LECTURE 17:SYBILGUARD: DEFENDING AGAINST SYBIL ATTACKS VIA SOCIAL NETWORKS

COMP4941: Social Information Network Analysis and Engineering

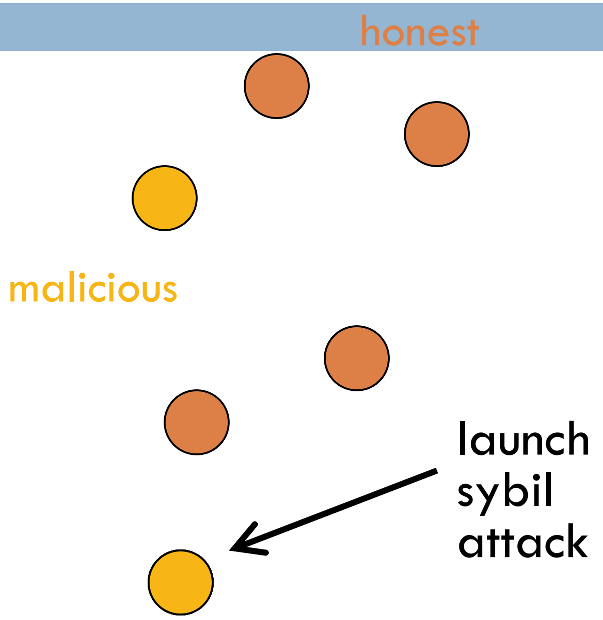
Friday April 24th 2015

Mid-term

- Highest 98, Lowest 31
- Mean 68, STD 14.9

Background: Sybil Attack

- Sybil attack: Single user pretends many fake/sybil identities
  - Creating multiple accounts from different IP addresses
- Sybil identities can become a large fraction of all identities
  - Out-vote honest users in collaborative tasks



Background: Defending Against Sybil Attack

- Using a trusted central authority
  - Tie identities to actual human beings
- Not always desirable
  - Can be hard to find such authority
  - Sensitive info may scare away users
  - Potential bottleneck and target of attack

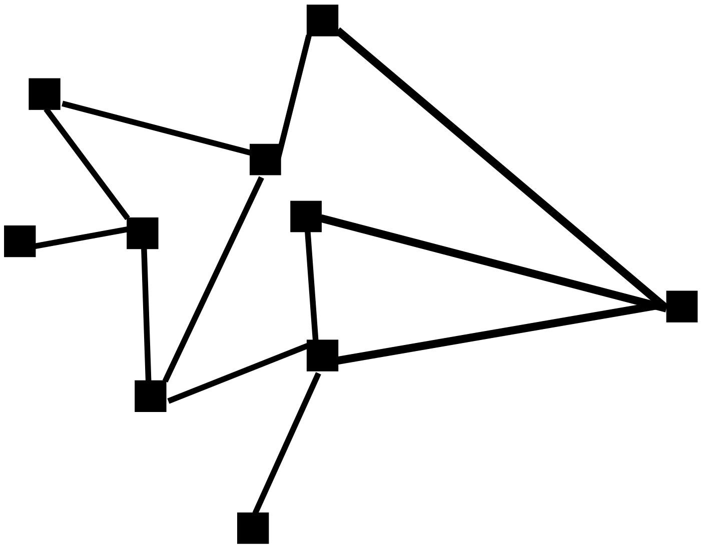
- Without a trusted central authority
  - Impossible unless using special assumptions [Douceur'02]
  - Resource challenges not sufficient -- adversary can have much more resources than typical user

SybilGuard’s Central Authority

- Main Idea: Use a social network as the “central authority”
- A node trusts its neighbors
- Each node learns about the network from its neighbors

SybilGuard Basic Insight: Leveraging Social Networks

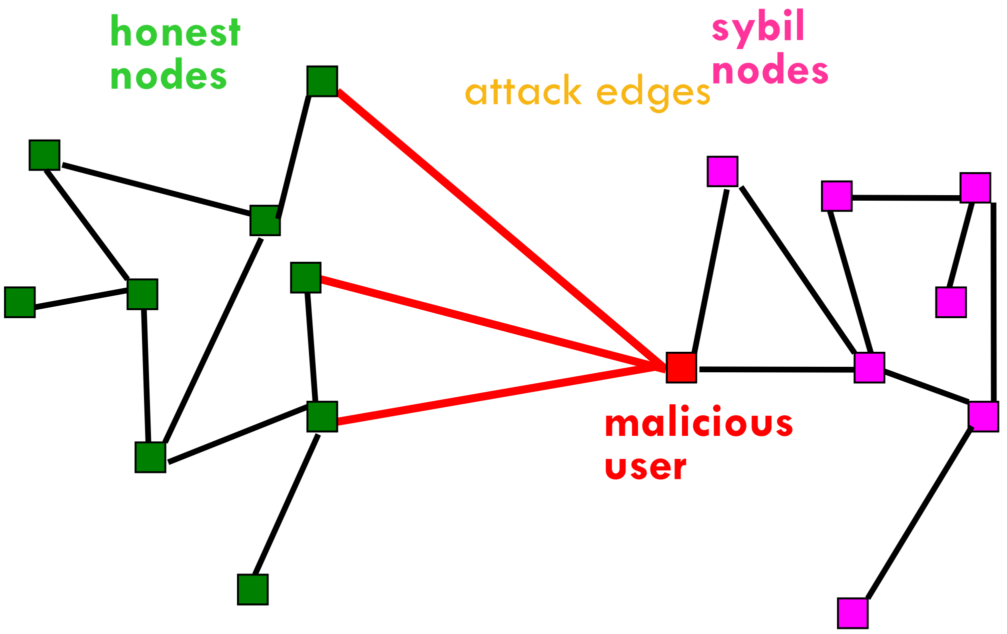
Our Social Network Definition



- Undirected graph
- Nodes = identities
- Edges = strong trust
  - E.g., colleagues, relatives

SybilGuard Basic Insight

- *n* honest users: One identity/node each
- Malicious users: Multiple identities each (sybil nodes)



•Edges to honest nodes are "human established"

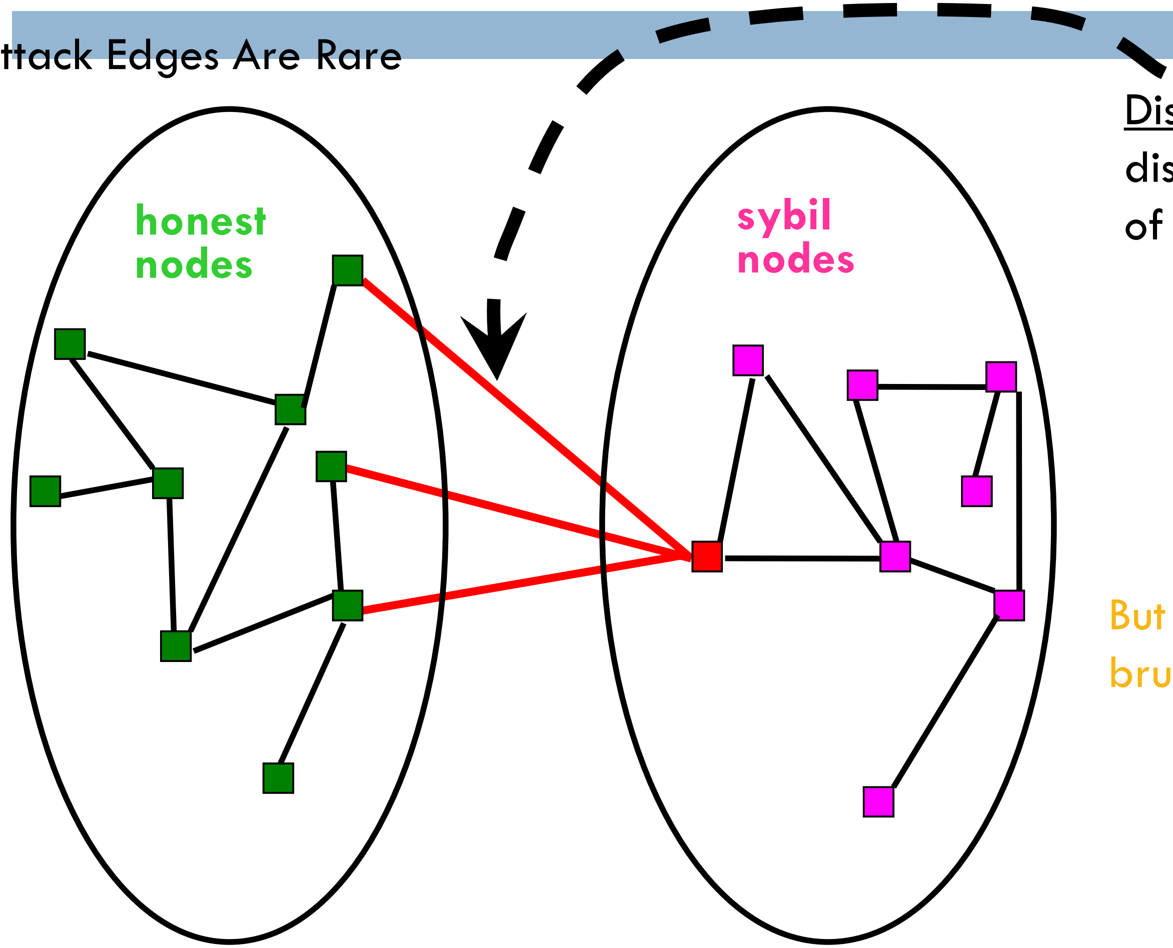
•Attack edges are difficult for Sybil nodes to create

•Sybil nodes may collude – the adversary

Observation: Adversary cannot create extra edges between honest nodes and sybil nodes

SybilGuard Basic Insight

Attack Edges Are Rare



Dis-proportionally small cut disconnecting a large number of identities

But cannot search for such cut brute-force…

SybilGuard’s Model

- A social network exists containing honest nodes and Sybil nodes
- Honest nodes provide a service to or receive a service from nodes that they “accept”

Goal of Sybil Defense

- Goal: Enable a *verifier* node to decide whether to accept another *suspect* node
  - Accept: Provide service to / receive service from
  - Idealized guarantee: An honest node accepts and only accepts other honest nodes

- SybilGuard:
  - Bounds the number of sybil nodes accepted
  - Guarantees are with high probability
  - Accepts and is accepted by most honest nodes
  - Approach: Acceptance based on random route intersection between verifier and suspect

Random Routes

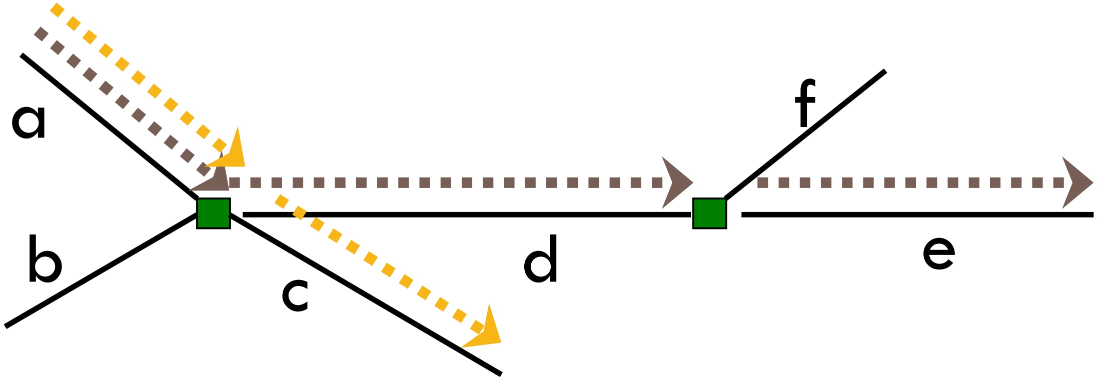
- Every node picks a random routing from input to output edges
- A directed edge is in exactly one route of unbounded length
- A directed edge is in at most w routes of length w

Clever Use of Random Routes

□ Each node finds all the length w random routes that start at the node itself

□ Honest node V accepts node S if most of V's random routes intersect a random route of S

Random Walk Review

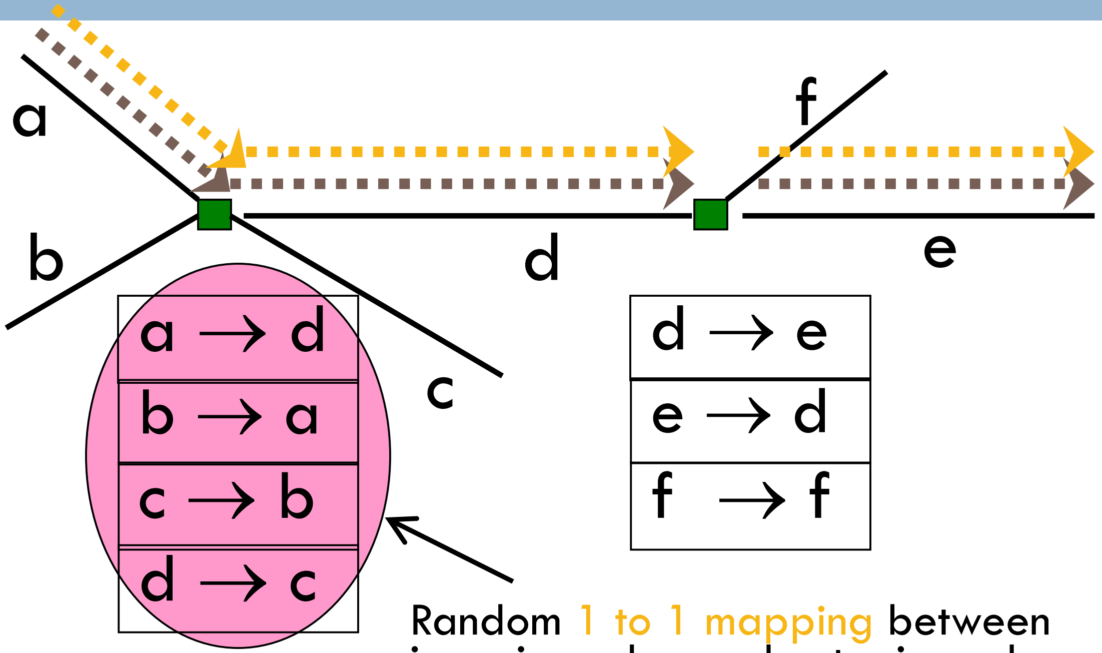


pick random edge d

pick random edge e

pick random edge c

Random Route: Convergence



randomized routing table

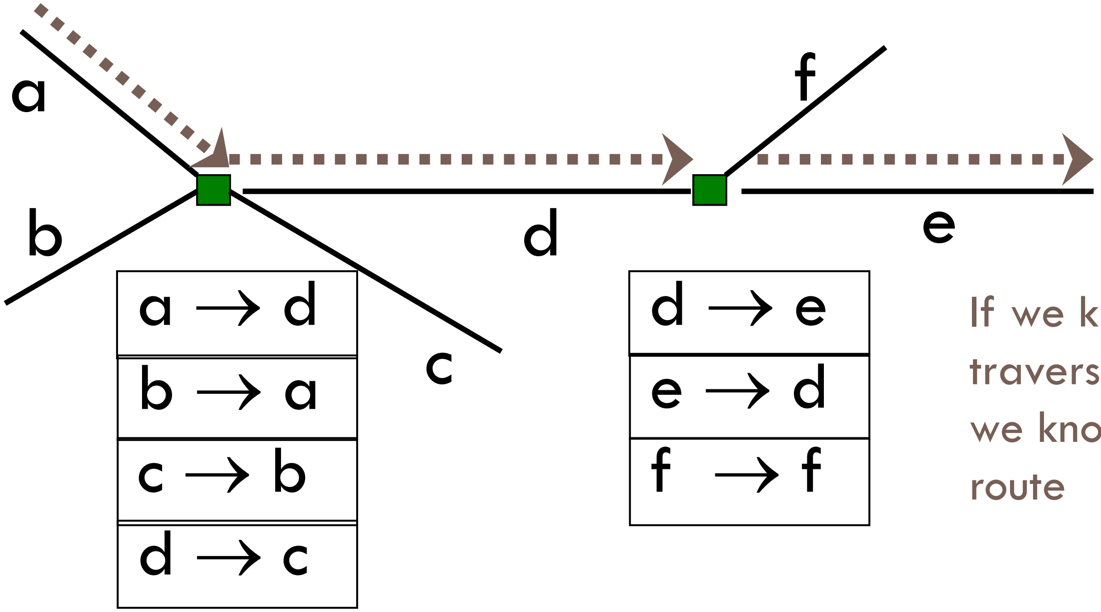
a → d
b → a
c → b
d → c

d → e
e → d
f → f

Random 1 to 1 mapping between incoming edge and outgoing edge

Using routing table gives Convergence Property:
Routes merge if crossing the same edge

Random Route: Back-traceable



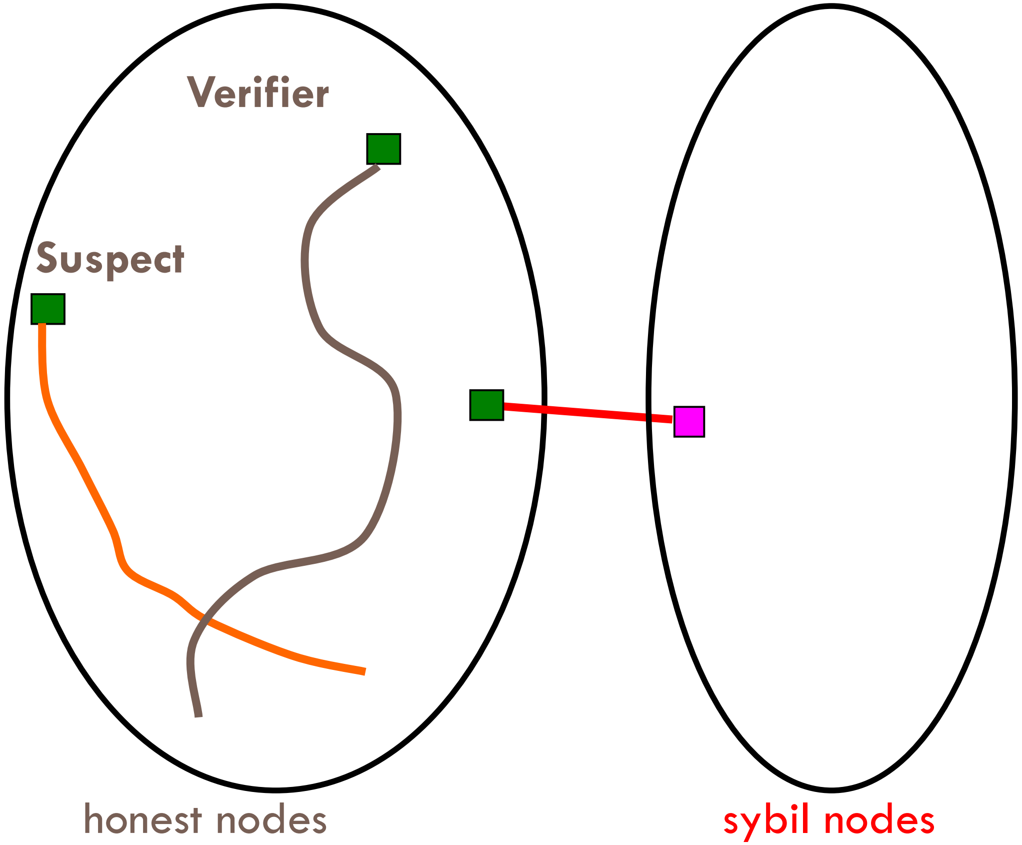
a → d
b → a
c → b
d → c

d → e
e → d
f → f

If we know the route traverses edge e, then we know the whole route

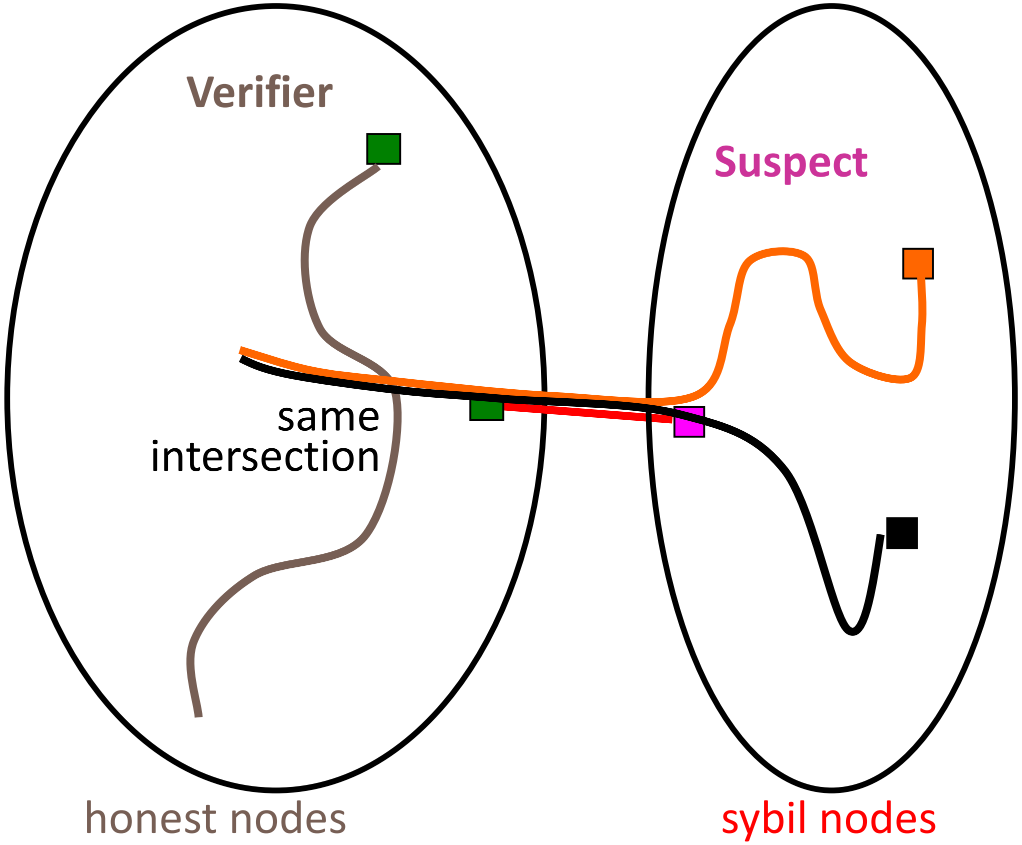
Using 1-1 mapping gives Back-traceable Property: Routes may be back-traced

Random Route Intersection: Honest Nodes



- Verifier accepts a suspect if the two routes intersect
  - Route length *w*: $\sim \sqrt{n}\log n$
  - W.h.p., verifier's route stays within honest region
  - W.h.p., routes from two honest nodes intersect

Random Route Intersection: Sybil Nodes



- Each attack edge gives one intersection
- Intersection points are SybilGuard's equivalence sets

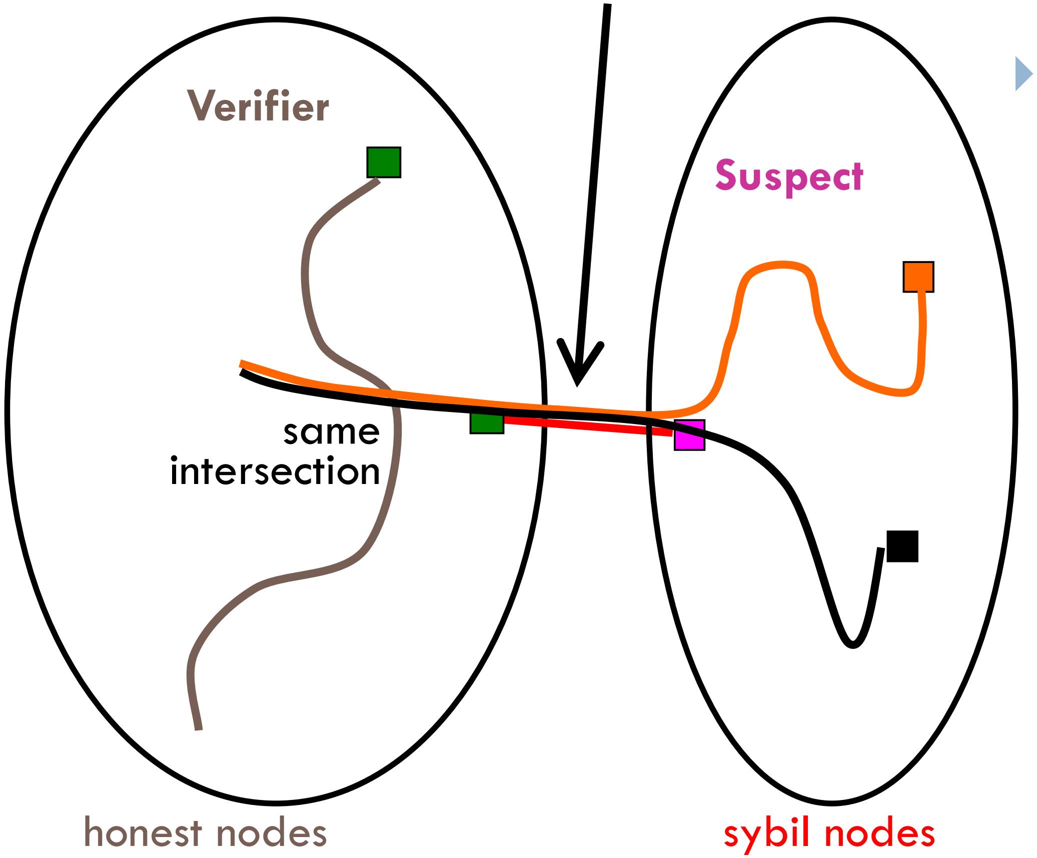
Random Route Intersection: Sybil Nodes

- SybilGuard bounds the number of accepted sybil nodes within $g*w$
  - g: Number of attack edges
  - w: Length of random routes

- Next …
  - Convergence property to bound the number of *intersections* within g
  - Back-traceable property to bound the number of accepted sybil nodes per intersection within w

Bound # Intersections Within *g*

must cross attack edge to intersect even if sybil nodes do not follow the protocol

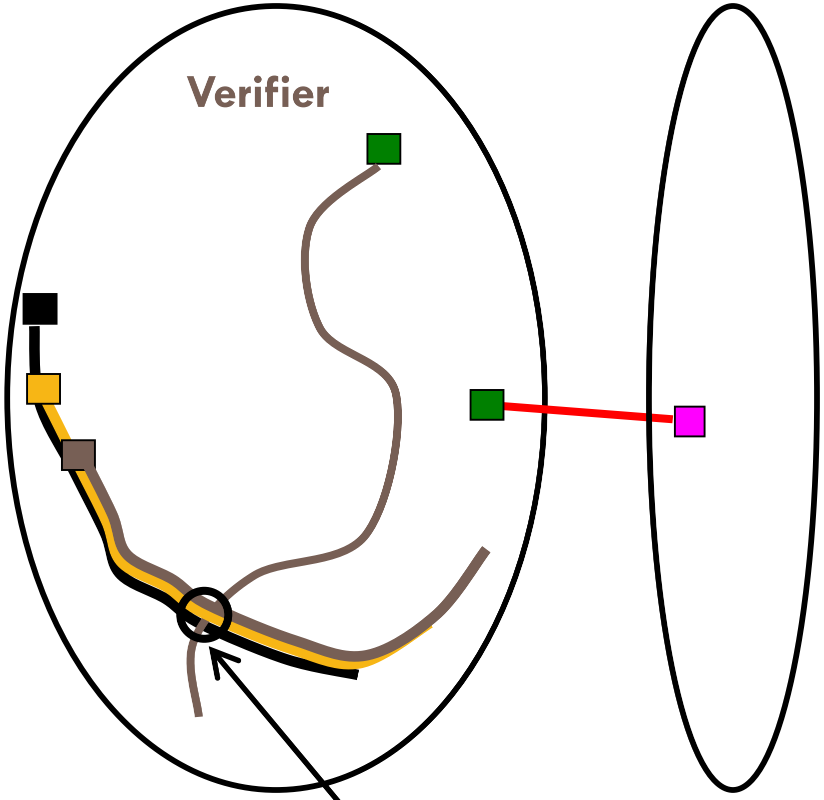


- Convergence: Each attack edge gives one intersection

  ⇒ at most *g* intersections with *g* attack edges

Intersection = (node, incoming edge

Bound # Sybil Nodes Accepted per Intersection within *w*



for a given intersection

- Back-traceable: Each intersection should correspond to routes from at most *w* honest nodes
- Verifier accepts at most *w* nodes per intersection
  - Will not hurt honest nodes

Bounds on Accepted Sybil Nodes

- For routes of length w in a network with g attack edges, WHP,
  - Accepted nodes can be partitioned into sets of which at most g contain Sybil nodes
  - Honest nodes accept at most w*g Sybil nodes

Summary of SybilGuard Guarantees

- Power of the adversary:
  - *Unlimited* number of *colluding* sybil nodes
  - Sybil nodes may not follow SybilGuard protocol

- W.h.p., honest node accepts ≤ $g*w$ sybil nodes
  - g: # of attack edges
  - w: Length of random route

If SybilGuard bounds # accepted sybil nodes within	Then apps can do
$n/2$	byzantine consensus
n	majority voting
not much larger than n	effective replication

SybilGuard Protocol

- Security:
  - Protocol ensures that nodes cannot lie about their random routes in the honest region
- Decentralized:
  - No one has global view
  - Nodes only communicate with direct neighbors in the social network when doing random routes

SybilGuard Protocol (continued)

- Efficiency: Random routes are performed only once and then "remembered"
  - No more message exchanges needed unless the social network changes
  - Verifier incurs O(1) messages to verify a suspect
- User and node dynamics:
  - Different from DHTs, node churn is a non-problem in SybilGuard …

Restrictions Imposed On Applications

- There must be a social network
  - Nodes must create and maintain their friendships
- How many social networks will we need?
  - One for each application, or
  - A single network used by many applications

Evaluation Results

Simulation based on synthetic social network model [Kleinberg'00] for 10^6, 10^4, 10^2 nodes

- With 2500 attack edges (i.e., adversary has acquired 2500 social trust relationships):
  - Honest node accepts honest node with 99.8% prob
  - 99.8% honest node properly bounds the number of accepted sybil nodes

Privacy Implications

- Information about friends spreads along routes
- Verification involves nodes sharing all their routes
  - Bloom filters help here
- Nodes are not anonymous

PRIVACY IN SOCIAL MEDIA

Content Sharing Privacy

- Before you post, ask the following:
  - Will this post/picture cause a problem for me?
  - Can I say this in front of my mother?
- Divide your Friends into groups, lists, or circles
- Limit the number of people that see it
- Share public information with the public
- Share inner thoughts and personal feelings with close friends

Networking Privacy

- Do not Friend or Connect with people that you have not met in person or know well
- Reject Friend requests and Connections from strangers
- Having a lot of Friends can work against you
    - Facebook may ask you to identify your Friends
- Limit your visibility on services

Location Privacy and Safety

- Limit your check-in information to friends only
- Never check in at your home, school, work
- A mayorship is a public "office"
- Avoid public lists for a location
- Do not let friends check you in
- Review posts you are tagged in

Service Specific Configuration Options

Google Security and Privacy

- Enable 2-step verification
  - Use Google Authenticator or text-based codes
  - Applies to (almost) all Google services
- Create Google+ circles based on sharing needs
- Turn off geo location data in photos
- Turn off “find my face” in photos and videos
- Manage your Dashboard data

Facebook Security Tools

- Enable
  - Secure Browsing
  - Login Notifications (text and email)
  - Login Approvals (text and mobile Code Generator)
- Select your ~~Trusted Friends~~
- Review and Monitor
  - Recognized Devices
  - Active Sessions
- Delete old and unused Apps

Facebook Privacy Tools

- Limit App access to your data
- Set your default audience to Friends
- Customize your timeline content settings
  - Who can post, tag you, tag reviews
  - Disable tag suggestions for photos uploaded
- Limit search engine inclusion
- Limit third-party and social ads
- Limit info that can be included by others in apps

Dropbox Security and Privacy

- Enable two-step verification
- Disable LAN sync on laptops
- Do not put sensitive data into Dropbox
- Encrypt files if needed
- Unlink old devices
- Review Apps linked to your account
- Turn on email for new devices and apps added
- Review your shared folders periodically

Twitter Security and Privacy

- Enable Protect My Tweets
- Enable HTTPS
- Require personal information for password reset
- Disable location data for tweets
  - Delete old location data too

Linkedin Privacy

- Turn off data sharing with third-party apps and sites
- Consider changing your photo visibility, activity broadcasts
- Remove Twitter access
- Disable ads from third-party sites
- Enable full-time SSL connections

Foursquare Privacy

- Do not include yourself in lists of people checked into a location
- Do not earn mayorships
- Do not let friends check you into places
- Do not let venue managers see you

Stay Safe

- Stay up to date on software and settings
- Be selective when choosing friends
- Using your thinkin' before you're tweetin'!
- Be mysterious