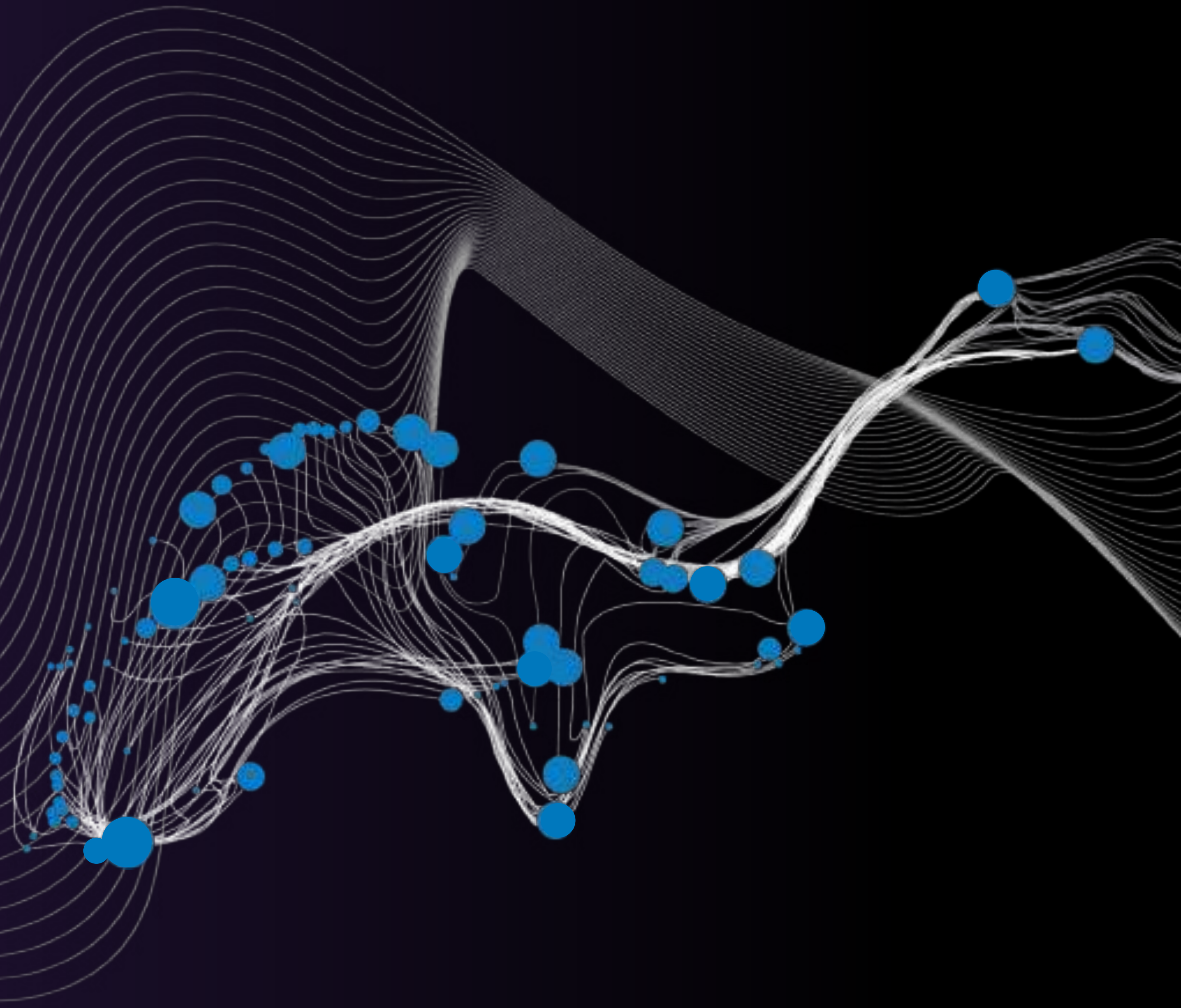
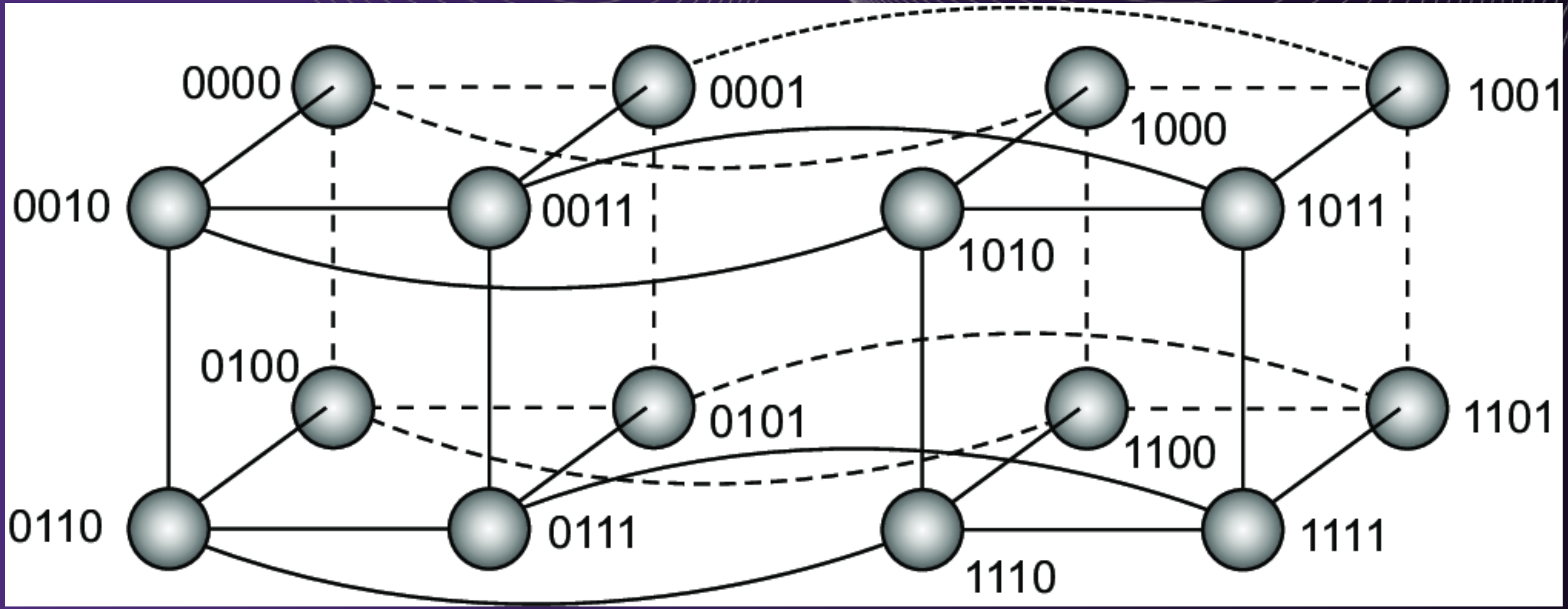
SECURITY IN PEER-TO-PEER NETWORKS



UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

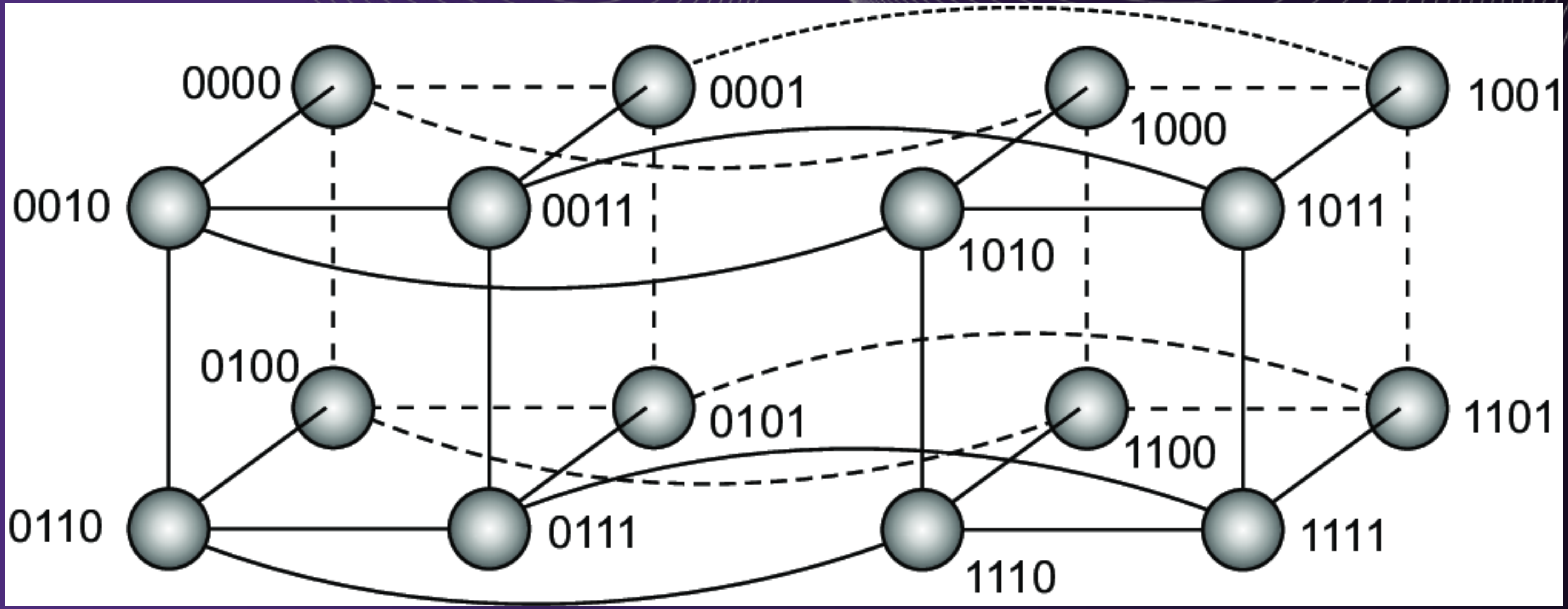
ESSENCE

- Each data item is associated with a unique key, e.g. *key(data item) = hash(data item value)*
- The P2P system stores *(key,value)* pairs.
- Lookups follow a predefined routing path from node where lookup is initated to node responsible for requested *key*.

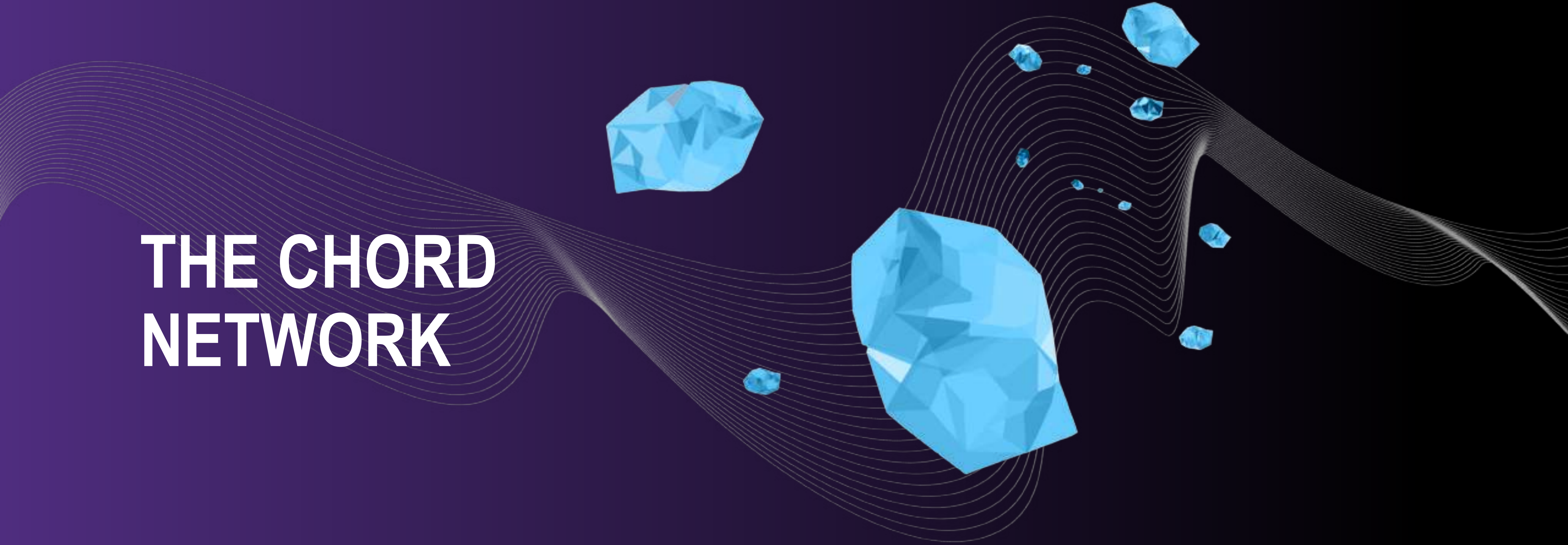


NOTE

- A peer-to-peer network is constructed as an overlay:
  - A node is formed by a (software) process
  - A link is formed, e.g. by a TCP connection through which one process sends messages to another (known) process
  - Links may change over time: it's really who knows who.



THE CHORD NETWORK



BASIC ORGANIZATION

- Chord nodes store files. Each file receives a unique *m*-bit key: *key(file) = hash(file contents)*.
- Nodes are organized in a logical ring, where each node gets a unique *m*-bit identifier.
- File *f* with key *k* is stored by node *p* with smallest $id(p) \geq k$ called successor $succ(k)$.
- Notation: node *p* is assumed to have id *p*.

BASIC ORGANIZATION

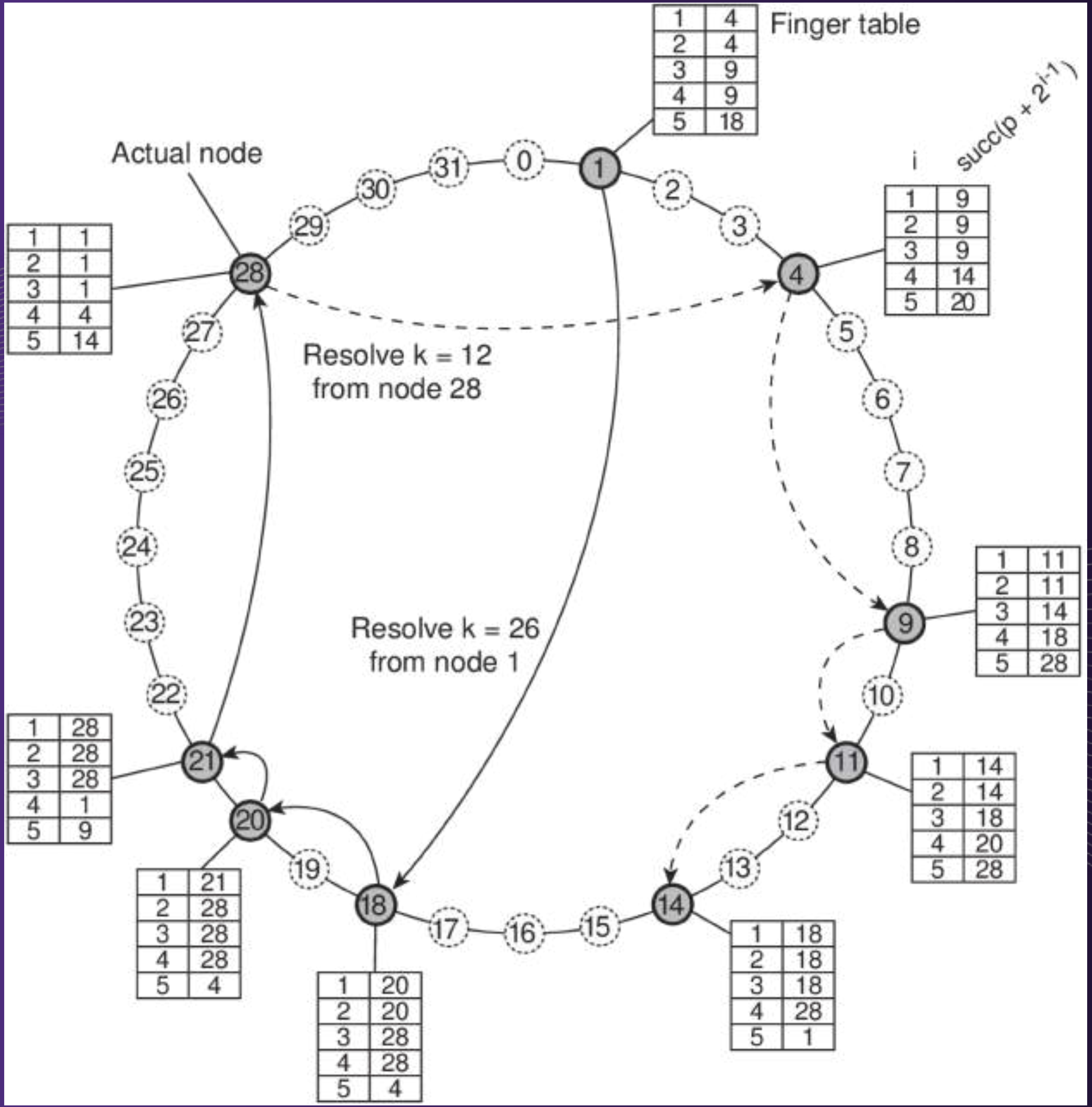
- Each node *p* maintains a finger table $FT_p[\,]$ with at most *m* entries:

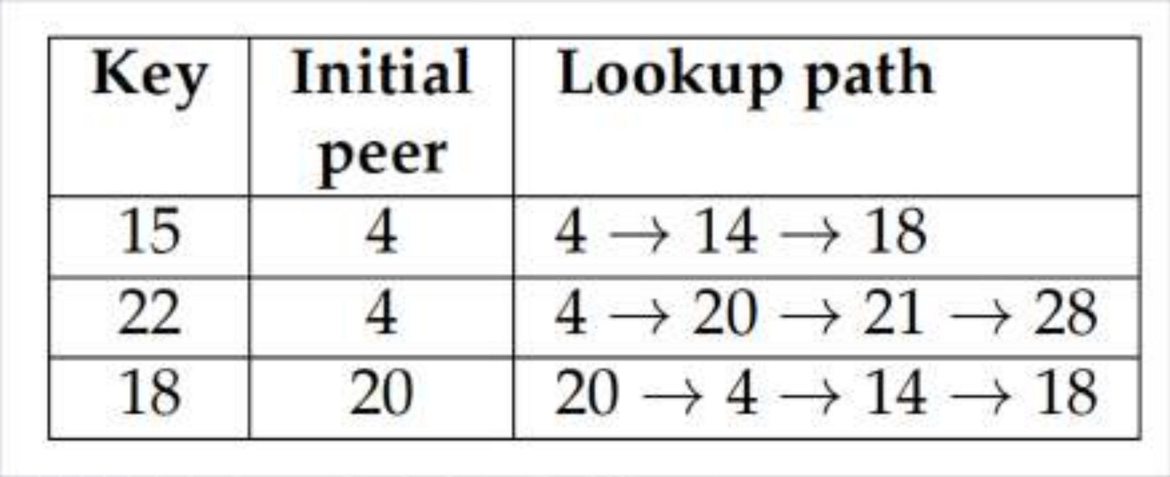
$$FT_p[i] = succ(p + 2^{i-1})$$

- Note: the *i*-th entry points to the first node succeeding *p* by at least 2^{i-1}.
- To look up a key *k*, node *p* forwards the request to node with index *j* satisfying

$$q = FT_p[j] \leq k < FT_p[j+1]$$

- If $p < k < FT_p[1]$, the request is also forwarded to $FT_p[1]$.





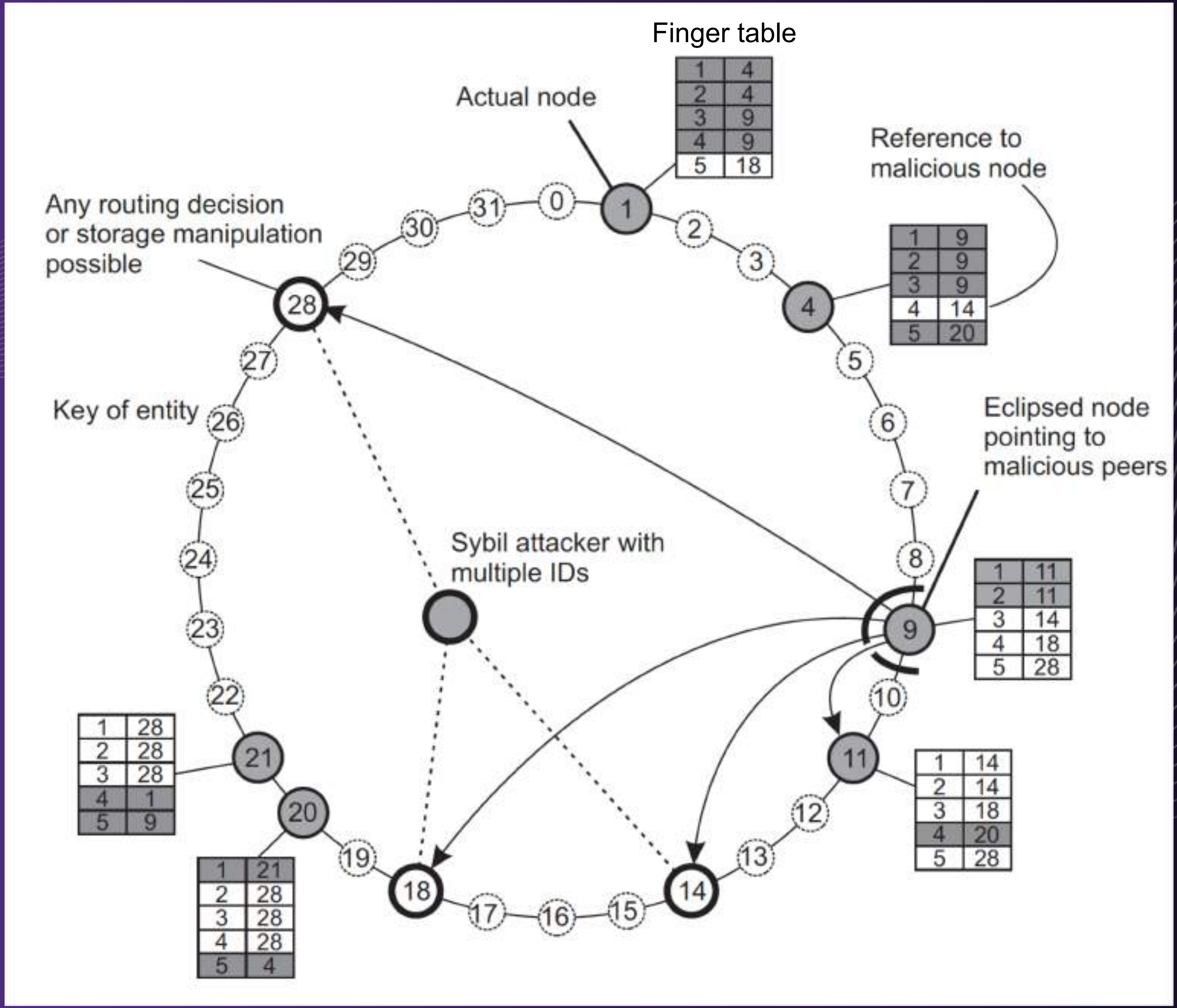
Key	Initial peer	Lookup path
15	4	4 → 14 → 18
22	4	4 → 20 → 21 → 28
18	20	20 → 4 → 14 → 18

SYBIL ATTACK

- A malicious entity simply launches a number of Chord nodes (e.g., as a bunch of processes distributed across several machines akin to botnets).
- Result: the collection of malicious nodes can easily collude in storing or modifying the files they are responsible for. Other effects are also possible.

ECLIPSE ATTACK

- Deliberately try to isolate benign nodes such they point mainly to malicious nodes.


Finger table
Actual node
Reference to malicious node
Any routing decision or storage manipulation possible
Key of entity
Eclipsed node pointing to malicious peers
Sybil attacker with multiple IDs
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

1	4
2	4
3	9
4	9
5	18

1	9
2	9
3	9
4	14
5	20

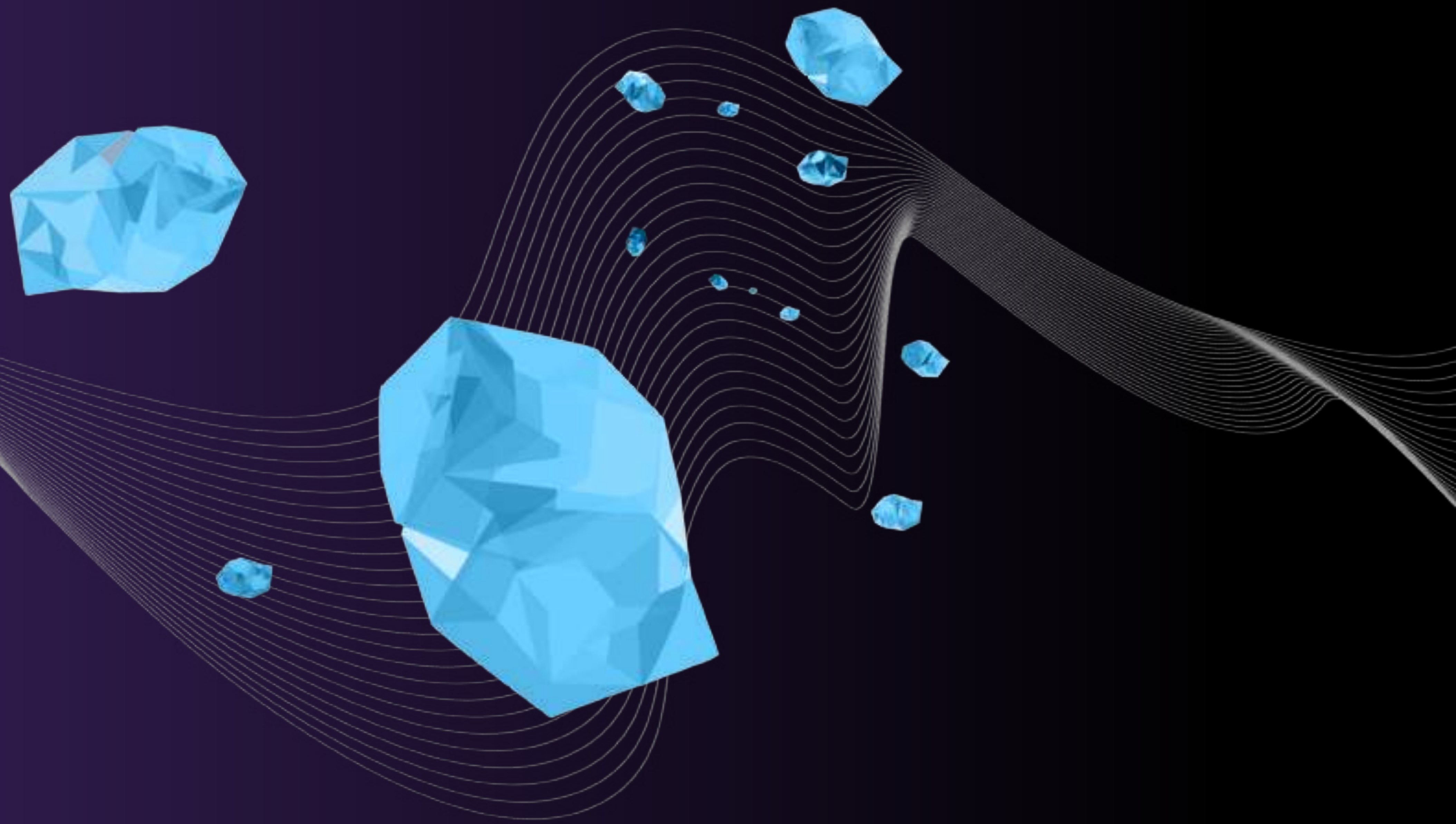
1	11
2	11
3	14
4	18
5	28

1	14
2	14
3	18
4	20
5	28

1	28
2	28
3	28
4	1
5	9

1	21
2	28
3	28
4	28
5	4

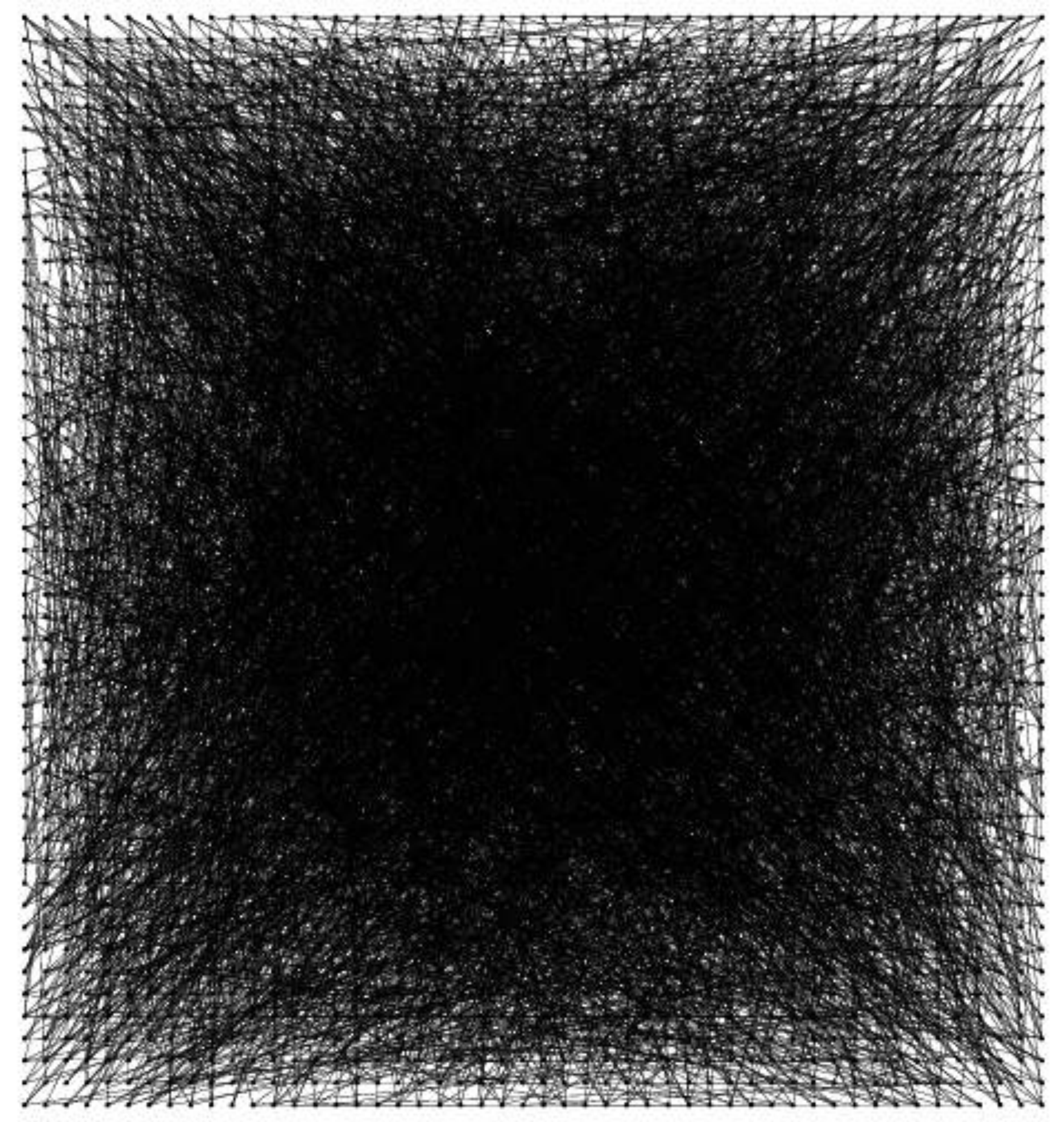
HOW BAD CAN ECLIPSE ATTACKS BE?



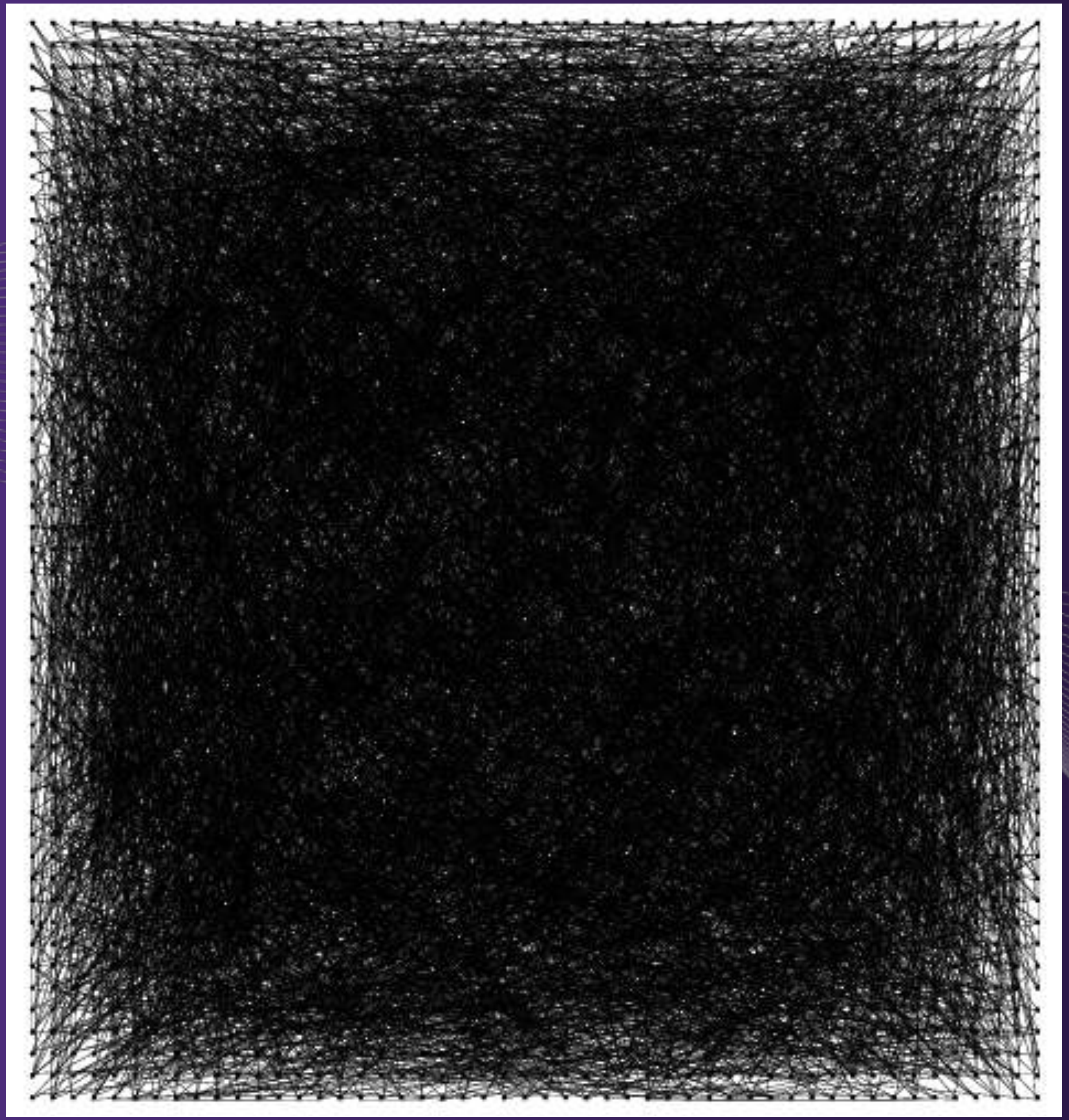
BUILDING OVERLAY NETWORKS

- Consider a collection of nodes that collectively need to construct an overlay network.
- Each node is capable of randomly selecting another node from the network (we’ll get back to this).
- Essence: if nodes can be selective in deciding which links to discovered other nodes, they shoul keep they can construct structured overlay networks.

- The network works in rounds: in each round, each node inspects a randomly selected other node.



- Every node *p* is assigned a group identifier *GID(p)*.
- Goal: partition the overlay into disjoint components (clusters) such that
- $dist(p,q) = \begin{cases} 1 & \text{if } GID(p) = GID(q) \\ 0 & \text{otherwise} \end{cases}$

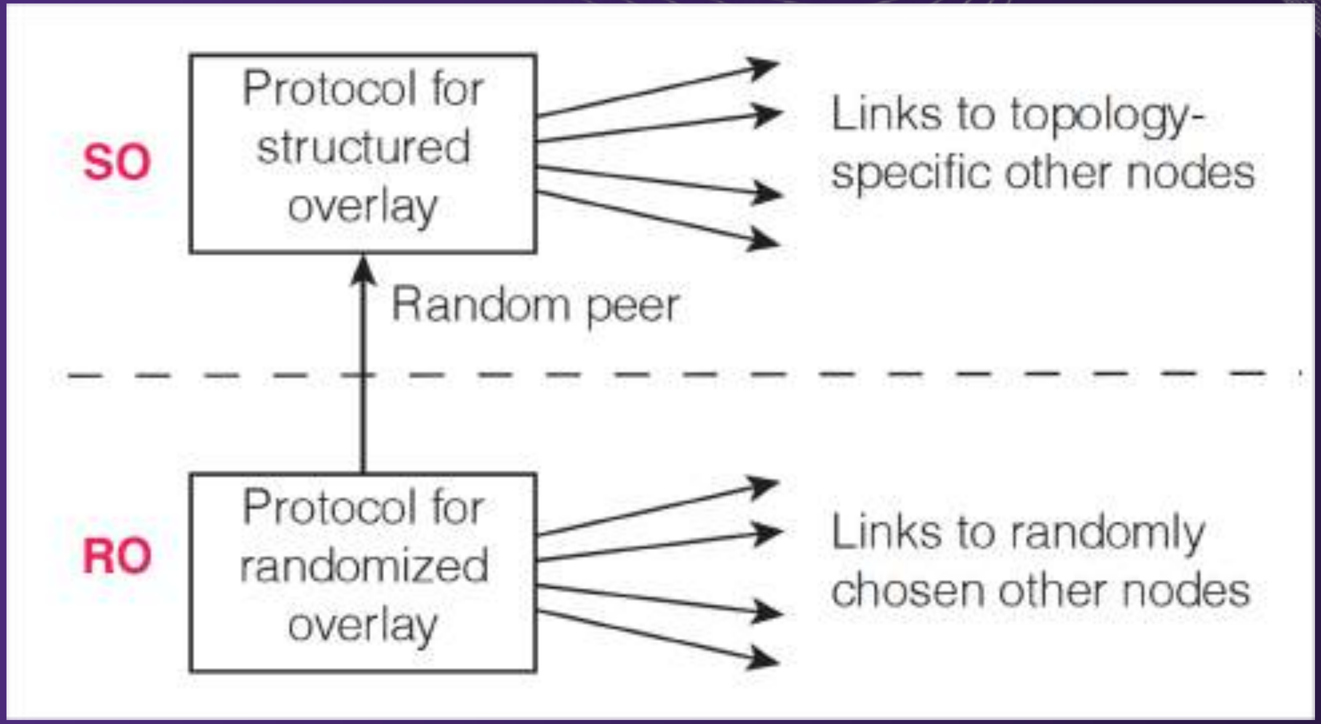


- Each node has an *(x,y)* coordinate and is placed on a 50x50 grid.
- Goal: keep links between *p* and *q* with minimal Euclidean distance:
- $dist(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

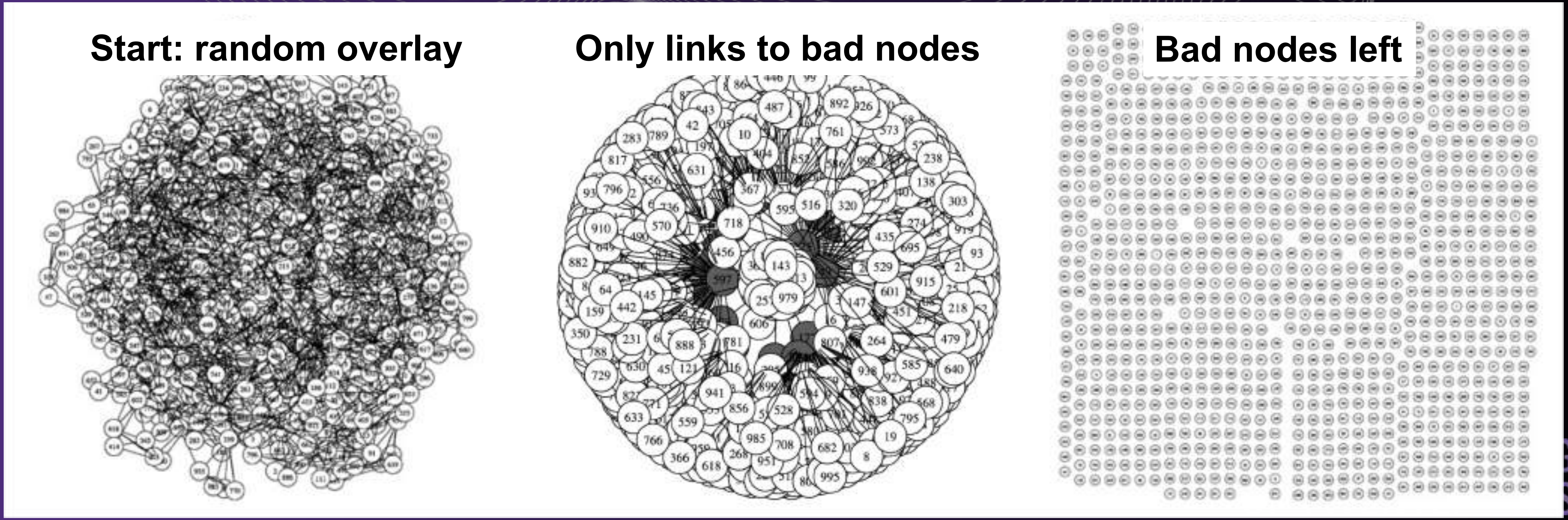
PICKING A RANDOM OTHER NODE

- Each node maintains a (local) list of *c* references to other nodes.
- A node *p* regularly selects a node *q* from its list, and exchanges a number of randomly selected references.
- It turns out that the list appears as a random sample of the entire network


SO
Protocol for structured overlay
Links to topology-specific other nodes
Random peer
RO
Protocol for randomized overlay
Links to randomly chosen other nodes

LET’S ASSUME A FEW COLLUDING MALICIOUS NODES

- When exchanging random references, the colluding node returns references to its malicious friends.
- Within just a few exchanges, all benign nodes are pointing only to malicious nodes: *c = 20; #colluders = 20; network = 1000 nodes*



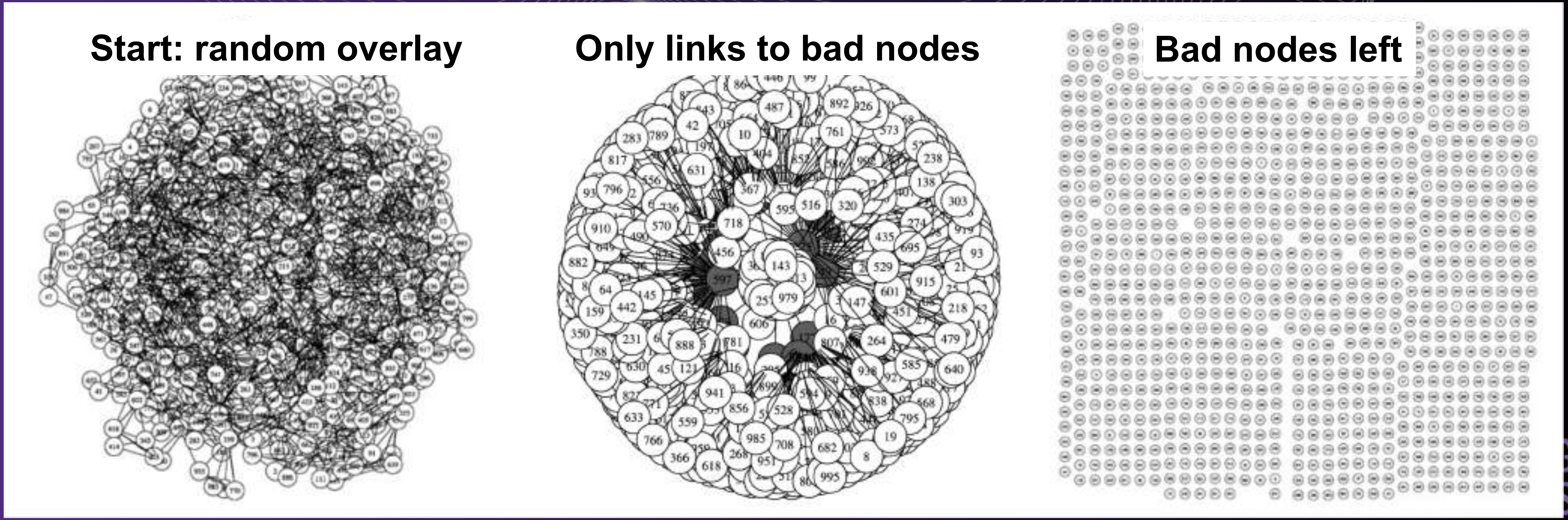
Start: random overlay

Only links to bad nodes

Bad nodes left

EFFECTIVENESS OF THE ATTACK

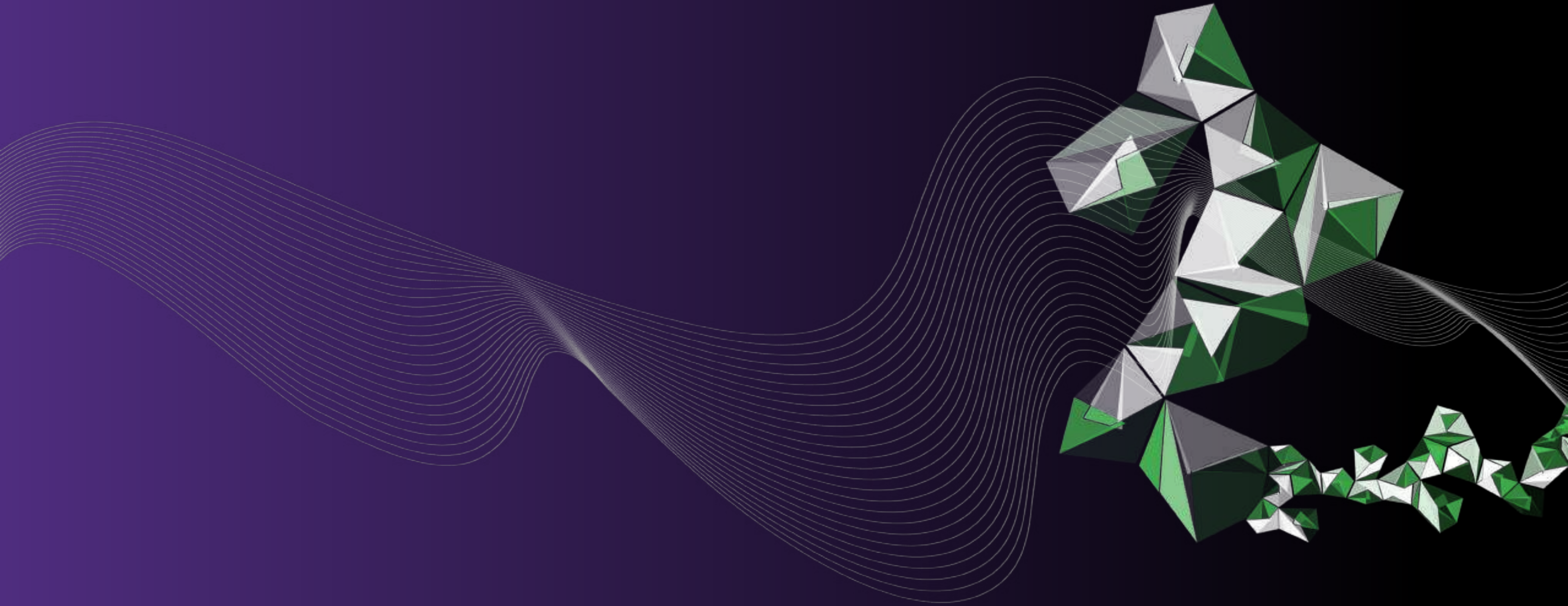
- It takes a mere 20 exchanges per node in a 10,000 node network, to completely partition the overlay.



Start: random overlay

Only links to bad nodes

Bad nodes left



UNIVERSITY
OF TWENTE.

DIGITAL SOCIETY
INSTITUTE

STARTING POINTS & SUGGESTIONS

- A Survey of DHT Security Techniques .
  G. Urdaneta, G. Pierre, M. van Steen.
  ACM Computing Surveys, vol. 43(2), June 2011.
- Contains lots of references toward proposed solutions. Have your pick and make sure you understand those solutions.
- The survey is from 2011. What about updates? Check Google scholar!

- Secure Peer Sampling.
  G.P. Jesi, A. Montresor, M. van Steen.
  Computer Networks vol. 54(12):2086-2098, August 2010.
- Follow the same approach in Google scholar to discover more recent work on eclipse attacks in P2P networks.

- Distributed Systems book
  H1, H2.3, H5.2, H6.7, H9.1, H9.2