SybilGuard: Defending Against Sybil Attacks via Social Networks

Christian Kretschmer

Albert-Ludwig Universität
Lehrstuhl Rechnernetze und Telematik
Prof. Schindelhauer

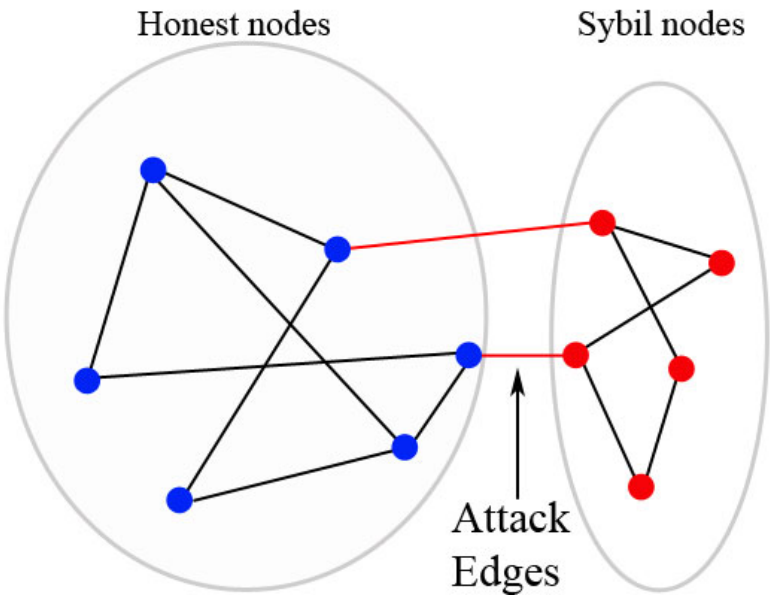
2. März 2007

Outline

- Social Network and Random Routes
- Limiting number and size of Sybil groups
- Verification process
- SybilGuard under dynamics
- Evaluation

CK

Social network



Social network:

- *Honest nodes* (called friends when connected)
- *edge key*: unique symmetric secret key between friends to authenticate messages (out-of-band distribution)
- Nodes inform friends about IP-Changes (IP is only a hint)

CK

The Sybil attacker

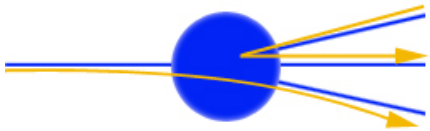
The adversary might try to increase the number of *attack edges* (g) in the following ways:

- Convince more honest node to establish a social trust in „real life “ (difficult)
- Creates more sybil nodes and try to convince the honest node to trust these nodes, too. But because of the convergence property and the single *edge key*, the number of attack edges remain unchanged.
- It’s possible to resurrect a dropped edge by using the old *edge key*. (prevented by requiring out-of-band confirmation when deleting edges)
- Compromising a large fraction: *SybilGuard* doesn’t help here!

CK

Random routes

Because only edges between friends are of little practical use *SybilGuard* bootstraps a protocol that enables honest nodes to accept other honest nodes. The basis of *SybilGuard* are the *random routes*.



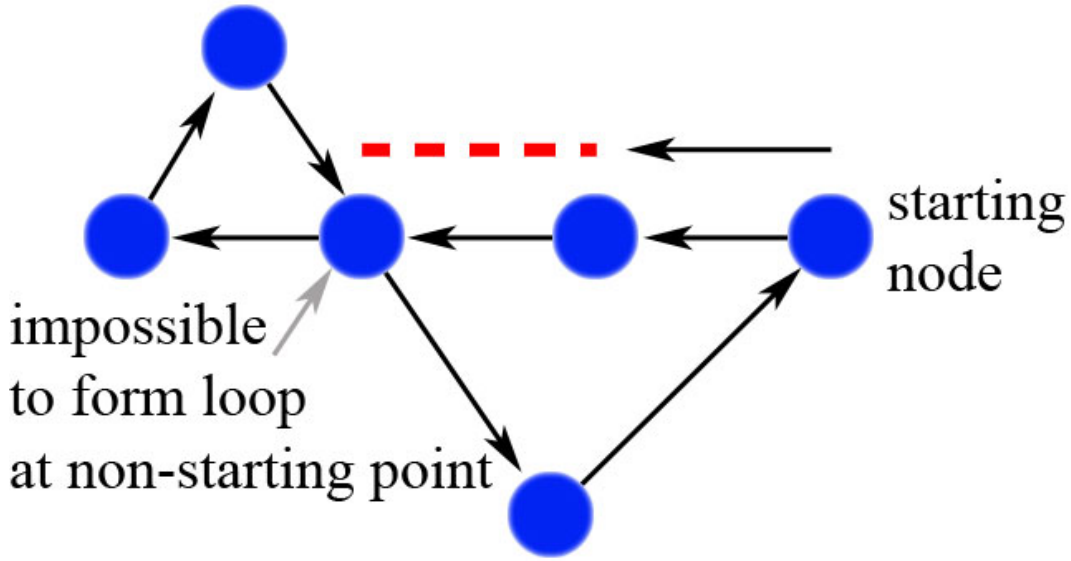
Properties of *random routes* are:

- *convergence property*: Two random routes entering an honest node along the same edge will always exit along the same edge.
- *back-traceable property*: The outgoing edge uniquely determines the incoming edge.

CK

Loops in random routes

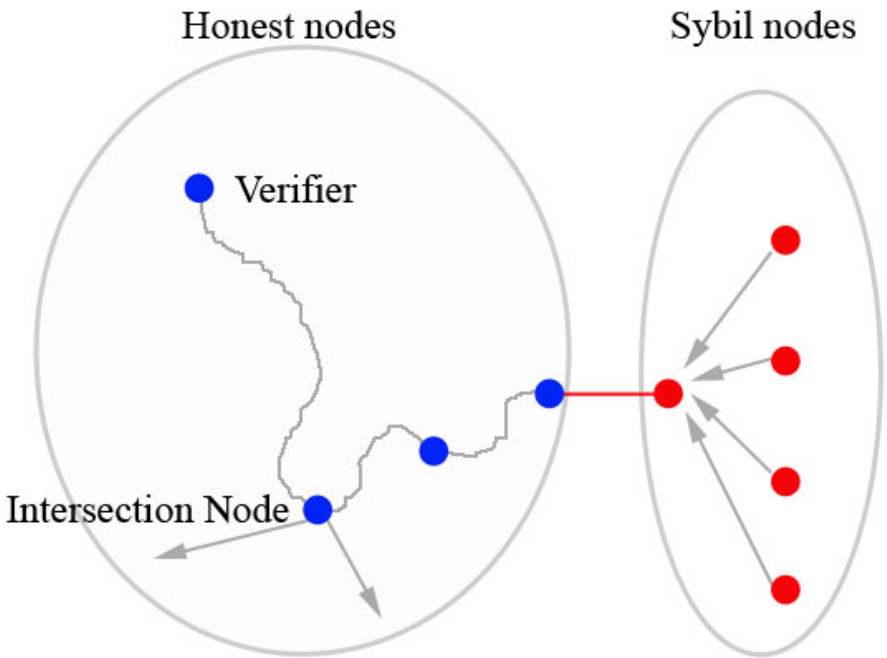
Loops are reducing the effective length of the route and the probability of an intersection but do not compromise security.



- Loops can only form at the starting node
- Smallest loop has three hops
- Loop-Probability gets smaller and smaller at each hop ($\frac{1}{d}, \frac{1}{d^2} \ldots$)
- Use random routes at all edges of an node to increase the intersection probability

CK

Bounding the number of Sybil-Groups



To intersect with the verifier's random route, a sybil node's random route must traverse the attack edges.
Because of the *convergence property*, the random routes from sybil nodes must merge.

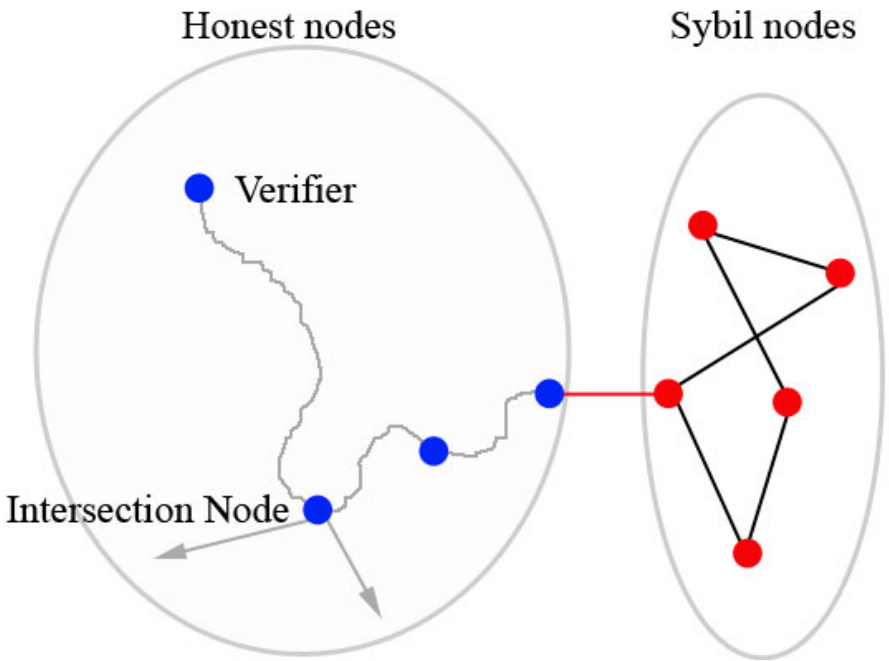
CK

Bounding the size of Sybil-Groups

Each node uses a randomized *routing table* to choose the next hop. Once established it will never be changed until a degree change happens . If two routes share an edge in the same direction, then one must start in the middle of the other because of the properties of random routes.

CK

Bounding the size of Sybil-Groups



- SybilGuard bounds the size of Sybil-Groups within the length of the random routes w.
- From the back-traceable property, there can be at most w distinct routes.
- Thus, the verifier accepts exactly one node for each of the w hops

CK

Why is bounding important?

- If g is bounded $\Rightarrow$ store your replicas on $g + 1$ different equivalence groups
- If $g * w < n \Rightarrow$ the probability of having a majority of replicas on honest nodes approaches 1.0 exponentially fast with the number of replicas.

CK

Registry and witness table

Each node needs to maintain only two local structures:

- *Registry table*: The nodes which random routes traverse me
- *Witness table*: The nodes which are on my random routes

The only action each node needs to perform is to propagate these structures to direct neighbors when performing a random route.

CK

Registration

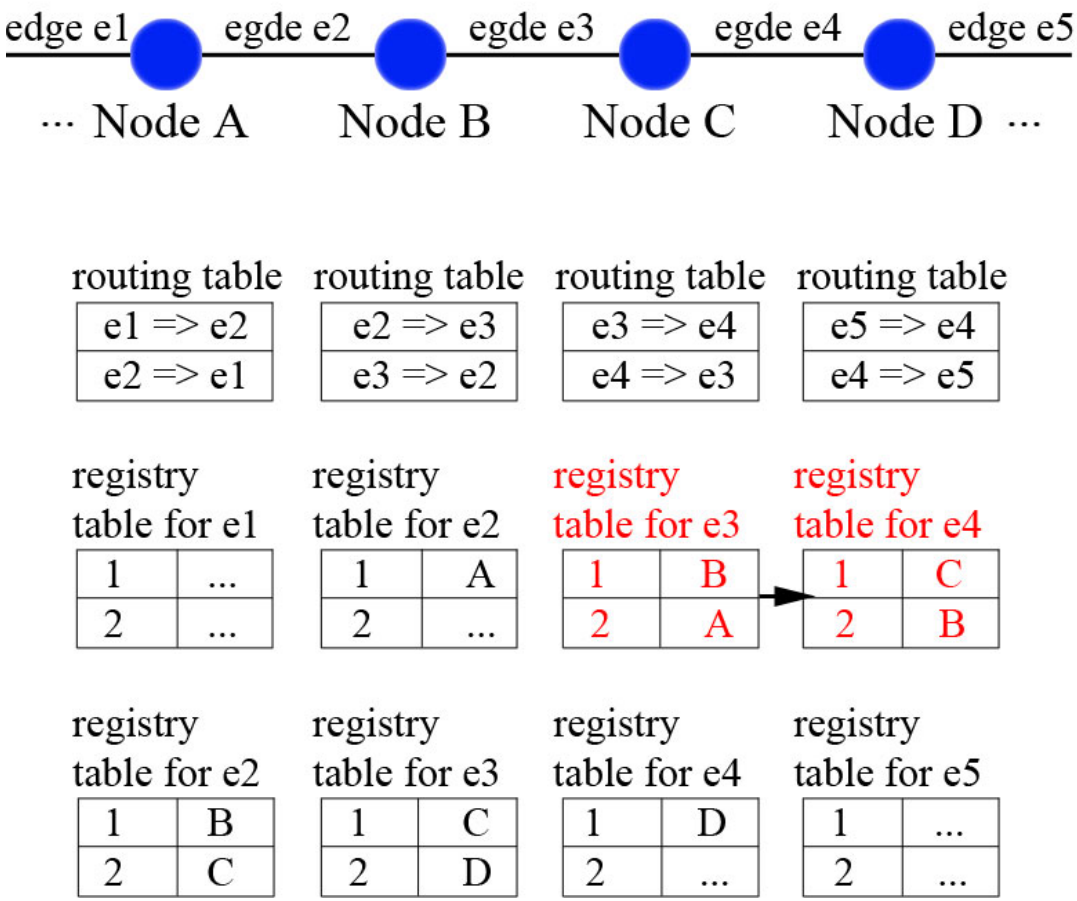
First of all the node has to register with all w nodes along each random route. This prevent a node from lying about its route.

In this process use a *token* that cannot easily forged.

- Initial design: IP address as *token*
- Current design: public key cryptography (privat / public key)
- IMPORTANT: privat / public key $\neq$ *edge key*

CK

Registry table



In order to simplify: w=2 and d=2 for all nodes, A,B,C.. stand for nodes' public keys.

CK

Bandwith needed for Registry table exchange

The overhead of the protocol with w=2000 and 1024 bit public keys:

- Each registry table: 256 KB
- 10 neighbors = 2.56 MB traffic
- Optimization: 160 bit hashes instead of public keys (400 KB traffic)

CK

Witness table

The witness table is propagated and updated in a similar fashion like the registry table. It contains:

- Public key (or its hash)
- IP address (only as a hint)

Difference to registry table:

- IP address changes should be updated
- can be done lazily

CK

Verification in detail

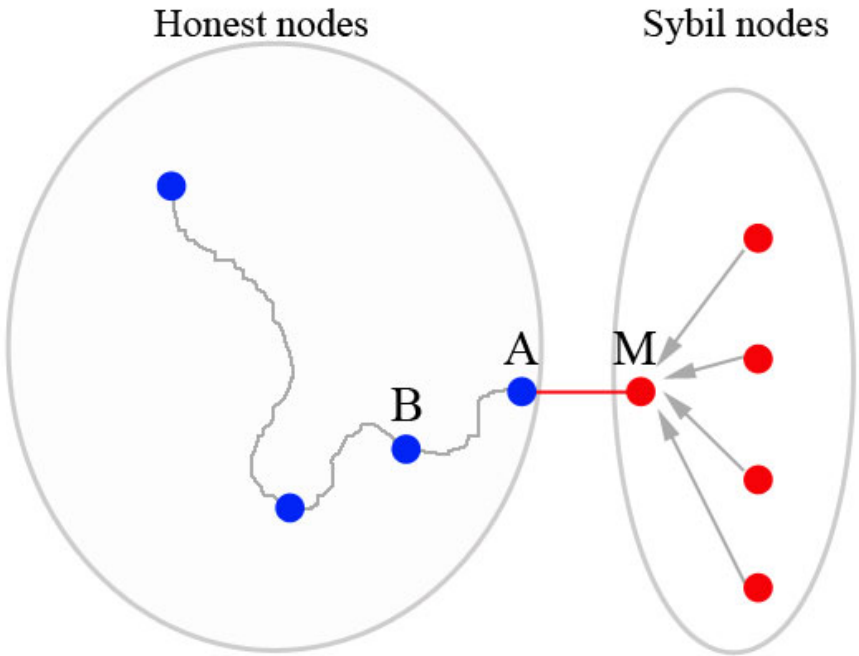
1. *Subject* S sends *witness tables* and his *public key* to the *verifier V*
2. Now V searches intersections with his *witness tables*
3. Determines *public key* of intersection node X (if any)
4. V contacts X with his recorded *IP*
5. If the *IP* is wrong then ask X's neighbors
6. V *authenticates* X by requiring X to sign each message
7. V with X checks S's *public key* is indeed present in X's *registry table* (Entry number is not relevant)
   - If present, this route accepts S
   - If at least half of V's routes accept S, V accepts S
8. From now on V always *authenticates* S by requiring S to sign every message using his *private key*

CK

Verification and sybil nodes

What if some nodes (sybil nodes) don't follow the protocol?

- For simplicity: all honest nodes degree is d and one attack edge
- Altogether there are n * d * w registry table entries



M can pollute $w + (w-1) + ... + 1 \approx \frac{w^2}{2}$ entries system-wide.

Even if there are g attack edges ($g * \frac{w^2}{2}$) is still less than half of the total number of entries.

CK

Designing the length of random routes

The value of w must be sufficiently small to ensure:

- Verifiers random route remains entirely in the honest region
- Size of sybil groups is not excessively large

But w has to be large enough to ensure that random routes intersect with a high probability!

$\Theta(\sqrt{n}logn)$ satisfies the above requirements.

CK

Designing the length of random routes

The only problem with $\Theta(\sqrt{n}logn)$ is that we don't have knowledge about n. Solution:

1. A performs a short random walk (e.g. 10 hops) ending at B (3 hops is a good estimation on w (see Evaluation))
2. Next A and B both perform random routes (send witness table) to determine how long the routes need to be
3. Then A obtains multiple samples and calculates the median *m*
4. Setting $w = 2.1 * m$ ensures a intersection-probability of 95% regardless of n

CK

Dealing with offline nodes

Communication needed only for:

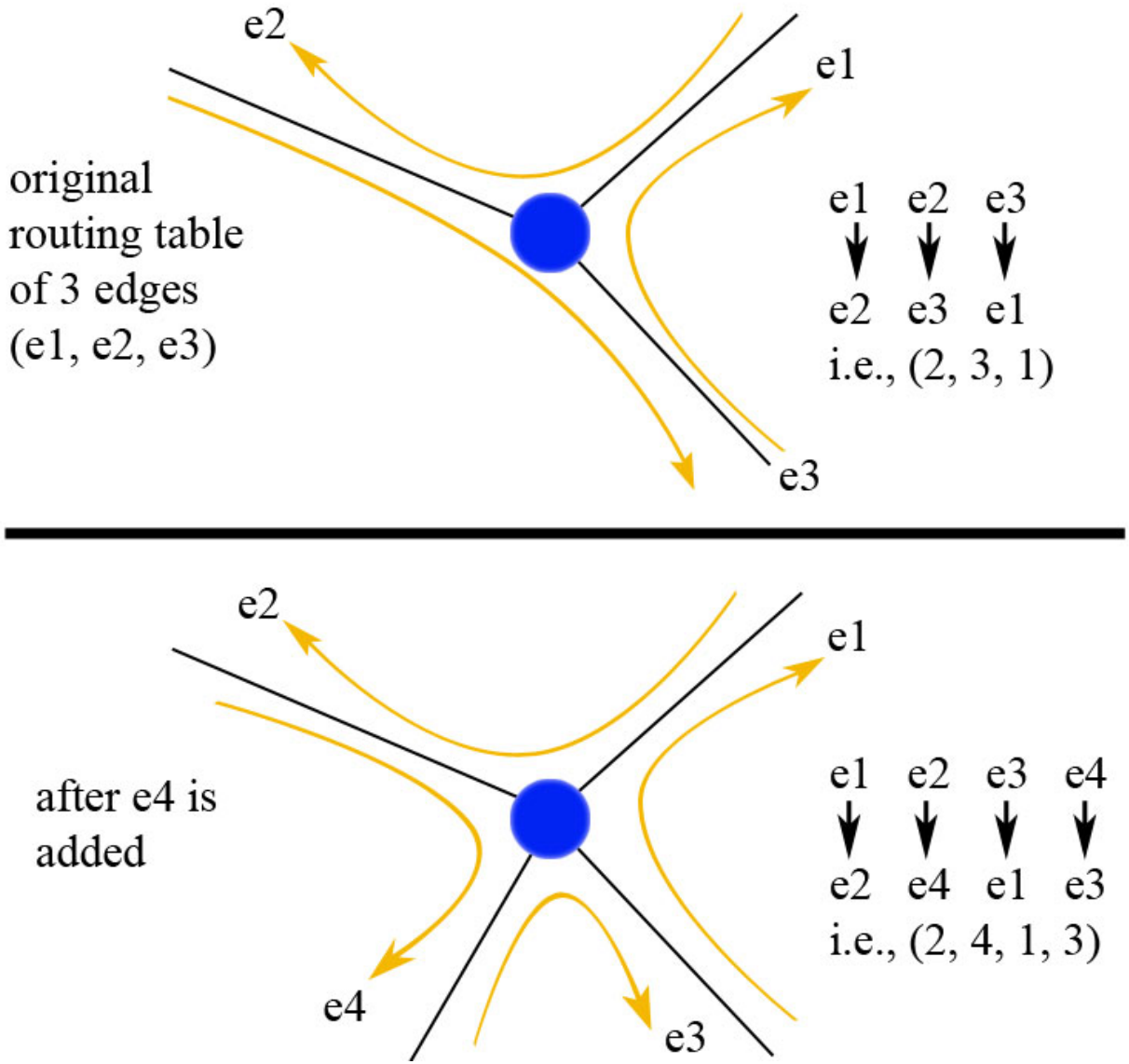
1. Verification
2. Propagation of registry and witness table

Because of multiple random routes and multiple intersections the Verification process works as long as the majority of the routes have at least one intersection node online.

Propagation is only done when a new user or edge is created or deleted. But in a social network people make or lose trust relations over months.
As Optimization for the Propagation: Lookahead routing table

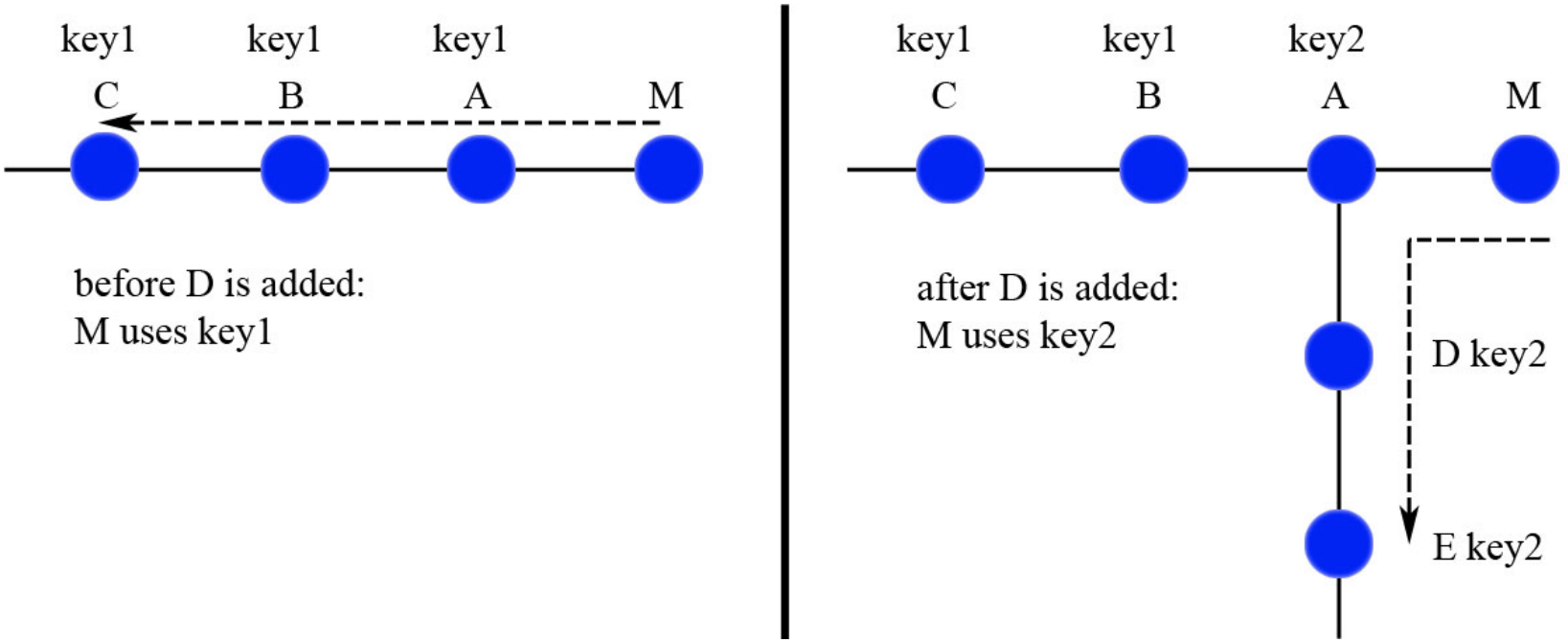
CK

Incremental maintenance of routing tables



CK

Attacks exploiting node dynamics



With random routes along all directions SybilGuard is secure against such attacks, because key1 will be overwritten by e.g. node D (back-traceable property).

CK

Evaluation

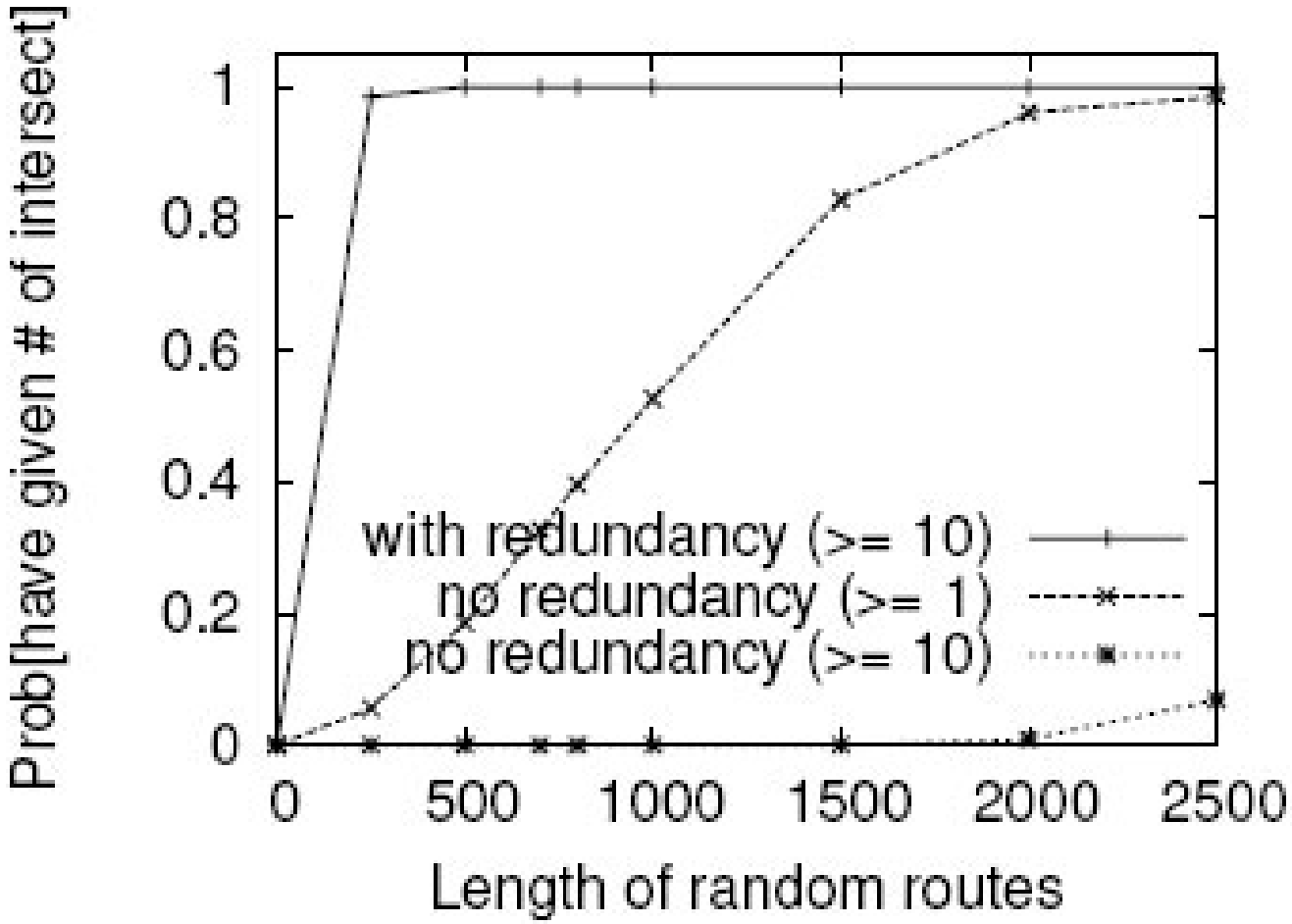
- Kleinberg's synthetic social network model
- Three different graph sizes:
  1. Million-node graph, degree = 24
  2. 10.000-node, degree = 24
  3. 100-node graph, degree = 12

Results with no malicious users:

- Loops are quite rare (0,3% in the 10.000-node graph - 10% in the 100-node graph)
- Honest nodes being successfully accepted (over 99%; see next slide)
- Length of the routes (also see next slides)

CK

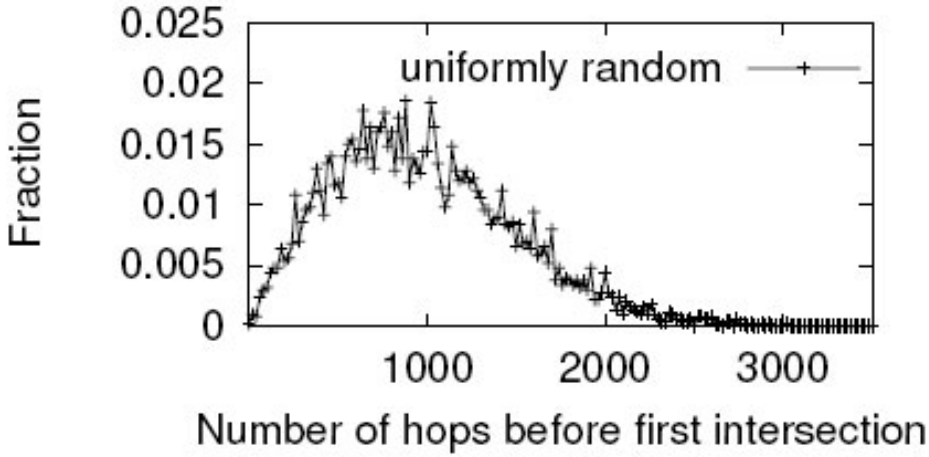
Probability of intersection

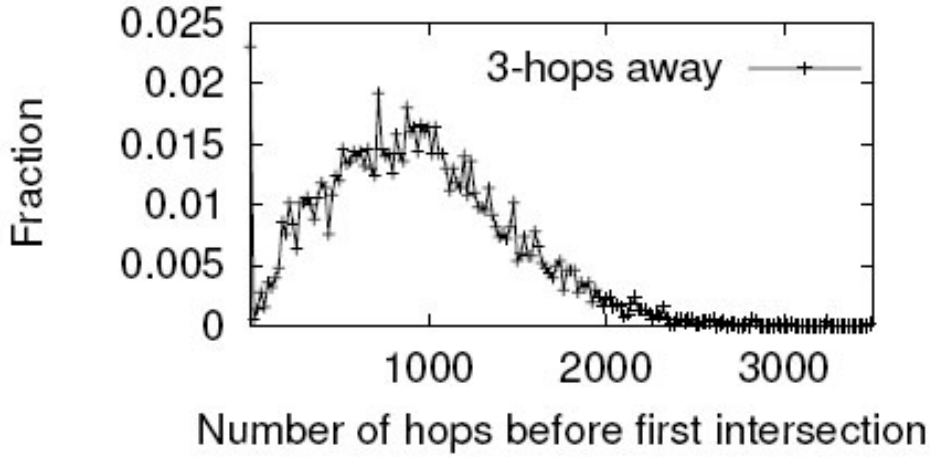


Source: SybilGuard: Defending Against Sybil Attacks via Social Networks

CK

Probability distribution histogram



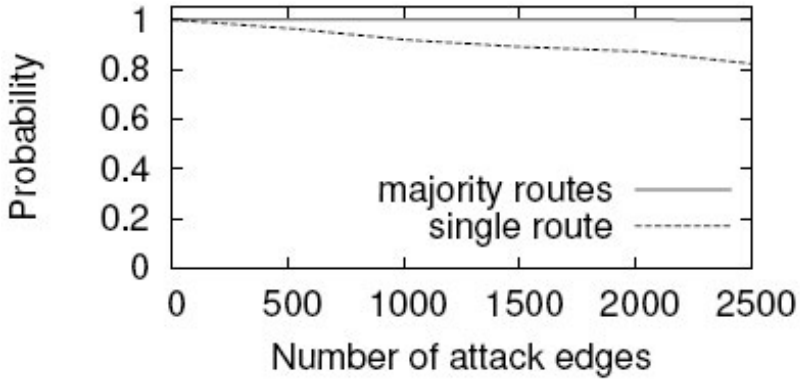


- w = 1906 $\pm$ 300 after 30 samples
- w = 197 $\pm$ 30 after 35 samples
- w = 24 $\pm$ 7 after 40 samples

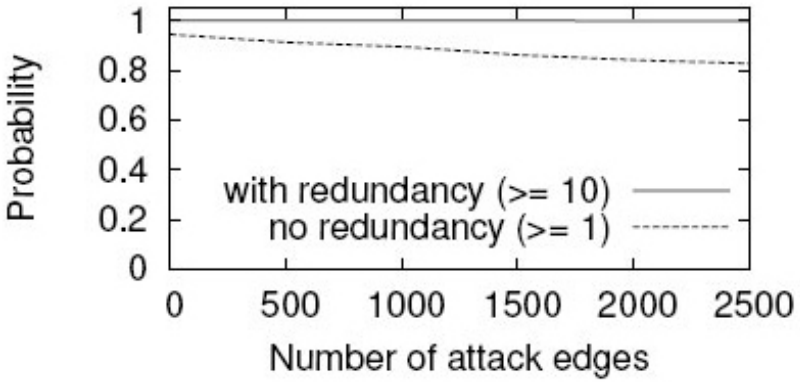
Source: SybilGuard: Defending Against Sybil Attacks via Social Networks

CK

Probability of Accepting honest nodes



Probability for the majority of an honest node's random route remain entirely in the honest region.



Probability of an honest node accepting another honest node.

Source: SybilGuard: Defending Against Sybil Attacks via Social Networks

CK

Thx for your attention!