



On the Evolution of Adversary Models (from the Beginning to Sensor Networks)*

Virgil D. Gligor
Electrical and Computer Engineering
University of Maryland
College Park, MD. 20742
gligor@umd.edu

Netted Sensors Community Workshop
MITRE Corporation
October 25, 2005

*based on joint work with H. Chan, B. Parno and A. Perrig



Overview

- 1. Resolved: security is a *fundamental* concern... of *secondary* importance**
- 2. Sensor Network security => an insurmountable opportunity to deny both characteristics of resolution 1; (i.e., make security is a *technology* concern of *primary* importance)**
- 3. “*Perfect is the Enemy of the Good*”: why we cannot have perfect security (and should *not* want it)?**



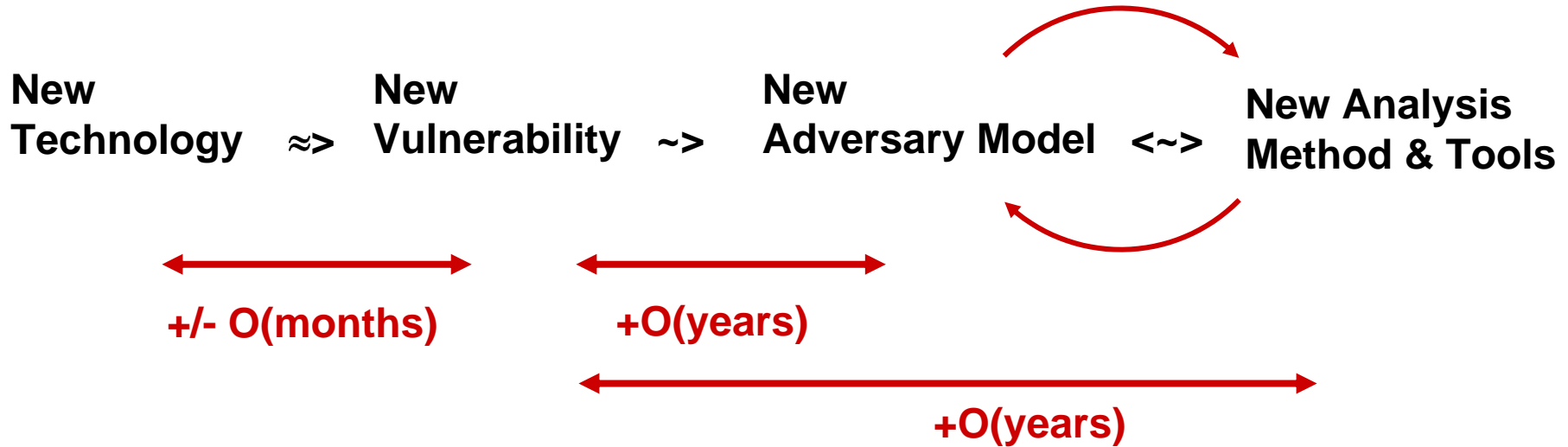
Why a fundamental concern ...

1. New Technology \approx Vulnerability \sim Adversary \leftarrow Methods & Tools

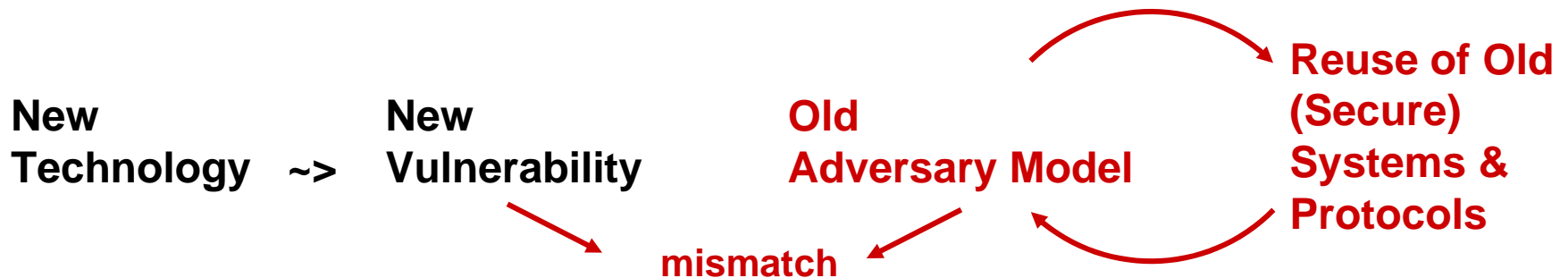
- sharing user-mode programs & data; - computing utility (early – mid 1960s)	confidentiality and integrity breaches; system penetration;	THs in user-mode programs & subsystems	sys. vs. user mode ('62- \rightarrow) rings, sec. kernel ('65, '72) FHM ('75) theory/tool ('91) acc. policy models ('71)
- shared <i>stateful</i> services e.g., DBMS, net. prot. (early - mid 1970s)	DoS instances	untrusted user processes; concurrent, coord. attacks	DoS general def. ('83-'85) formal spec. & verif. ('88) DoS models ('92 \rightarrow)
- PCs, LANs; public-domain Crypto (mid 1970s)	read, modify, block, replay, forge messages	“man in the middle” active, adaptive, mobile network adversary	informal: NS, DS ('78–81) semi-formal: DY ('81) Byzantine ('82 \rightarrow) crypto models ('84- \rightarrow) auth. prot. analysis (87- \rightarrow)
- internetworking (mid – late 1980s)	large-scale effects: worms, viruses, DDoS (e.g., flooding)	distributed, coordinated attacks	virus scans, tracebacks intrusion detection (mid '90s \rightarrow)

2. Technology Cost \rightarrow 0, Security Concerns persist

... of secondary importance



... and a perennial challenge (and opportunity)





Sensor Networks: new *and* real adversary ?

My Claim

Sensor Networks introduce:

- **new vulnerabilities:** (variable number of) nodes captured *and* replicated
new application vulnerabilities (e.g., in distributed sensing)
- **new adversary:** different from both Dolev-Yao and Byzantine adversaries

and

require new methods and tools: emergent algorithms & properties
(for *imperfect but good-enough* security)



Some Characteristics of Sensor Networks

1. Ease of Network Deployment and Extension

- scalability => simply drop sensors at desired locations
- net. + key connectivity => *neither* administrative intervention
nor TTP (e.g., base-station) interaction

2. Communication Constraints on Adversary

- cannot **block-modify-retransmit** message for all receivers
- **receiver anonymity** possible (F. Stajano)
- verifiable **obligation to broadcast** in neighborhood (G. Danezis)

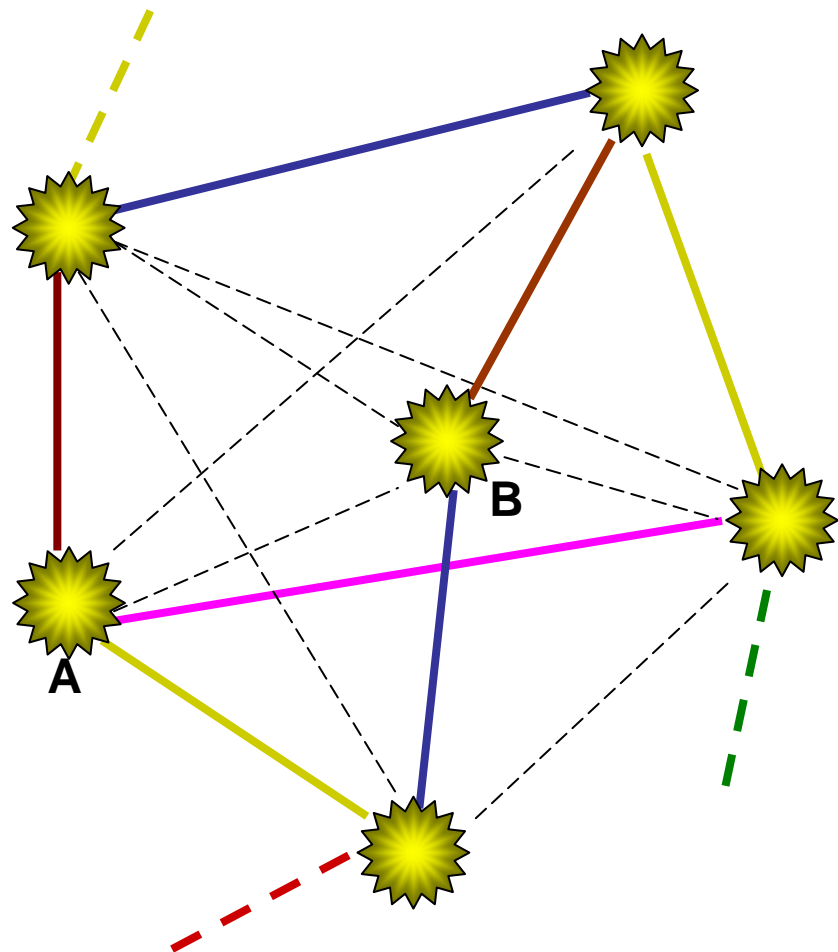
3. Impractical Protection of a Node's Internal State

- low cost => physical node shielding is impractical
=> ease of access to internal node state
- (Q: how good should physical node shielding be to prevent access to a sensor's internal state ? A: Impractically good.)

4. Unattended Node Operation in Hostile Areas => adversary can capture, replicate nodes (and node states)

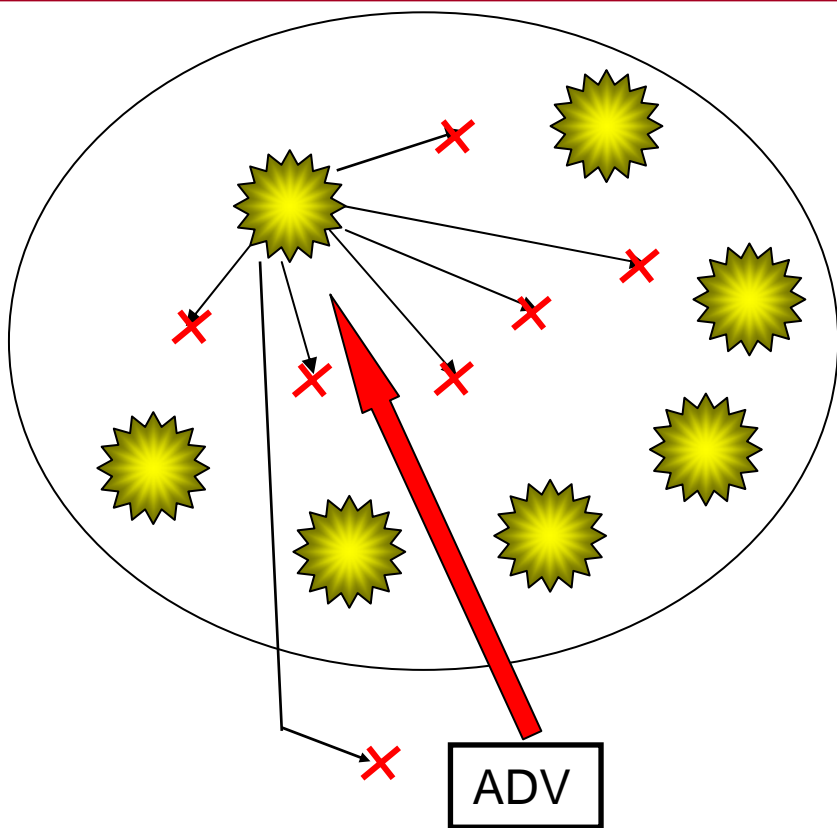
Ease of Network Deployment and Extension

■ An Example: Probabilistic key (pre)distribution



- **key pre-distribution**
 - generation of a *large pool* of P keys
 - random drawing of k keys out of P
 - loading of the *key ring* into each sensor
- **shared-key discovery**
 - upon initialization/extension every node discovers its neighbors with which it shares keys
 - broadcast key IDs; for anonymity
 - broadcast $\langle a, E_k(a) \rangle$ for all keys k
- **Positive Consequences**
 - network self-organization and extension w/o admin. or TTP (e.g., base station) help
- **Negative Consequences**
 - node-to-node authentication
 - limited resilience to node capture
- **Mitigation of neg. consequences**

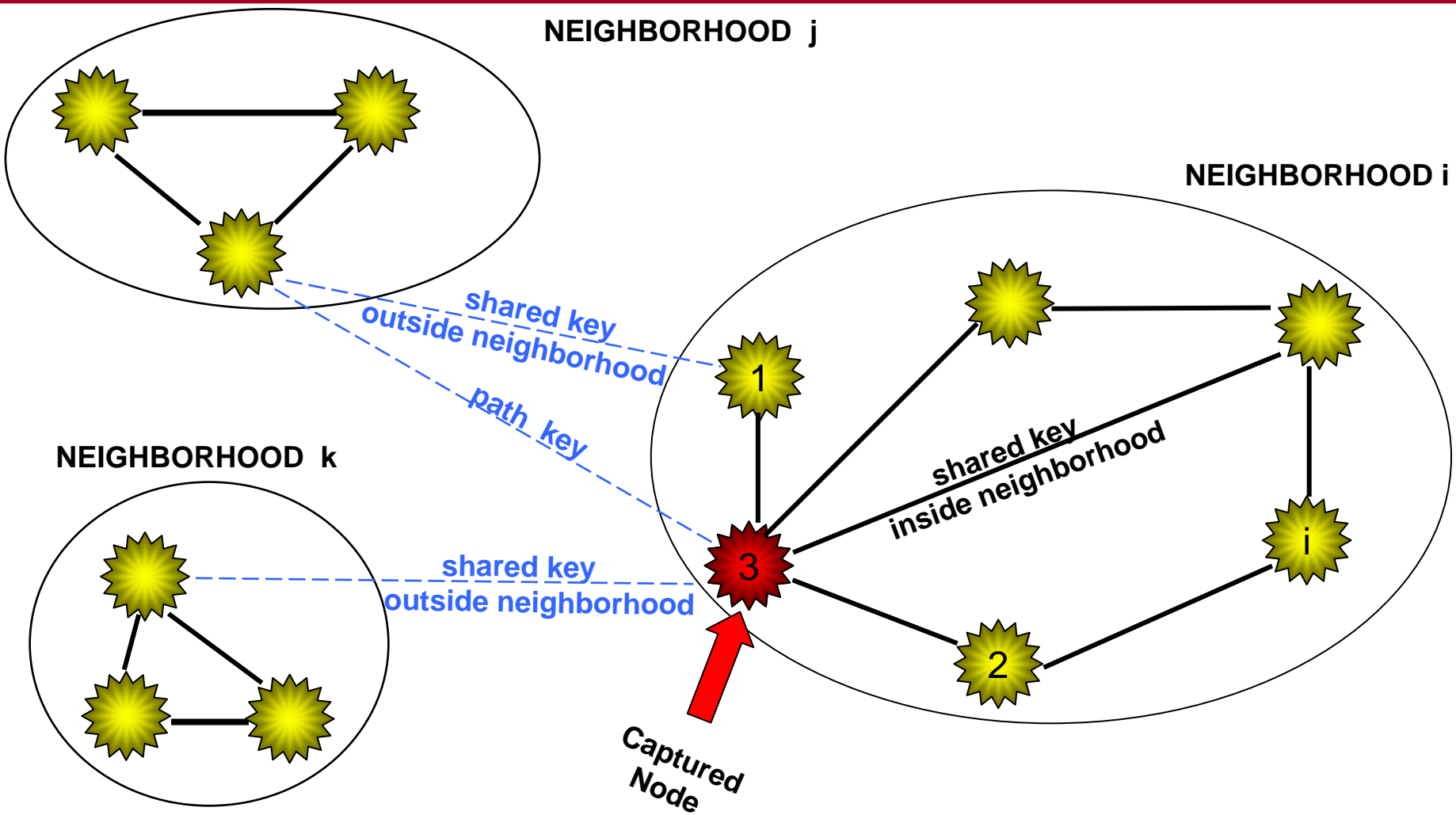
Communication Constraints on Adversary



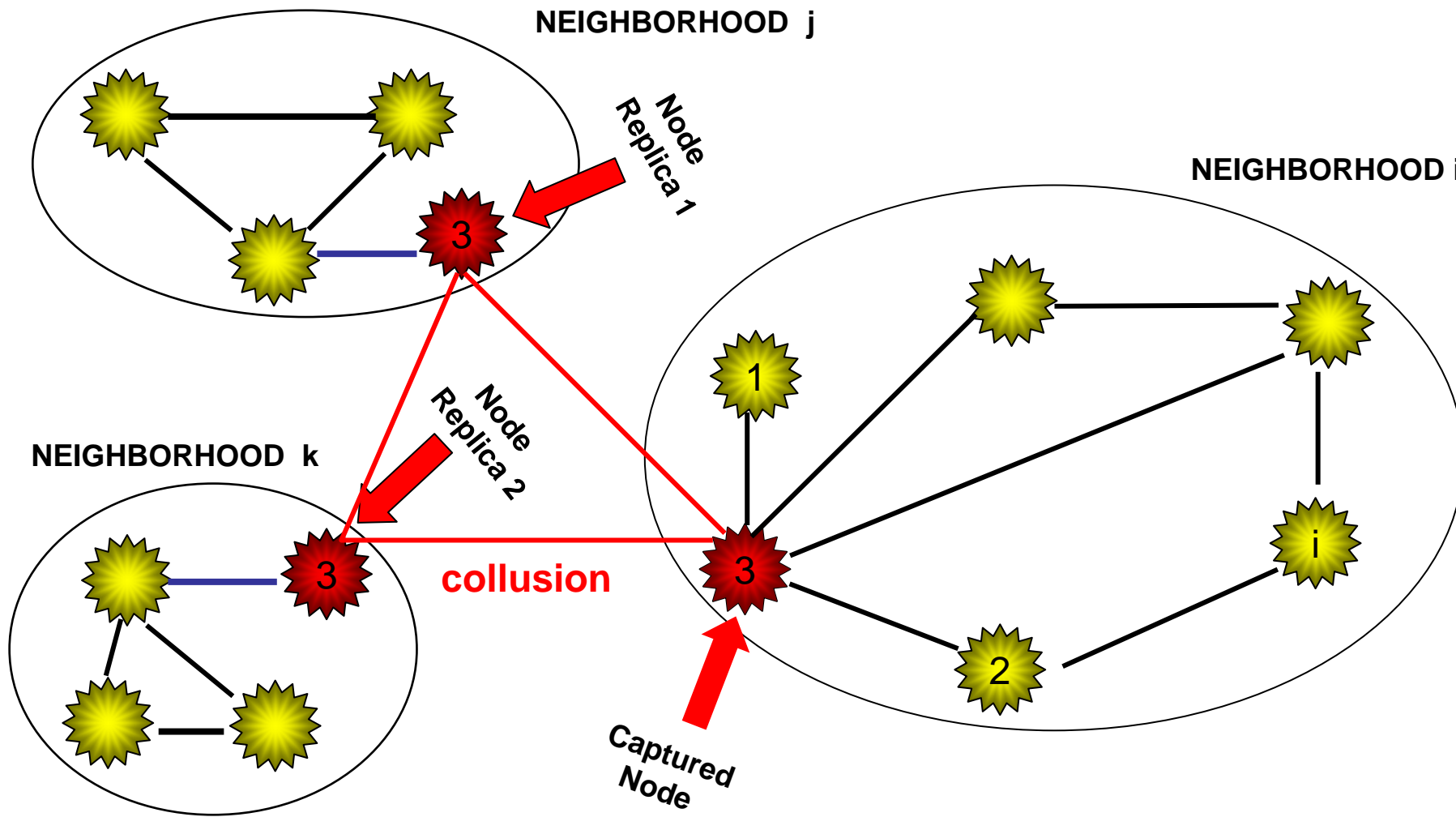
- **cannot block-modify-retransmit message for all receivers**

- **receiver anonymity**
- **if sender is not jammed, some other un-jammed node must have heard sender's message => **verifiable obligation to act****

Replicated Node Insertion: How Easy ?



Attack Coordination among Replicas: How Easy ?



Note: Replica IDs are cryptographically bound to pre-distributed keys and cannot be changed



New (Replication) vs. Old (Dolev-Yao) Adversary

Old (Dolev-Yao) Adversary can

- control network operation
 - **man-in-the-middle**: read, replay, forge, block, modify, insert messages *anywhere in the network*
- send/receive any message to/from any legitimate principal (e.g., node)
- act as a legitimate principal of the network

Old (Dolev-Yao) Adversary *cannot*

- discover a legitimate principal's *secrets*
- *adaptively* capture legitimate principals' nodes
- modify *network* and *trust* topology (e.g., by node replication)

New (Replication) Adversary \neq Old (Dolev-Yao) Adversary

- can block/modify/insert messages *only at* specific (captured, jammed) nodes
- replicated nodes can *adaptively* modify *network* and *trust* topology



Distributed Sensing: A New Application and its Adversary



Distributed Sensing

Application: a set of m sensors observe and signal a global event

- each sensor broadcasts “1” whenever it senses the global event; else, it does nothing
- if t broadcasts, all m sensors signal the event to neighbors; else do nothing
- sensing (detection) threshold $t \leq m$

Operational Constraints

- *absence of the global event cannot be sensed* (e.g., no periodic “0” broadcasts)
- broadcasts are *reliable* and *synchronous* (i.e., counted in sessions)

Adversary Goals: *violate integrity (i.e., issue any set of $t \leq m/2$ false broadcasts, deny service (i.e., $t > m/2$, suppress $m-t+1$ broadcasts)*

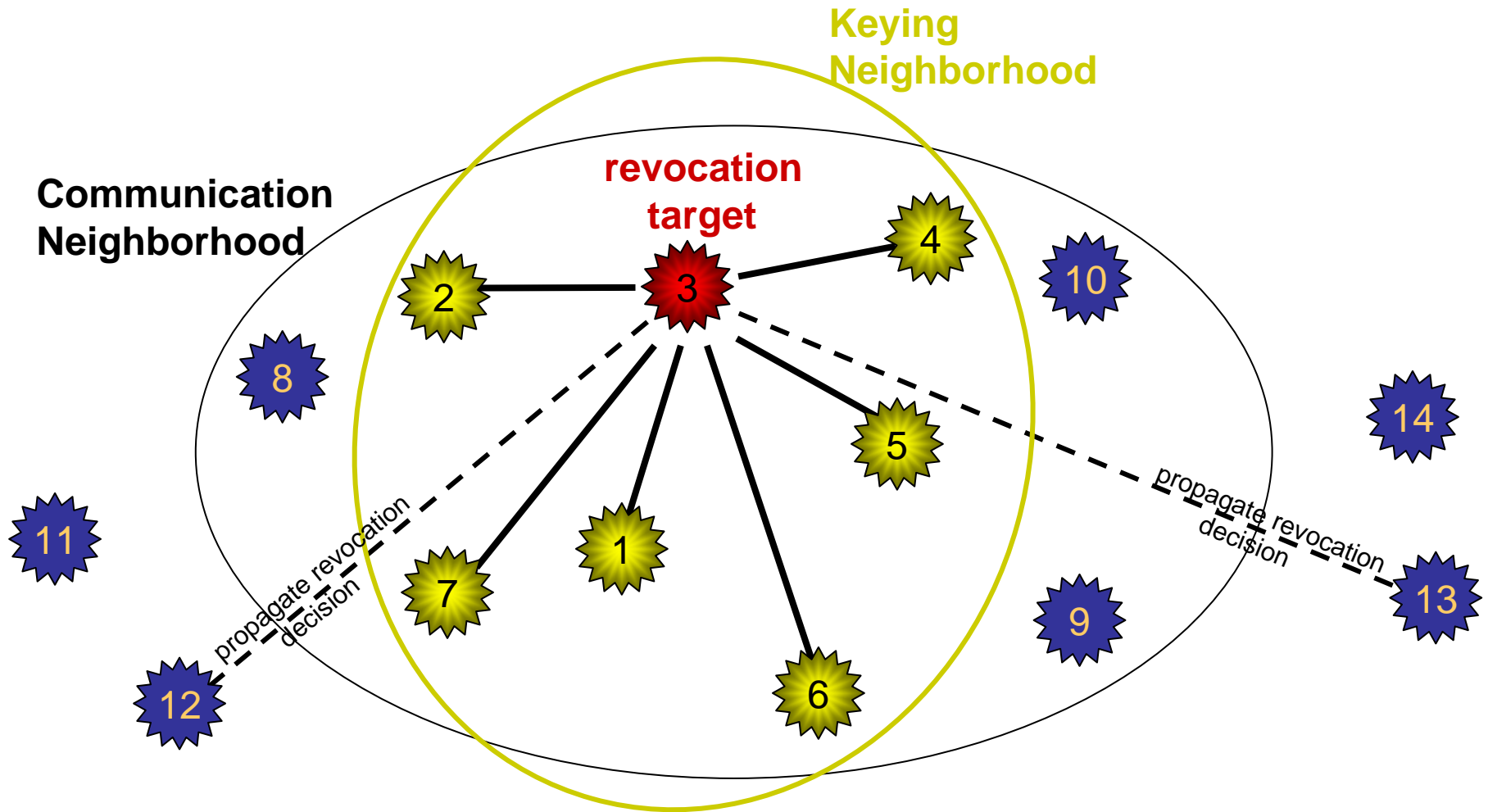
New (Distributed-Sensing) Adversary

- captures insiders (i.e., *any of m*) nodes, forge, replay or suppress broadcasts (within same or across different sessions)
- increases broadcast count with outsiders’ false broadcasts

An Example: *distributed revocation decision*

[IEEE TDSC, Sept. 2005]

$m=6, t = 4$ votes in a session => revoke target





New (Distributed Sensing) vs. Old (Byzantine) Adversary

Q: A Byzantine Agreement Problem (w/ *similar* operational constraints)?

- **both global event and its absence are (“1/0”) broadcast by each node**
- **strong constraint on t ; e.g., no PKI $\Rightarrow t > 2/3m$; PKI $\Rightarrow t > m/2$**
- **broadcasts are *reliable* and *synchronous* (i.e., counted in sessions)**

A: No. Byzantine Agreement Problem \Rightarrow

- \Rightarrow Constrained Distributed Sensing (i.e., “1/0” broadcasts, constrained t)
- \Rightarrow Distributed Sensing

New (Distributed-Sensing) Adv. \neq Old (Byzantine) Adv.

- new adversary need *not* forge, initiate, or replay “0” broadcasts
- new adversary’s strength depends on a weaker t (e.g., $t < m/2$)
- new adversary may modify membership to increase broadcast count ($> t$)

Note: Replication Adversary must also be countered

- Replication Adversary \Rightarrow membership violation
(not possible with Byzantine Adversaries)



Emergent Vulnerabilities

1. Collusion to Subvert Applications

- Ex. 1: subvert aggregation of sensor data; replicated nodes *block* legitimate transmissions, *modify* legitimate data and *inject false data*
- Ex. 2: subvert “distributed sensing”; e.g., *sense false events*

2. Collusion to Subvert Network Operation

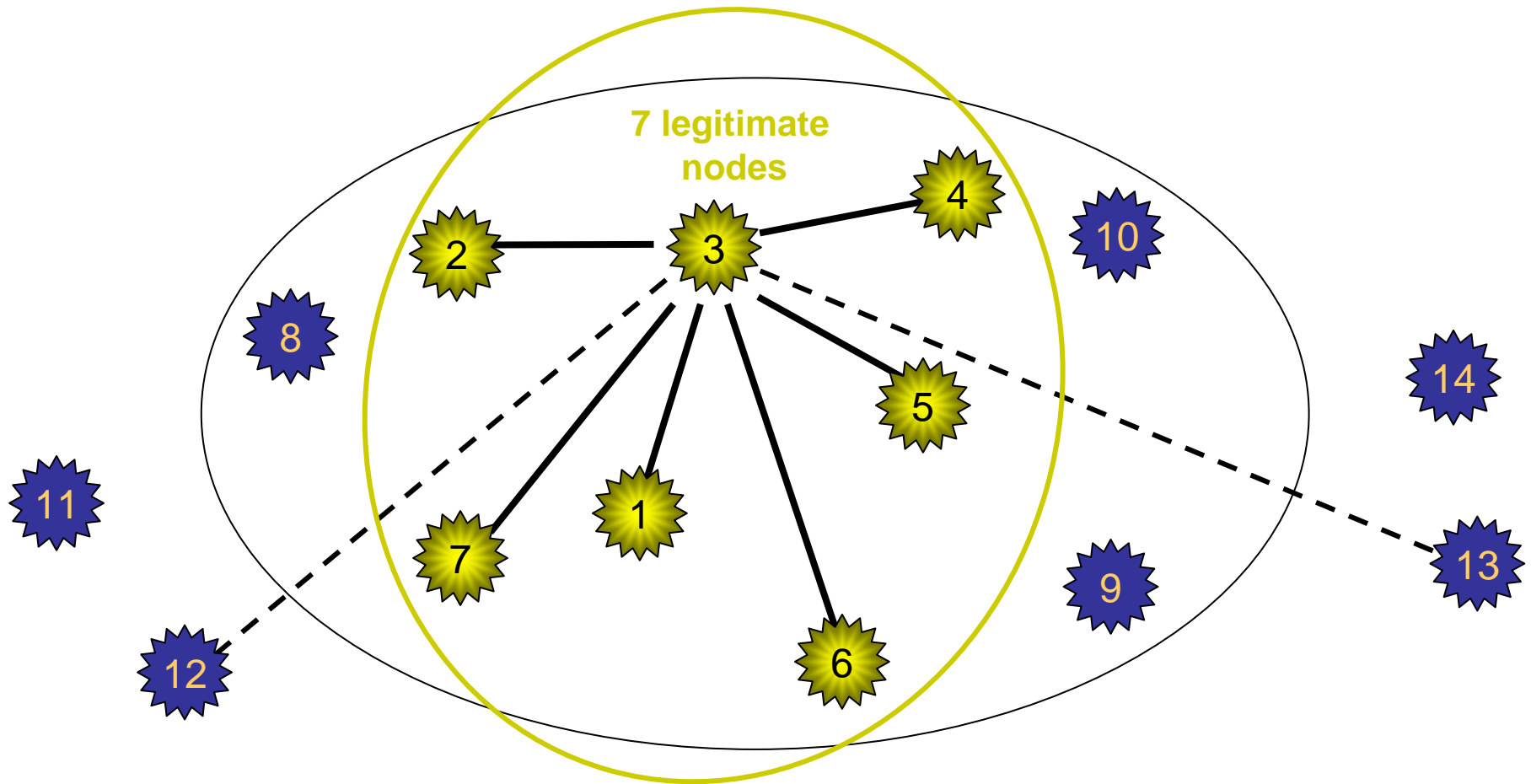
- Ex. 1: replicated nodes block traffic to *partition the network*
- Ex. 2: captured nodes revoke legitimate nodes by execution of revocation protocol and cause *loss of secure connectivity*

3. Circumvent Intrusion Detection (and net’s “immune” system)

- Ex: *spread abnormal behavior* over multiple replicas to avoid detection

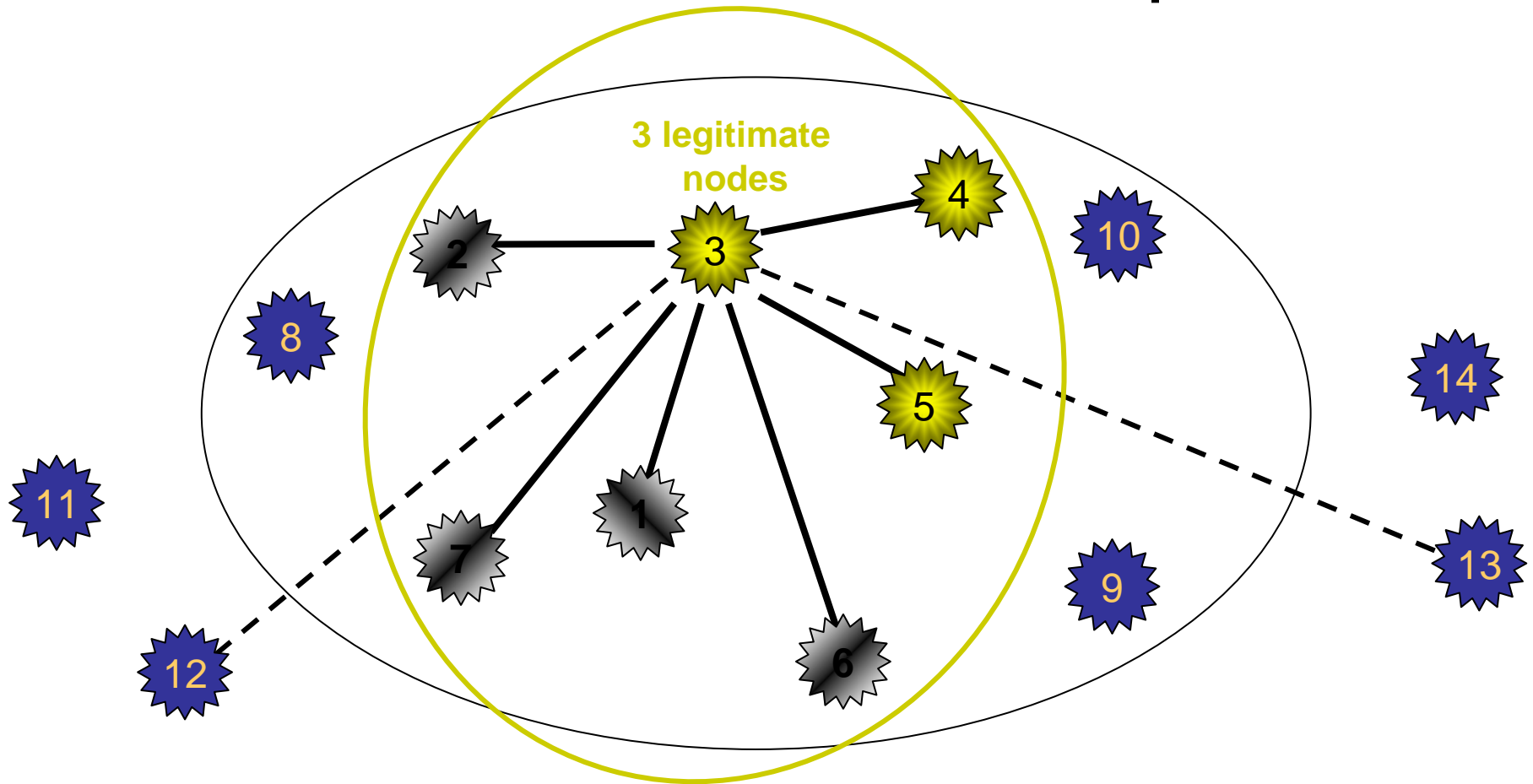
Emergent Vulnerability Example

Initial State: Secure Connectivity



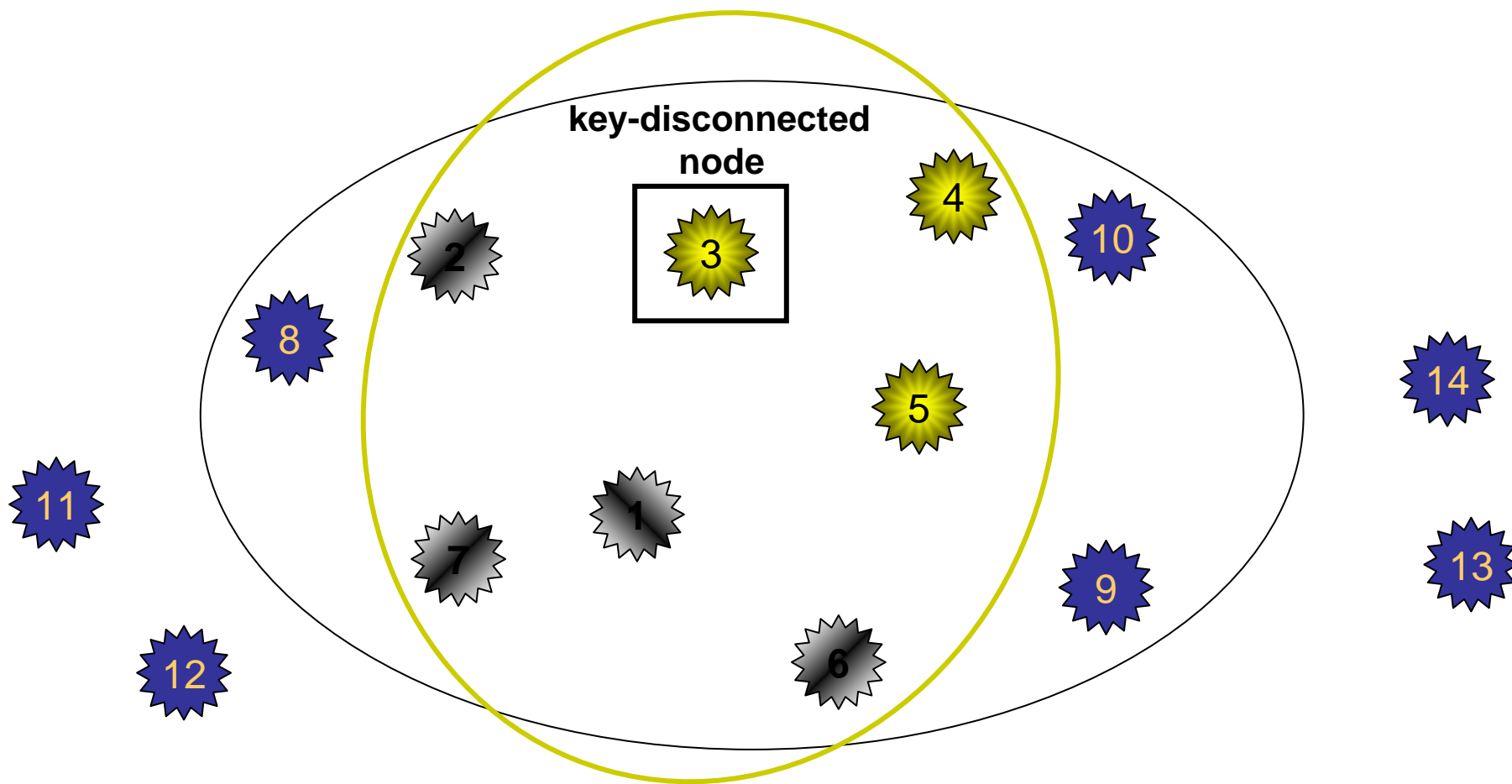
Emergent Vulnerability Example

Intermediate State: $m=6$, $t = 4$ compromised nodes execute revocation protocol



Emergent Vulnerability Example

Final State: Lost Secure Connectivity





Conclusions

- 1. Q: Sensor Networks: New *and* Real Adversary ?**
A: Yes. (And, in any case, we must avoid “failures of imagination”)
- 2. Challenge => Opportunity**
 - anticipate new Vulnerabilities and define new Adversary Models *before* network deployment
 - reexamine of analysis Methods and Tools *if (old crypto) protocols are reused*
- 3. “Perfect is the Enemy of the Good”**
 - some (new) adversary attacks => “emergent undesirable properties” => desirable “*emergent algorithms and properties*” as countermeasures (viz., Parno, Perrig, Gligor, 2005 IEEE Security and Privacy)
 - must accept *probabilistic security properties (with non-asymptotic 0/1 probabilities)*