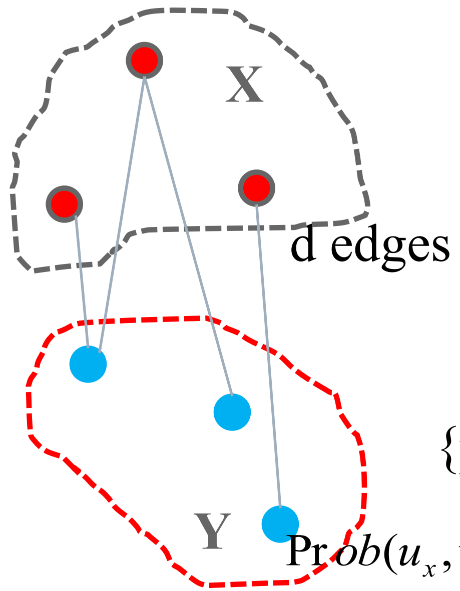
[2] Control the number of edges between groups $\frac{d}{|X||Y|} \le \frac{1}{k}$

k=3

0: friend

1: family





$\{p_1, p_2, p_3\}$

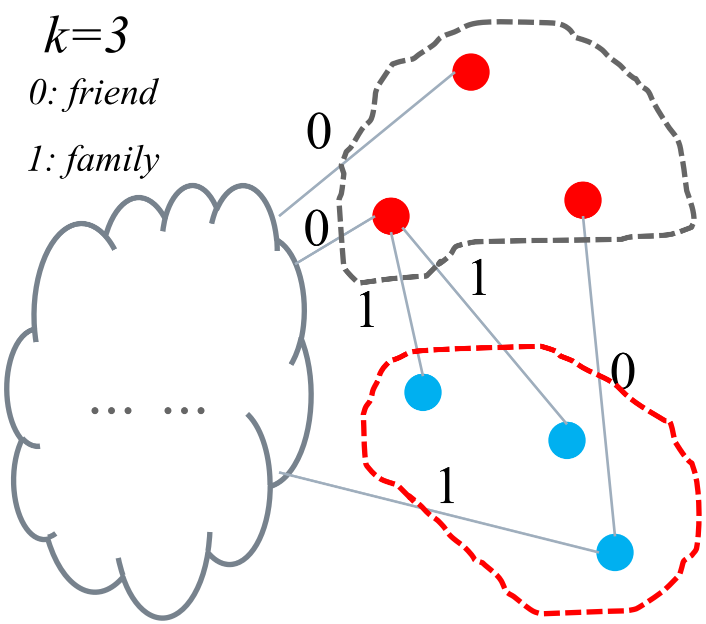
$\{p_4, p_5, p_6\}$

$$Prob(u_x, v_y) = \frac{d}{|X||Y|} \le \frac{1}{k}$$

Avoid Attacks Using Knowledge 2

Objective: For any group that contains one node need this level's protection, make all the nodes in it have the same degree

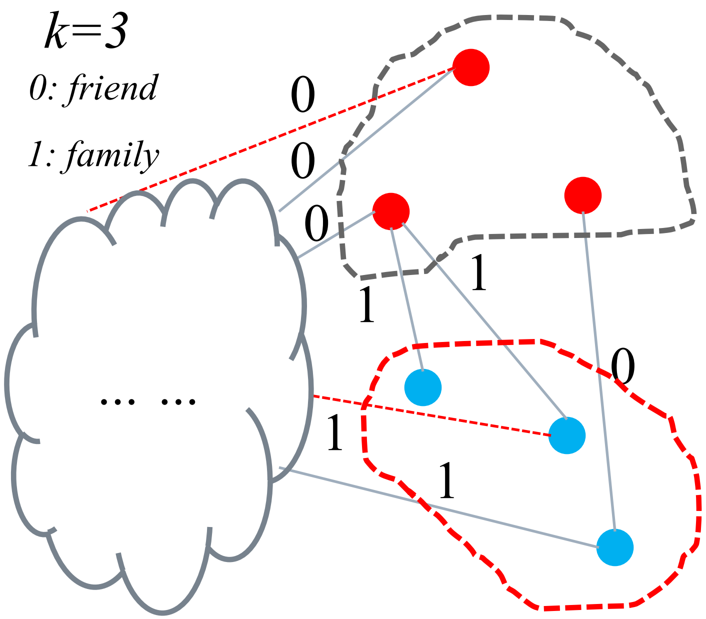
Method: Add noise edges/nodes under the two safety conditions



Avoid Attacks Using Knowledge 2

Objective: For any group that contains one node need this level's protection, make all the nodes in it have the same degree

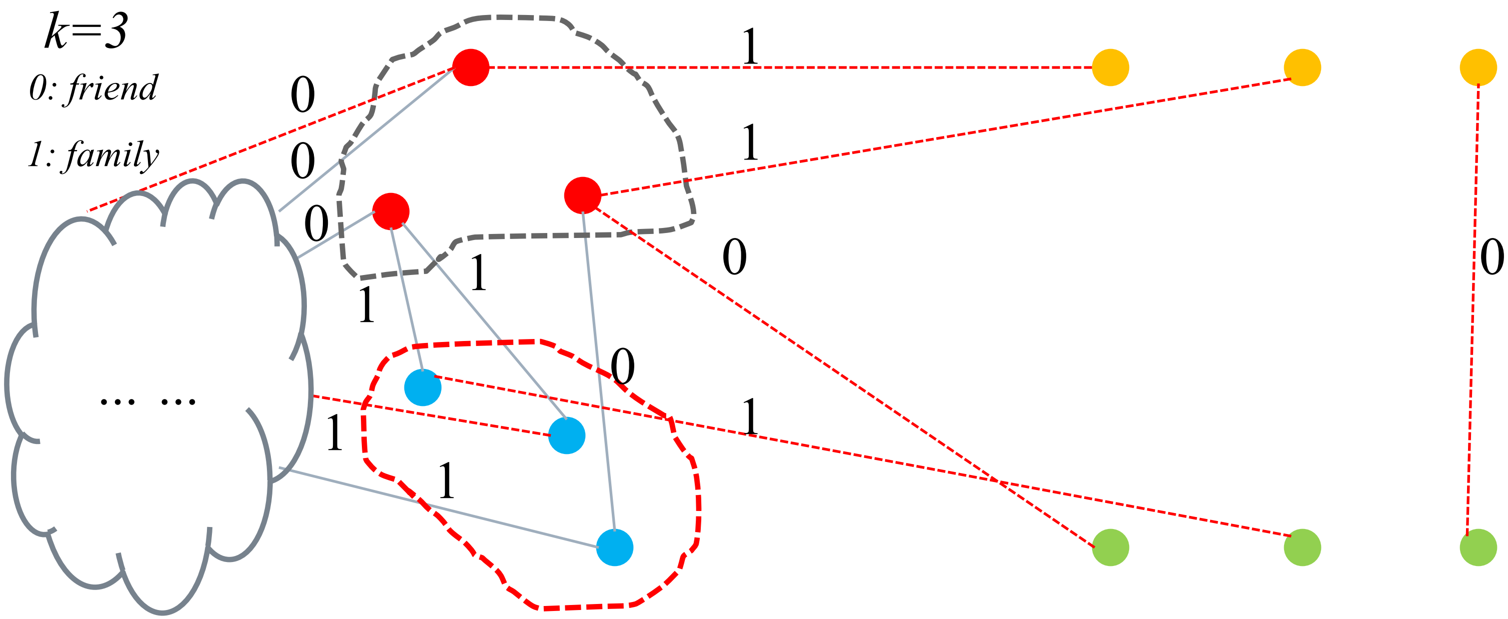
Method: Add noise edges/nodes under the two safety conditions



Avoid Attacks Using Knowledge 2

Objective: For any group that contains one node need this level's protection, make all the nodes in it have the same degree

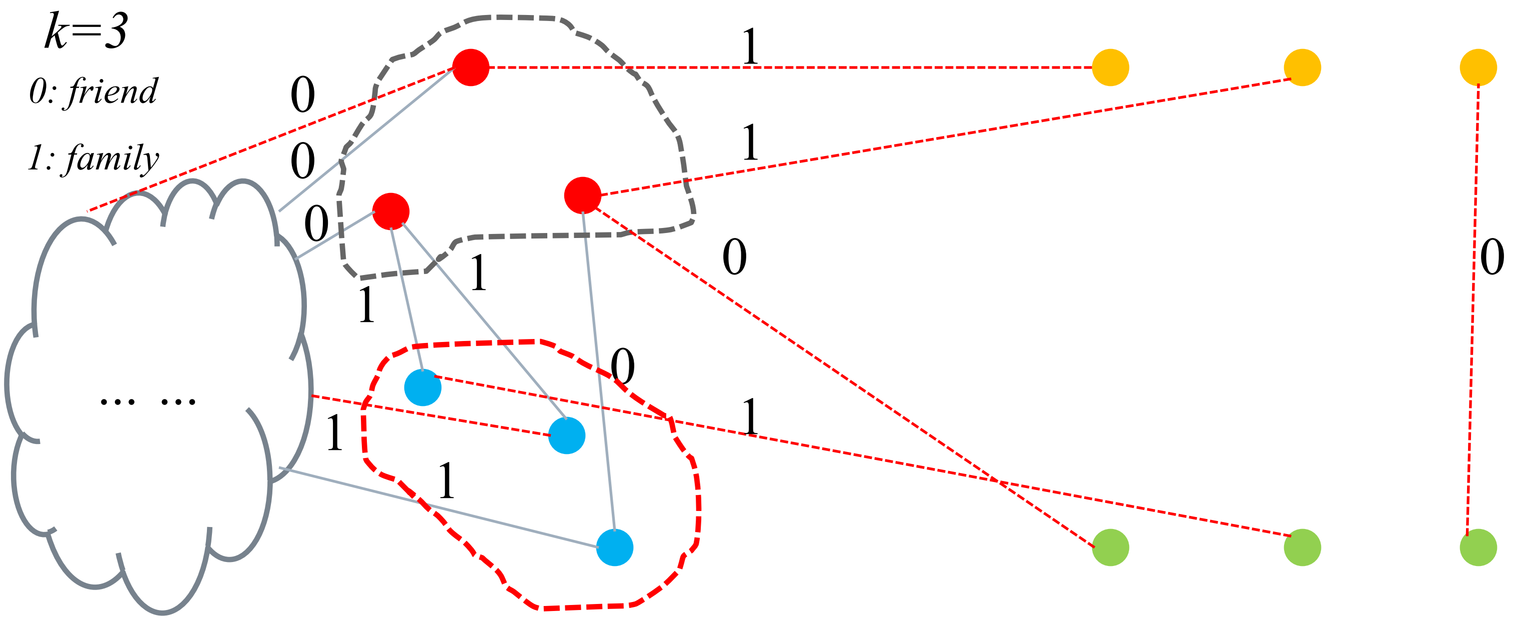
Method: Add noise edges/nodes under the two safety conditions



Avoid Attacks Using Knowledge 3

Objective :For any group that contains one node need this level's protection, make all the nodes in it have the same degree label sequence

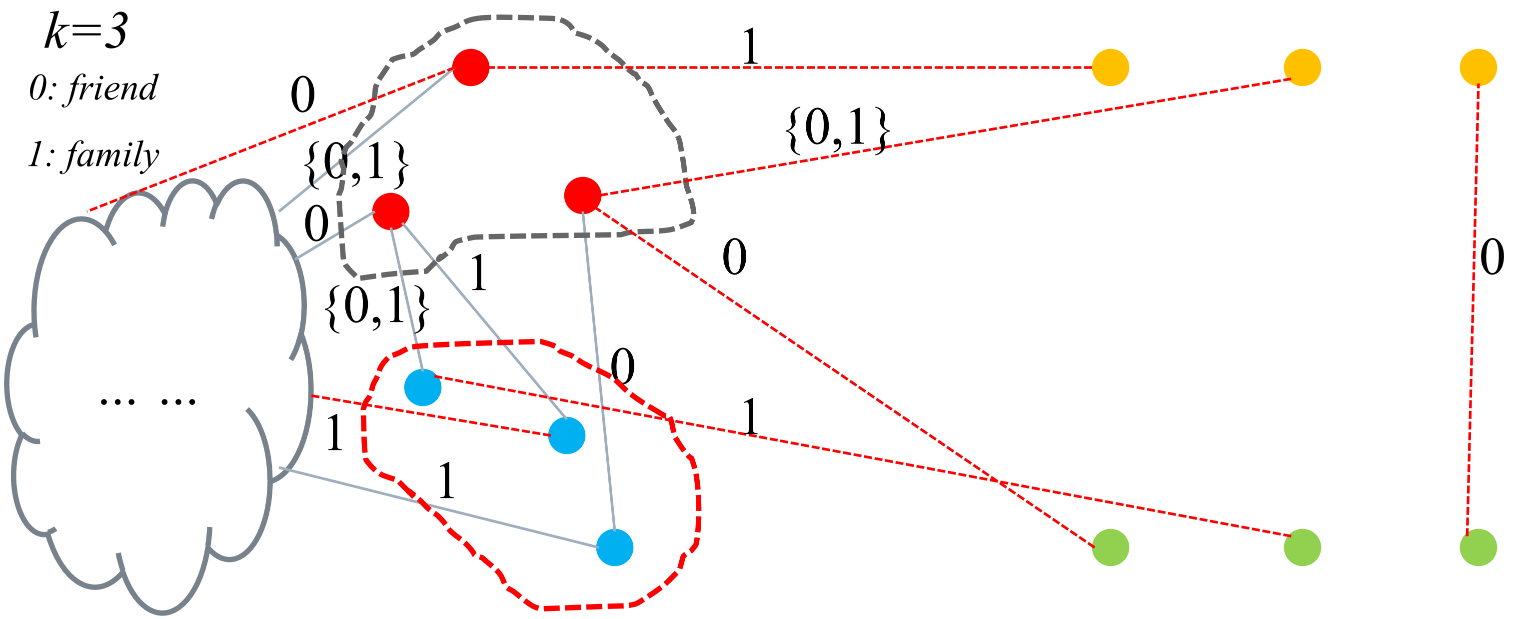
Method: Generalize the edge labels



Avoid Attacks Using Knowledge 3

Objective :For any group that contains one node need this level's protection, make all the nodes in it have the same degree label sequence

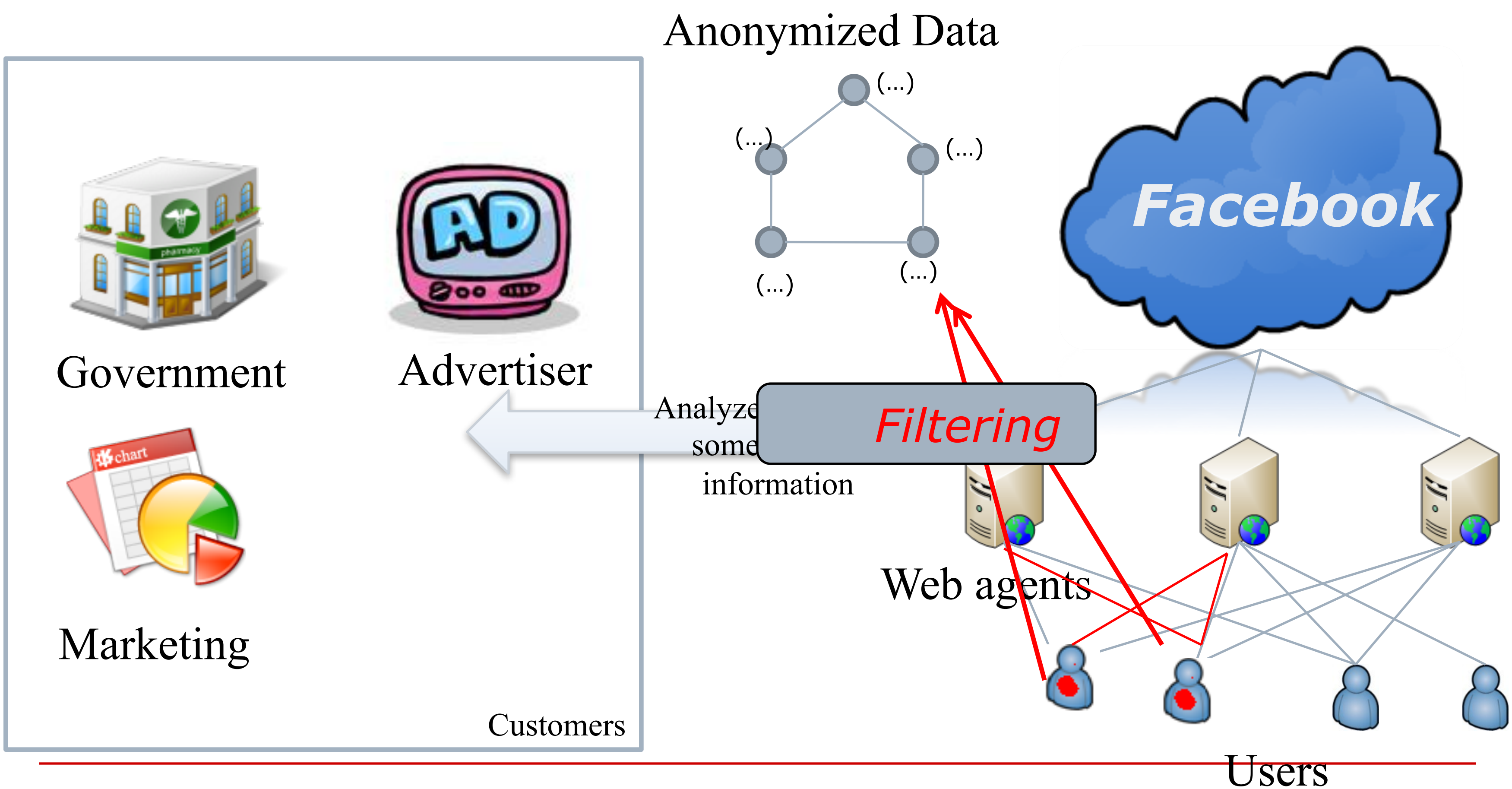
Method: Generalize the edge labels



Publishing sanitized graph

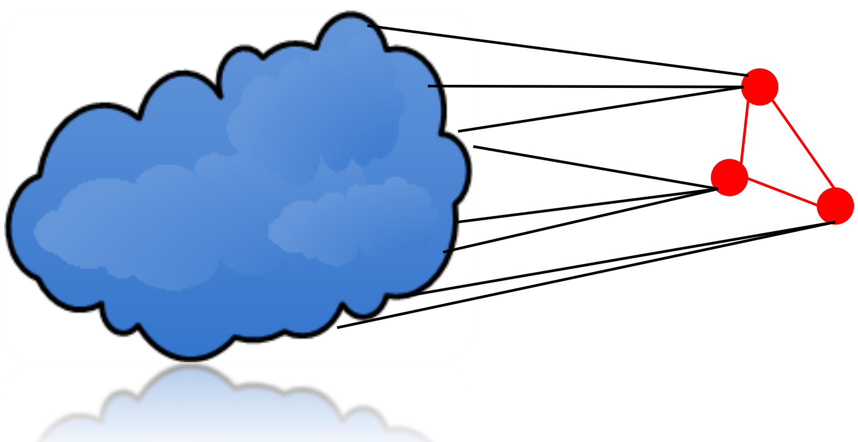
1. Privacy protection and the attack models
2. Preventing passive attacks
3. Preventing active attacks
4. Other works

Anti Active attack

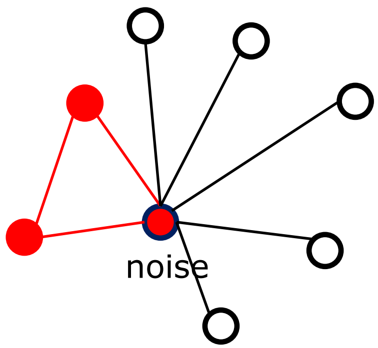


RLA on email networks [11]

- Random link attack (RLA)
  - A group of noise nodes
    - Form communities themselves
      - Preventing to be filtered as outlier nodes
    - Randomly link to a large number of victims



Observations

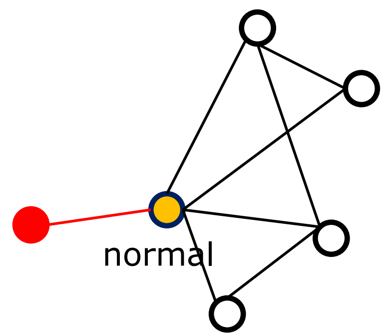


[1] Victims are randomly selected

Most of their friends do not have connection

Cluster Coefficient is small

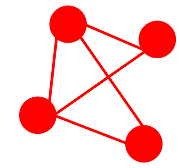
Triangle ratio is very low



Most of its friends know each other

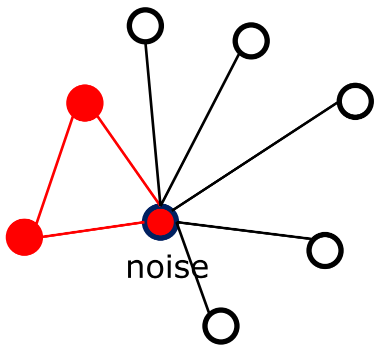
Cluster Coefficient is large

Triangle ratio is high



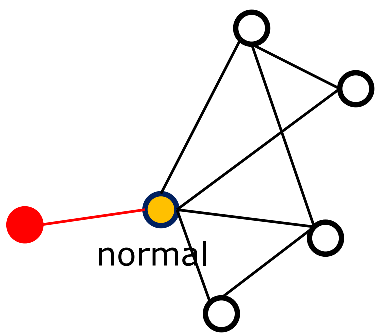
The noise nodes
form communities

Two step filtering



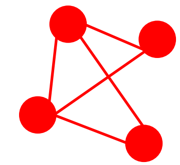
Cluster Coefficient is small

Triangle ratio is very low



Cluster Coefficient is large

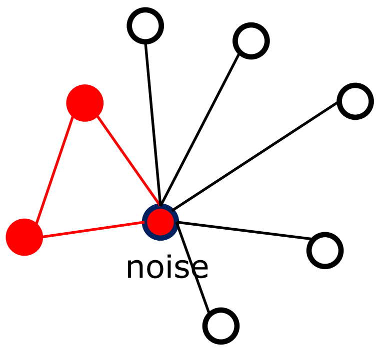
Triangle ratio is high



The noise nodes form communities

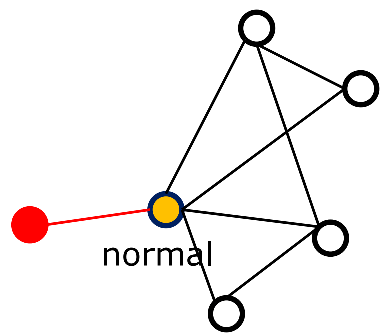
Find suspects → Filtering the densely connected nodes around all suspects

Two step filtering



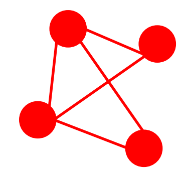
Cluster Coefficient is small

Triangle ratio is very low



Cluster Coefficient is large

Triangle ratio is high

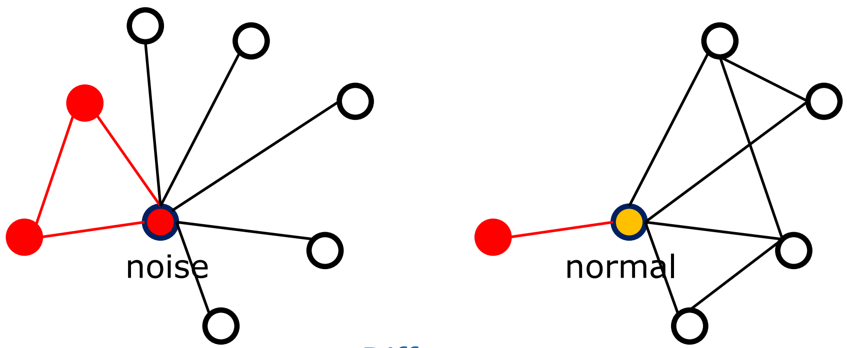


The noise nodes form communities

Find suspects

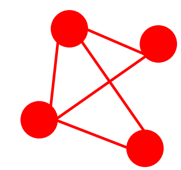
Filtering the densely connected nodes around all suspects

Find suspects by spectral characteristics [25]



Spectral values — Different — Spectral values

i.e. the values eigen vectors — i.e. the values eigen vectors



The noise nodes form communities

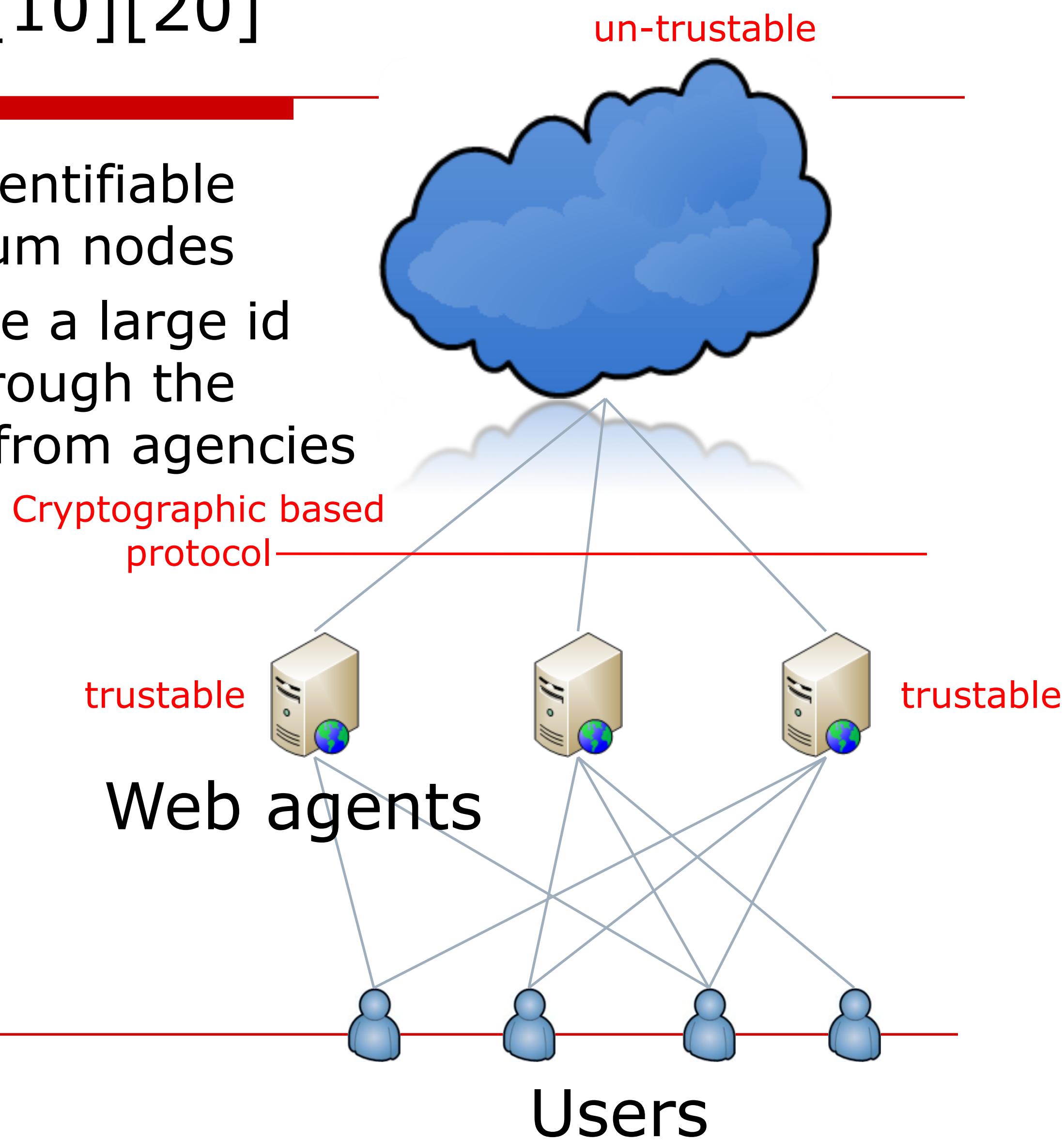
Find suspects → Filtering the densely connected nodes around all suspects

Publishing sanitized graph

1. Privacy protection and the attack models
2. Preventing passive attacks
3. Preventing active attacks
4. Other works

Other Works[10][20]

- How to embed a re-identifiable subgraph with minimum nodes
- How to safely compose a large id anonymized graph through the sub-graphs gathered from agencies



Outline

- Information Sharing in On-line Social networks
- Understanding Your Privacy Risk
- Managing Your Privacy Control

Outline

- Information Sharing in On-line Social networks
- Understanding Your Privacy Risk
  - Privacy risk due to what you shared explicitly
- Managing Your Privacy Control

Privacy risk due to what you shared explicitly

- Basic Idea
  - Privacy risk is measured by Privacy Score[1]
  - Privacy Score takes into account what information you've shared and who can view that information
- Basic Premises of Privacy Score
  - Sensitivity
    - The more sensitive the information revealed by a user, the higher his privacy risk
  - Visibility
    - The wider the information about a user spreads, the higher his privacy risk

The framework

name, or gender, birthday, address, phone number, degree, job, etc.

Privacy Score of User *j* due to Profile Item *i*

$$PR(i,j) = \beta_i \times V(i,j).$$

sensitivity of profile item *i*

visibility of profile item *i*

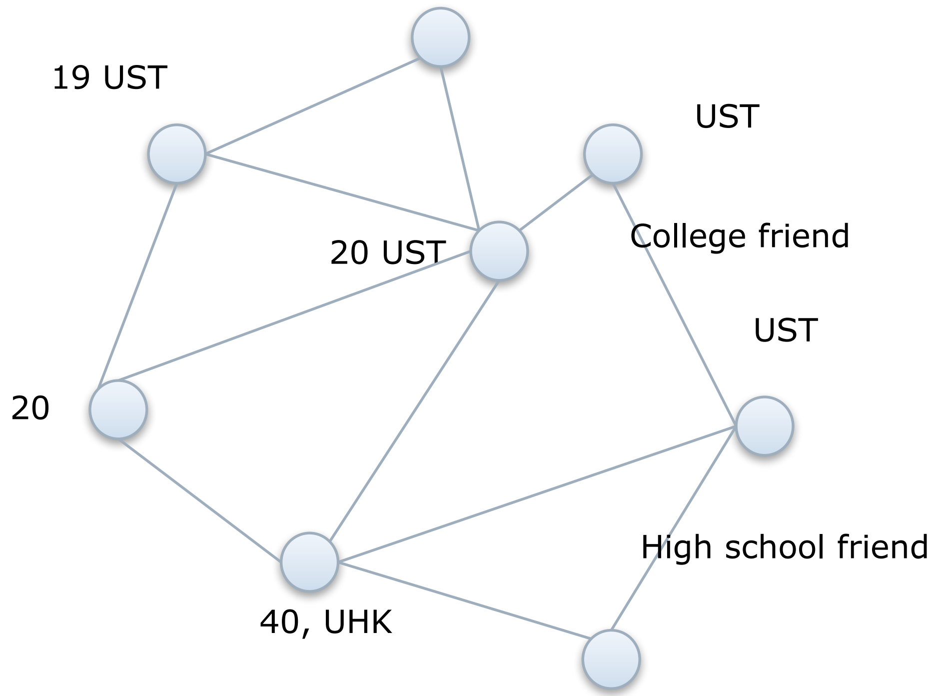
Overall Privacy Score of User *j*

$$PR(j) = \sum_i PR(i,j) = \sum_i \beta_i \times V(i,j).$$

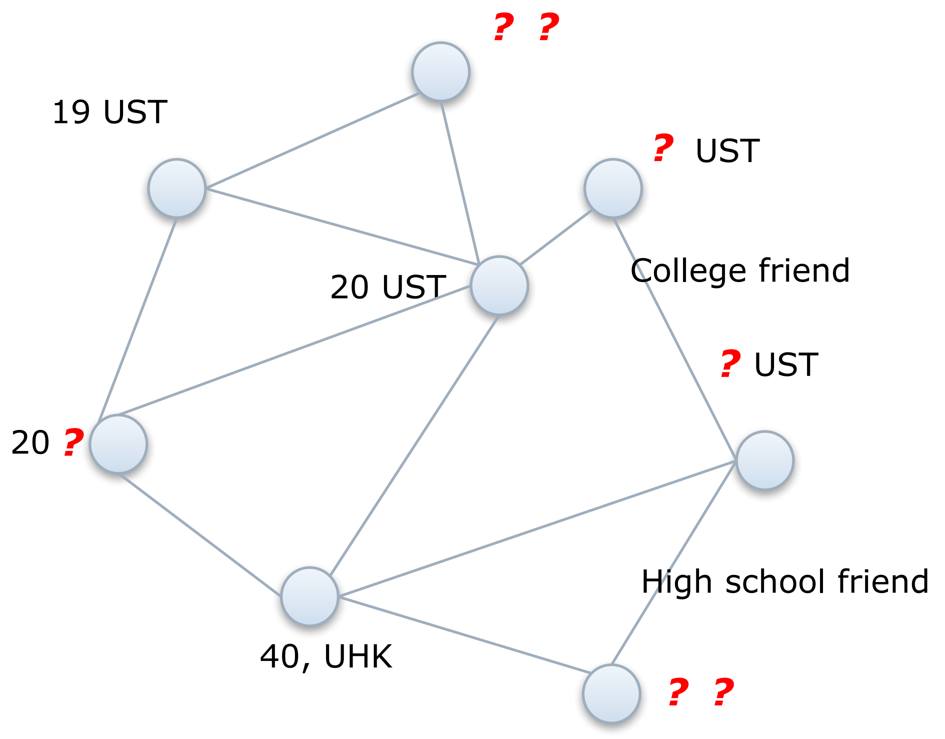
Outline

- Information Sharing in On-line Social networks
- Understanding Your Privacy Risk
  - Privacy risk due to what you shared explicitly
  - Privacy risk due to what you shared implicitly
- Managing Your Privacy Control

What is node classification?



What is node classification?



Privacy Risk due to What You Shared Implicitly

- Privacy information can be inferred from
  - Your public profile, friendships, group memberships, etc.
- Private information can be inferred using
  - Majority voting[1][2]
  - Community detection[3]
  - Classification[1][4]

Classification Methods

- Naive Method
  - Based on network distribution
- Local Classification Methods
  - Based on friendship links
    - AGG, BLOCK, LINK
  - Based on social groups
    - CLIQUE, GROUP, GROUP*
  - Based on both links and groups
    - LINK-GROUP
  - Iterative Classification Method (CC)
- Random Walk Based Methods

Outline

- Information Sharing in On-line Social networks
- Understanding Your Privacy Risk
  - Privacy risk due to what you shared explicitly
  - Privacy risk due to what you shared implicitly
  - Tools to visualize your privacy policies
- Managing Your Privacy Control

Outline

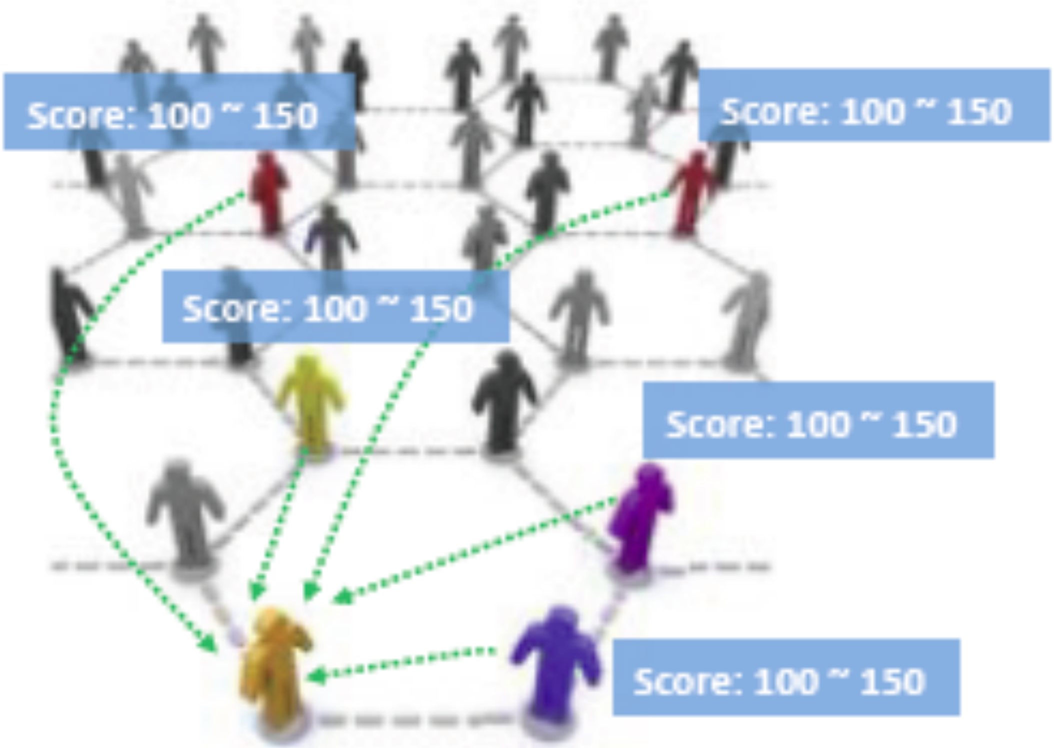
- Information Sharing in On-line Social networks
- Understanding Your Privacy Risk
  - Privacy risk due to what you shared explicitly
  - Privacy risk due to what you shared implicitly
  - Tools to visualize your privacy policies
- Managing Your Privacy Control

Privacy Management of Individuals

- Social Navigation[1][7]
- Preventing Inference Attacks[4]
- Learning Privacy Preferences with Limited User Inputs[8][9]

Social Navigation

Social navigation helps users make better privacy decisions using community knowledge and expertise.



Preventing Inference Attacks

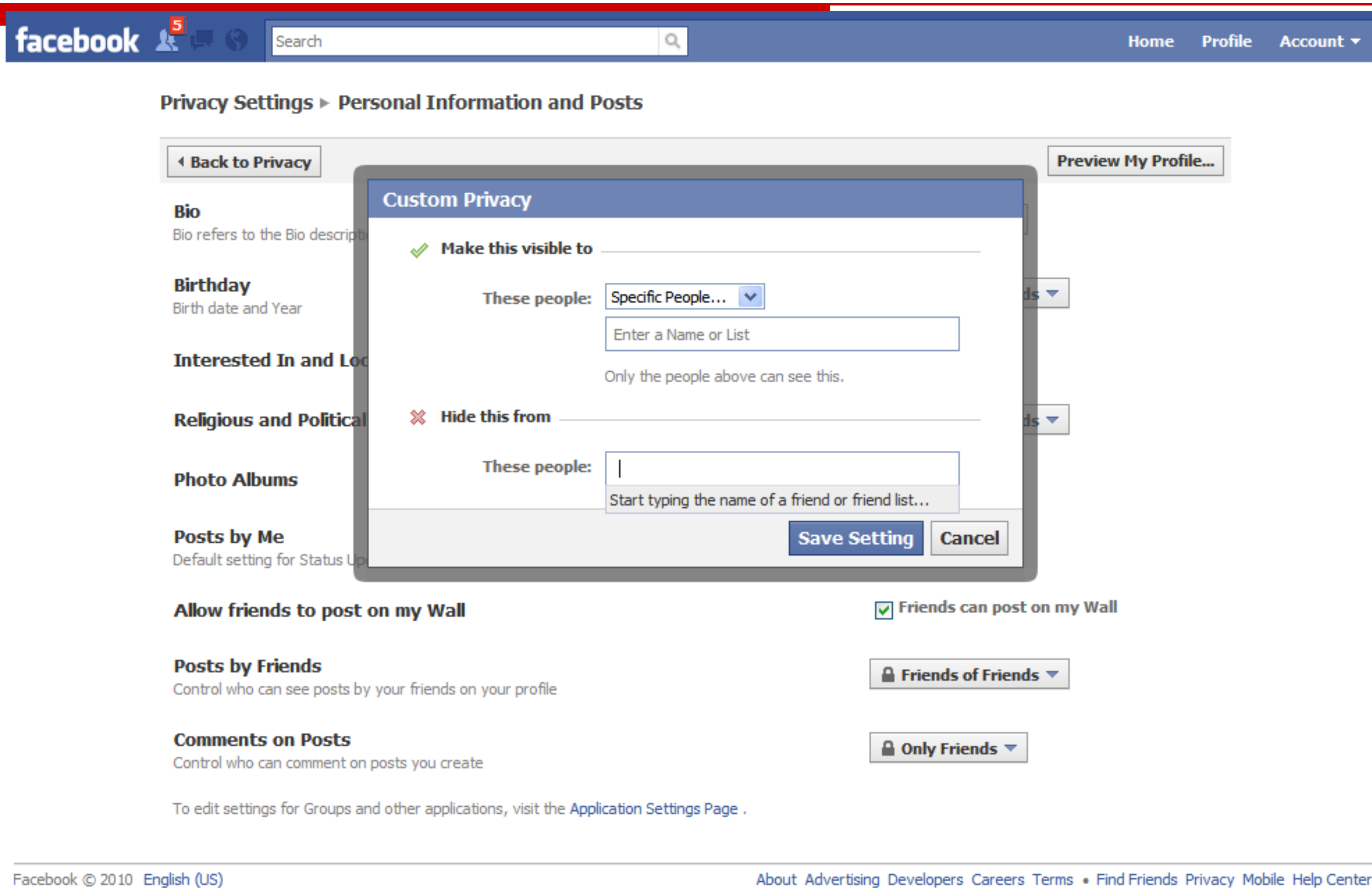
Remove/hide risky links, profiles or groups that contributed most to the inference attacks.

Pr(political views = 'conservative' | group = ~~'texas conservatives'~~, ~~edge$_{AB}$~~, ~~edge$_{AC}$~~, ~~edge$_{AD}$~~)

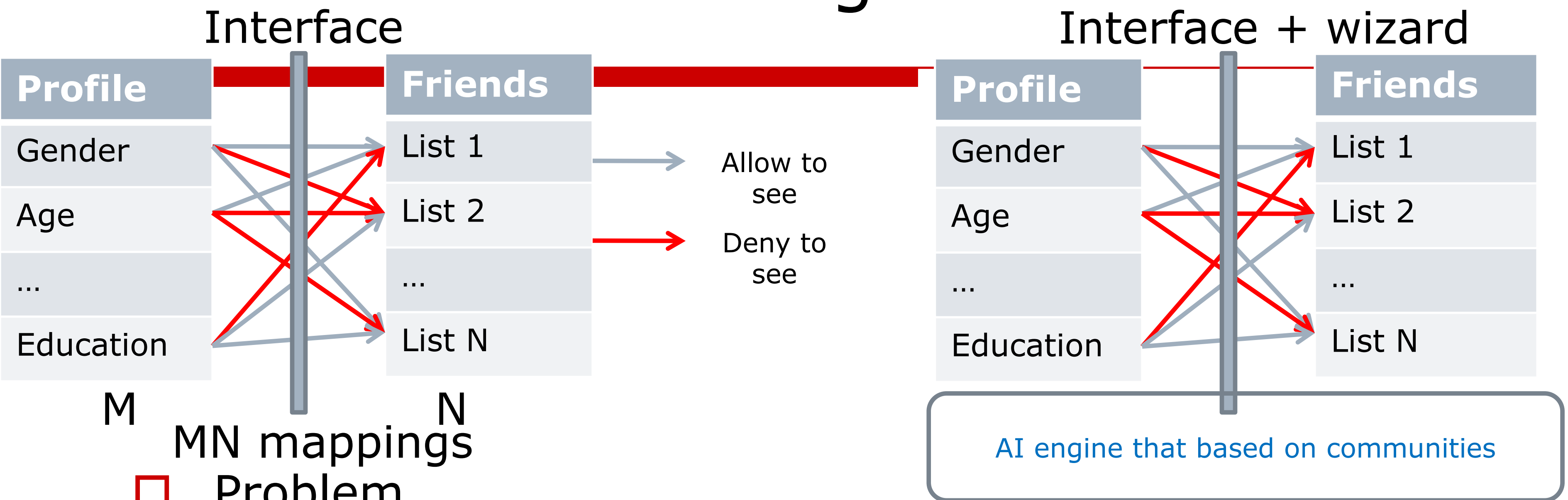
Learning Privacy Preferences

- Privacy Wizards for Social Networking Sites
  - Best student paper in WWW 10

Privacy preference setting in Facebook



Problem and Challenges



- Problem
  - User need to manually create user lists
  - Too many friends (average 130 in facebook)
  - Users are not familiar with complex privacy rules

Time consuming

- Solution: a privacy wizard based on an implicit set of rules
- Challenges
  - Low Effort, High Accuracy
  - Graceful Degradation
  - Visible Data

Basic observation

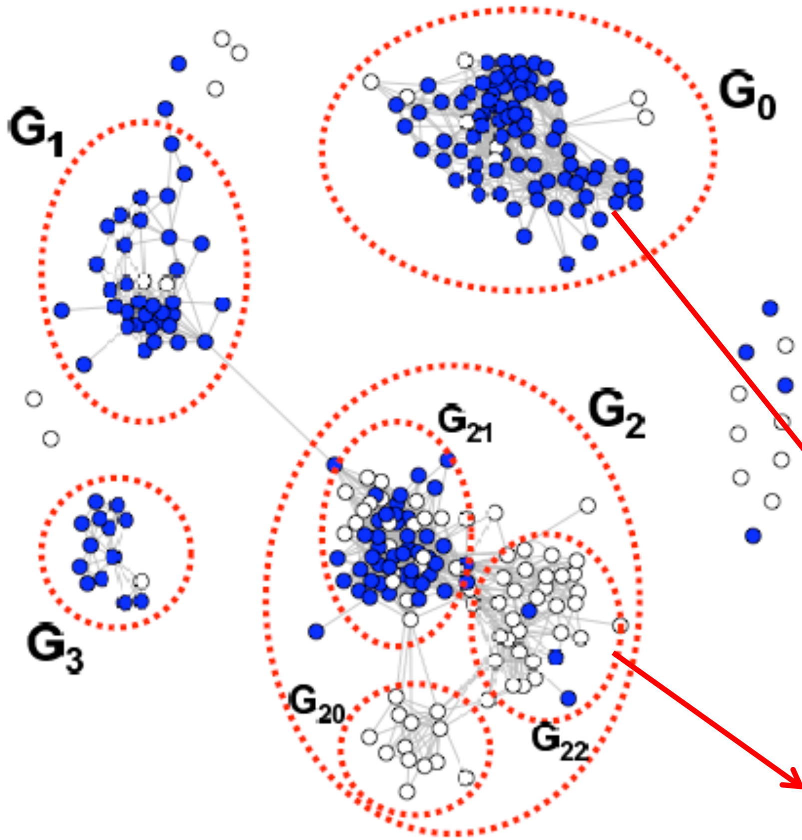


Figure 1: User K's neighborhood graph, and her privacy preferences toward Date of Birth. (Shaded nodes indicate *allow*, and white nodes indicate *deny*.) Notice that K's privacy preferences are highly correlated with the *community* structure of the graph.

- The privacy setting is related with the communities in a user's neighborhood graph

Off-line extract features:

[1] Community Structure

[2] Other profile information

Using classifier to recommend the friend list based on users current settings

Using classifier to recommend the friends that the classifier is most uncertain about them.

Recommend the user to set these friends's privacy in the next step

Structure and Enhancement

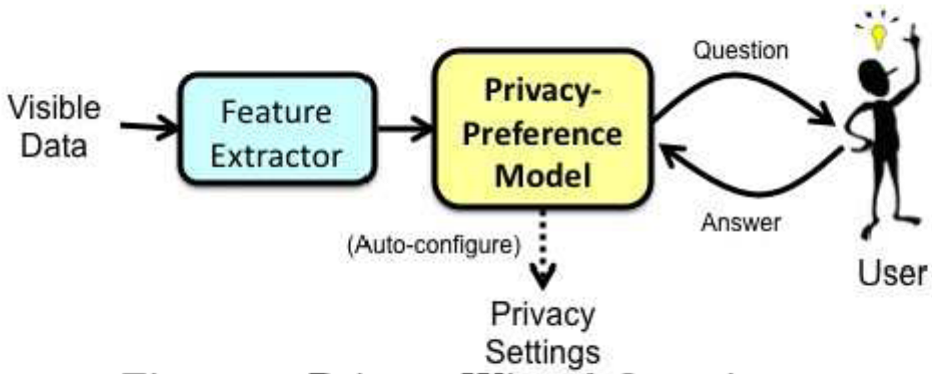
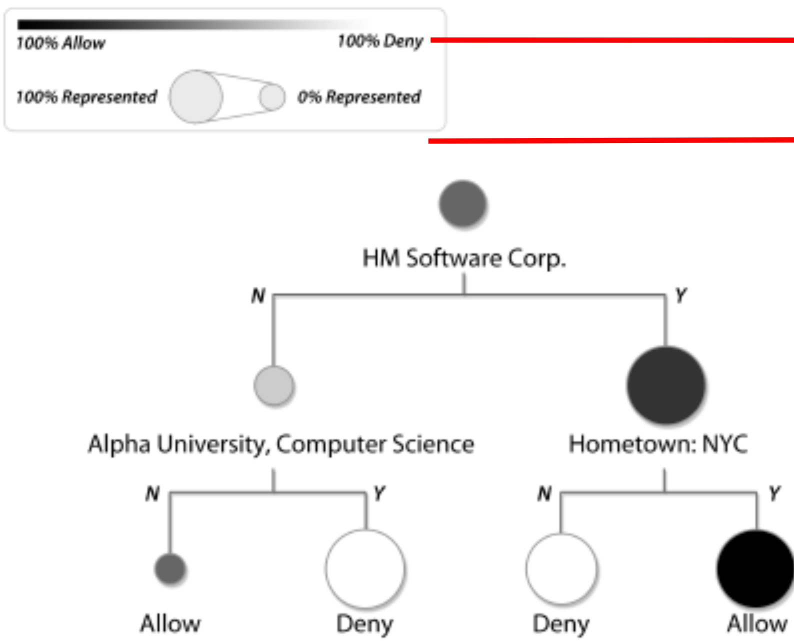


Figure 2: Privacy Wizard Overview

For experienced users, let himself to select the next setting friends

Display a decision tree to represent the classifier



Decision distribution

Number of labeled friends in each node

Figure 4: Visualization of Decision Tree Model

Summary

- You have certain control of the information you are sharing
- You often cannot estimate the long term risk vs. shot term gain
- Algorithms to measure potential privacy risks due to information shared either explicitly or implicitly
- Models to alleviate your burden on privacy management