NSL

UNIVERSITY OF WASHINGTON

Private Identification in RFID; Defining Vulnerability Metrics; Secret Keys using Channel Reciprocity

Bertinoro PhD. Summer School, July 2009

Radha Poovendran

Network Security Lab

Electrical Engineering Department

University of Washington, Seattle, WA

http://www.ee.washington.edu/research/nsl

NSL

Outline

- Introduction to RFID systems
- The RFID controversy
- The RFID paradox
- Degrees of privacy
- Classification of private protocols
- Conclusion and open problems

NSL

Radio Frequency Identification

- Radio Frequency Identifications (RFIDs) enable the unique identification over the wireless medium
- RFID systems consist of three main components
  - Tags
  - Readers
  - Central storage (database)

NSL

Radio Frequency Identification

- Tags
  - Active: have their own battery
  - Passive: use the received radio waves as power source
  - Typically, tags have limited computational power (especially passive ones)
- Readers and the database are computationally powerful devices
- Communications between readers and database is assumed to be secure in most applications (either via wired connections or secure cryptographic primitives)

Passive tag

Active tag

Reading tags

RFID Basic Operations

- Reader interrogates tags in its vicinity (and charge them if they are passive)
- Each tag responds with a quantity that enables the reader to uniquely identify it
- Readers access the database and obtain information about interrogated tags

NSL

The RFID Controversy

- Tags respond to readers' interrogations without their owners' approval nor awareness
- Tags can be tracked by illegitimate readers, thus, potentially violating their owners' privacy
- Privacy activists call RFID tags "spy chips" and "tracking devices"
- In response to privacy activists organizing to boycott, Benetton once officially repudiate any RFID testing plans

NSL

RFID Requirements

- There are three basic requirements in RFID systems
  - Identification
    - By itself, can be as easy as broadcasting identifiers in clear text
  - Privacy
    - Prohibits broadcasting identifiers in clear text
    - Identifiers must be randomized to avoid unauthorized tag identification
  - Security
    - Requires that tags' responses be long enough (to mitigate cryptographic attacks, such as random guess and exhaustive search)

NSL

Problem Statement

Design RFID systems that
allow unique tag identification,
while preserving tags' privacy,
and secure against cryptographic attacks

NSL

Degrees of Privacy

- **Universal Untraceability**
  - Tags' responses after a protocol run with an authorized reader is uncorrelated to previous responses
  - An adversary observing authorized reader-tag interactions cannot identify tags
  - Implies privacy against passive adversaries
- **Existential Untraceability**
  - Tags' responses cannot be correlated by unauthorized readers
  - An adversary interrogating the same tag consecutive times cannot correlate its responses
  - Implies privacy against active adversaries

NSL

The RFID Paradox

- Computationally powerful devices can provide private identification rather easily
  - Ex: the tag can encrypt a randomized version of its identifier with the reader's public key, and only the reader can extract the encrypted identifier
- However, low-cost tags in most application are unable to perform public key cryptography!
  - Low-cost RFID systems are usually restricted to the use of symmetric key cryptography

In one hand, tags are required to encrypt their identifiers with their corresponding secret keys

On the other hand, the reader must know the tags' secret keys to decrypt their response and extract their identities

NSL

Identification Classifications

- RFID protocols can allow identification in one of two methods
  - synchronous identification
    - Tags' identifiers are updated, both by readers and tags, after every successful protocol run
    - Unauthorized observers cannot correlate updated and outdated identifiers
    - Privacy is gained by the ability to complete protocol runs
    - Active adversaries interrogating the same tag consecutive times will receive the same response
    - Only universally untraceable
    - Also known as stateful protocols
  - asynchronous identification
    - No synchronization is required
    - Privacy is gained by tags ability to generate random numbers
    - Can provide existential untraceability

NSL

Time Complexity Classification

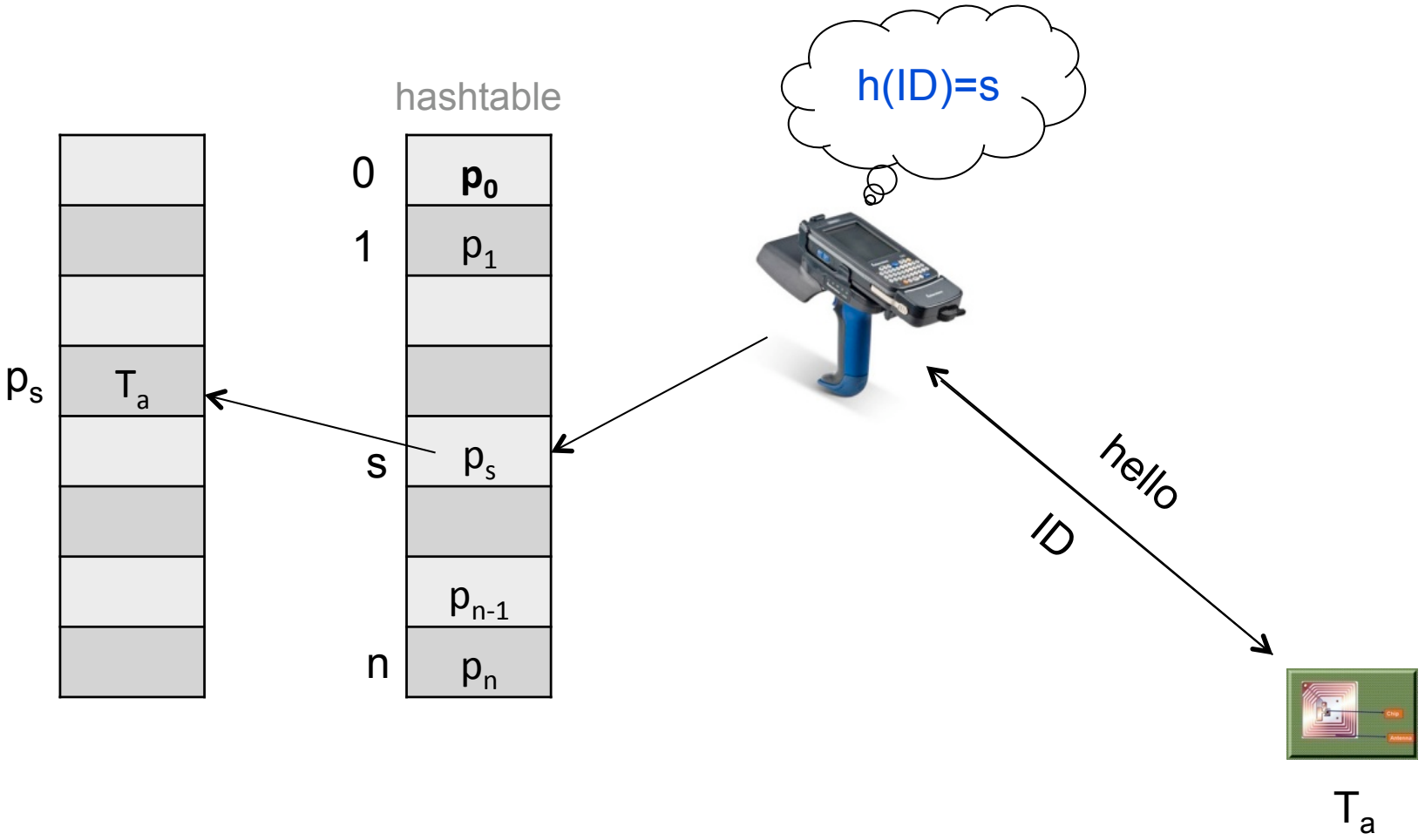
- Based on their scalability, RFID protocols can be divided into three main classes
  - Constant-time identification
  - Linear-time identification
  - Logarithmic-time identification

NSL

Constant-time Protocols

- From the valid reader's standpoint, in synchronous protocols, each tag in the system has only one possible response
- Therefore, tags can always be identified in constant time
- Even if tags' identifiers are sufficiently long and no practical storage can be built to accommodate all possible responses, data structures like hashtables can be used to allow constant-time identification

Example



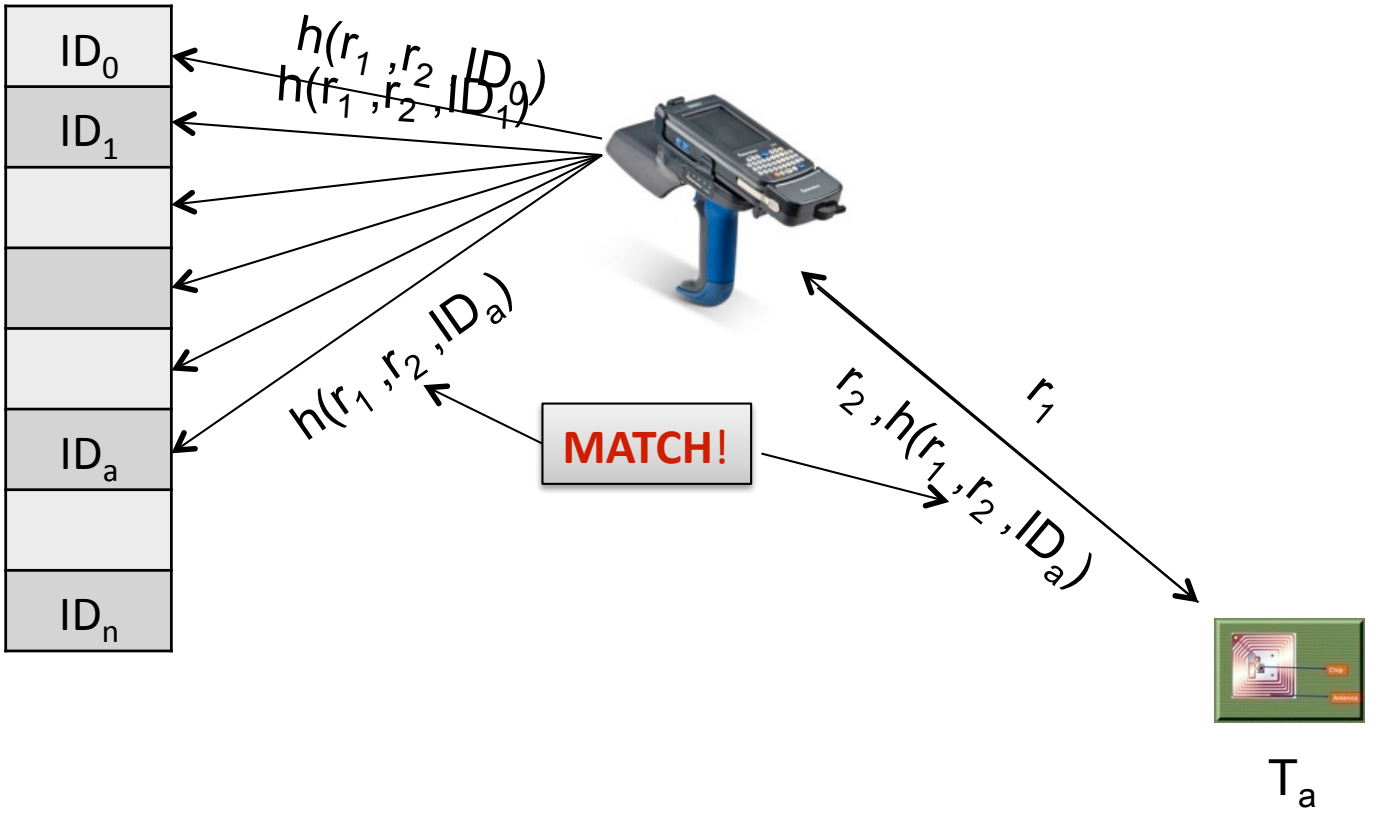
NSL

Linear-time Protocols

- In asynchronous protocols, unlike synchronous ones, tags' responses can be any possible string
- That is, since tags' responses are randomized, each tag's response can be any possible string of the length of the identifier (exponential in the length of the response)
- Since the length of tags' responses is sufficiently large, no physical storage can be built for direct addressing
- To overcome this problem, most RFID protocols in the literature compromise by incurring more computational overhead on the database
- Given the tag's response, the reader must search all tags in the database to identify each response

Example

Need to perform linear search for every identification



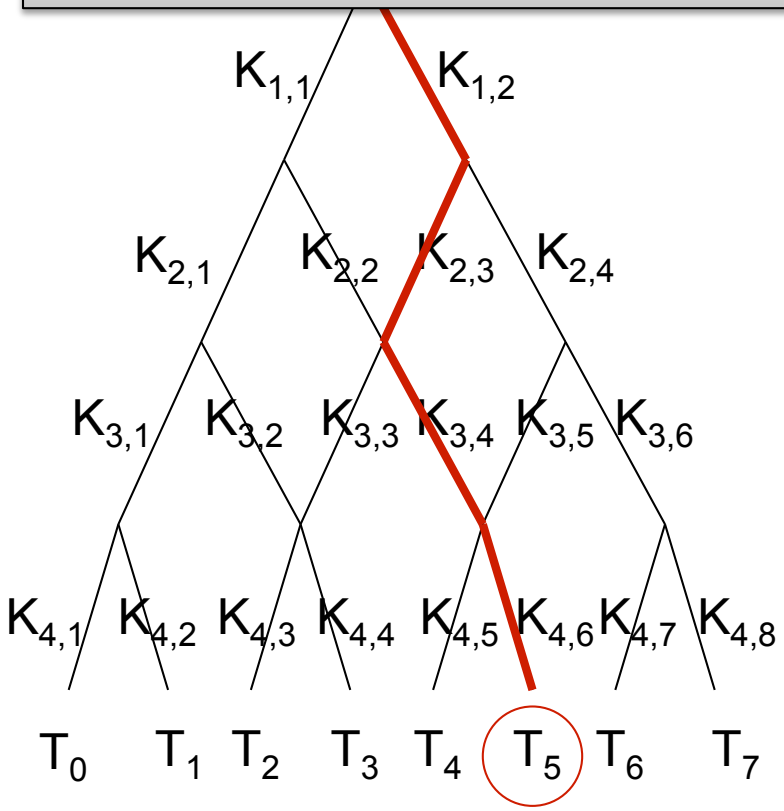
NSL

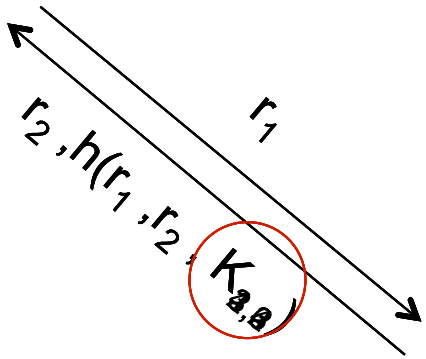
Logarithmic-time Protocols

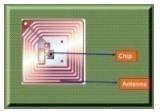
- For large-scale systems, performing linear search on all tags in the system to identify a single response can be very time-consuming
- Logarithmic-time protocols have been proposed to reduce identification complexity
- The reduction is achieved by trading-off more communication and computation overhead

Example

After every reader-tag interaction the reader can move down one level in the tree







T_5

Tags are identified in logarithmic time using logarithmic interactions

NSL

Logarithmic-time Protocols

- Reducing identification complexity from linear to logarithmic, while maintaining existential untraceability is a milestone in the design of RFID systems
- However, apart from the extra communication overhead, it introduced a new security threat
- Since tags are arranged in the tree based on their secret keys, compromising a single tag will reveal secret information about other uncompromised tags
- Several papers have been devoted for the analysis and the mitigation of tag compromise attacks in tree-based RFID systems
- Still an open research area

NSL

Conclusion and Open Research

- RFID protocols can be synchronous or asynchronous
- Synchronous protocols can allow for constant-time identification
- Asynchronous can be identified in linear-time or logarithmic time
- Synchronous protocols are only universally untraceable while asynchronous can be existentially untraceable
- Logarithmic-time protocols, while provide practical identification time, introduced a dangerous compromise threat
- An open research problem in the design of low-cost RFID systems is the design of constant-time protocols that provide existential untraceability; and the mitigation of the compromise attack in tree-based systems

NSL

Ongoing work on Network Vulnerability Metric

- Quantify network resilience to failures and/or attacks
- Produce network level metric using node and link level information
- Compute the impact of a node or link removal on the overall network (sensitivity/gradient)
- Compare two different networks

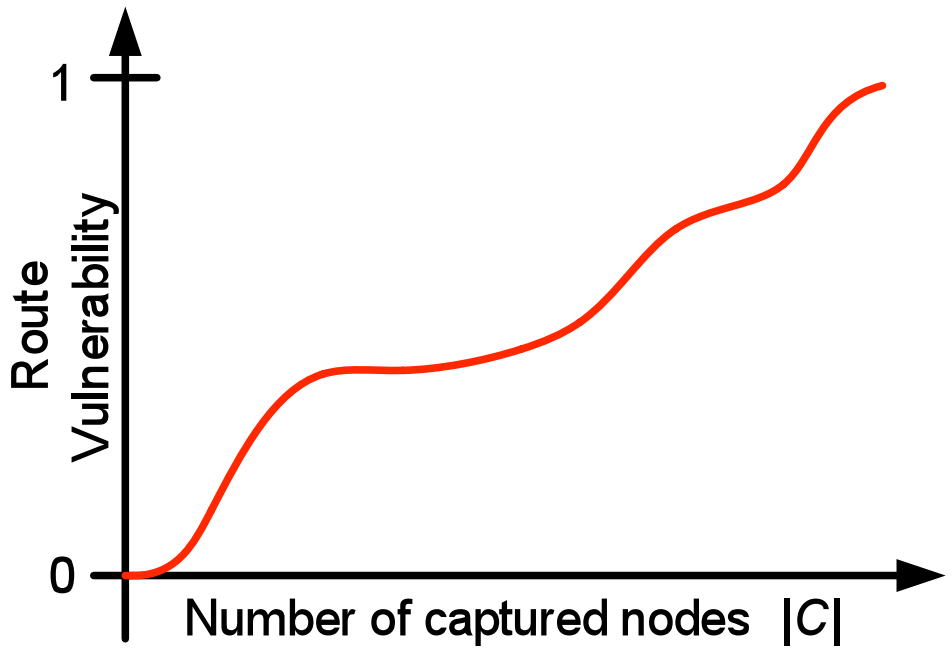
NSL

Ongoing work on Network Vulnerability Metric

- Understanding attack progress on a network
- Need a common form for security as well as network flows

Evaluating Degradation

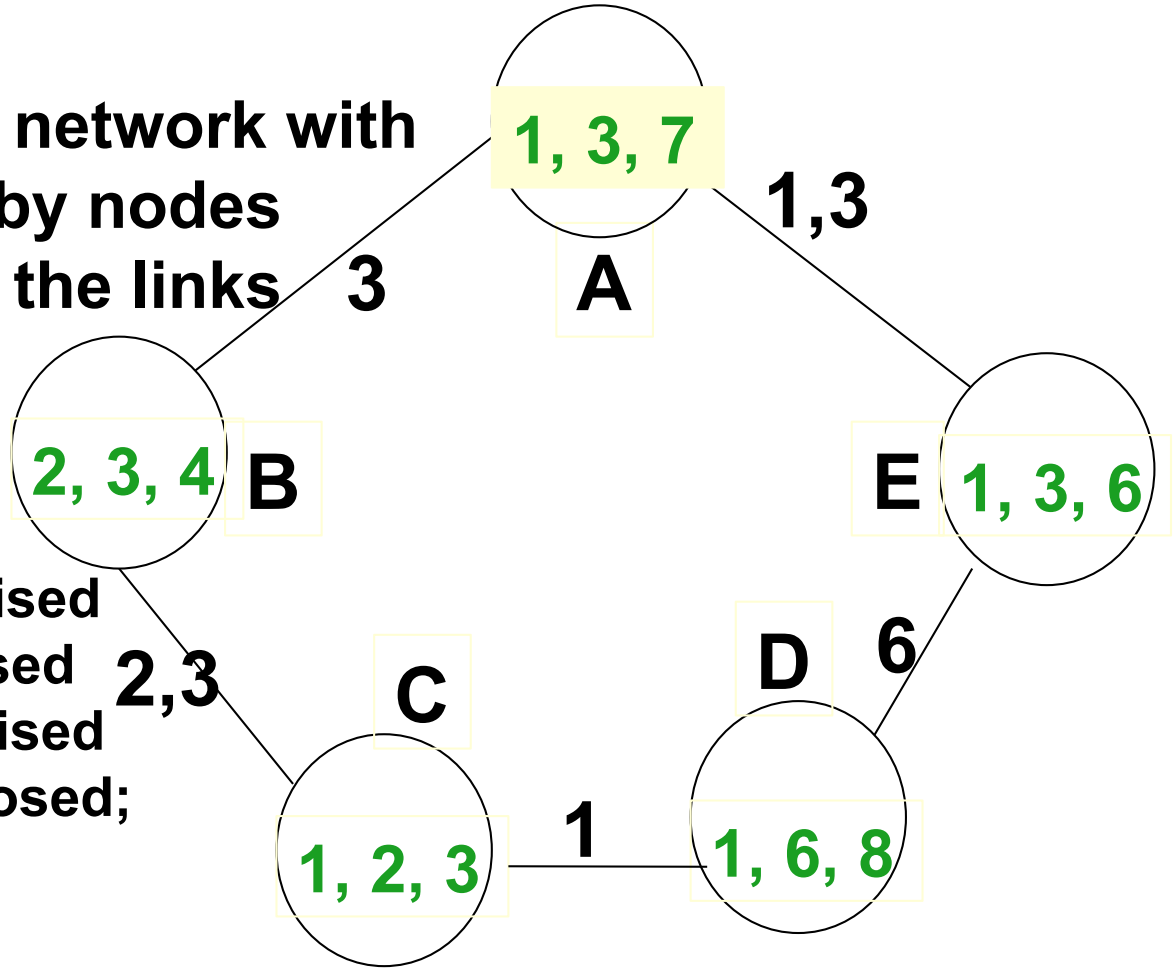
- "*Route vulnerability*"
  - Compromise is binary
  - Real-valued function $h_C(s,d)$ for route R_{sd}
  - Captures *normalized* degradation in data security



Route vulnerability *depends implicitly on network topology*

Example 1: Node capture attack

Figure shows a five node network with
-Indices of the keys held by nodes
-Indices of keys securing the links



If node A were to be compromised
-Links B-A; A-E; C-D are exposed
If node D were to be compromised
1. Links C-D and C-E are exposed;

Link B-C is partially exposed.

Example 2: Case of Two Networks

- **Two networks with the same parameters (network topology, size of key pool, number of keys per node) but different vulnerability to node capture**

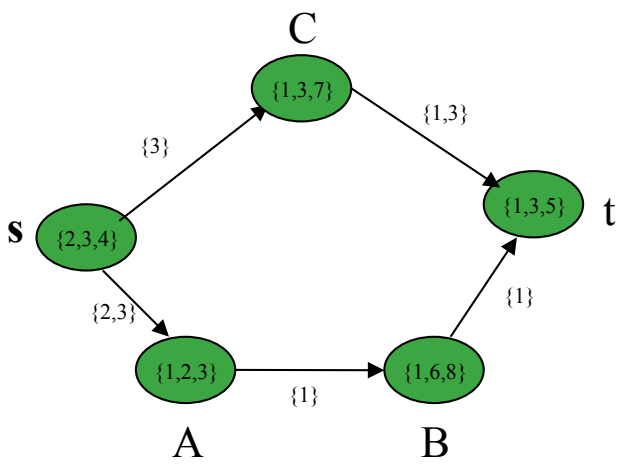


Fig 1 : Network 1 (The “bad” one)

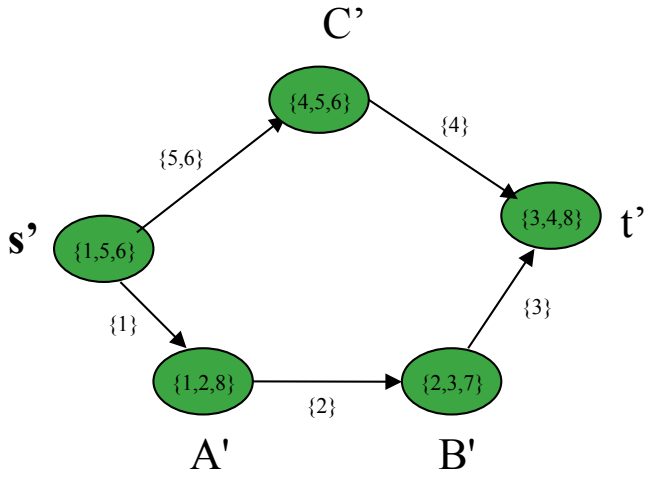


Fig 2: Network 2 (The “better” one)

NSL

A Choice of link vulnerability

- **Number of nodes holding a link key is not a good expression for the vulnerability of a link**
  - **We care mainly about the security of the strongest keys, so the info for all keys overstates the vulnerability**
  - **There may be overlap, e.g. one node may hold several of the keys securing the link, and would therefore be counted multiple times**

NSL

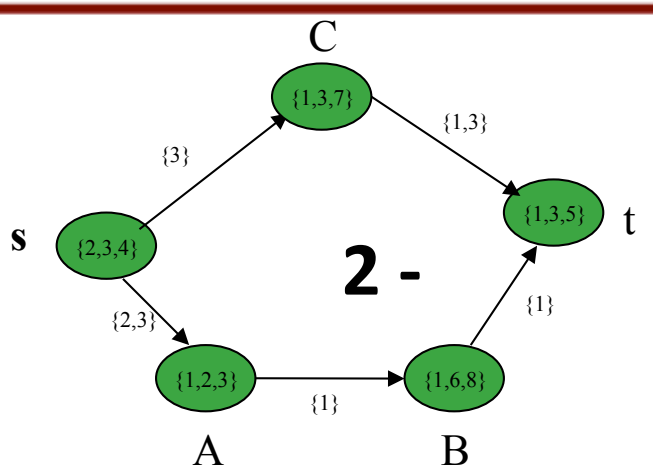
An *inclusion-exclusion* based link weight

1. Set vulnerability weight of link (i,j) to 0
2. Form sets of nodes holding each key securing the link (i,j)
3. Identify the nodes that hold the largest number of keys, and for each such node, add the fraction of the keys held by such node to the vulnerability weight
4. Remove the node from the list and repeat steps 2-4 until one of the sets of nodes holding each key is empty. Link weight at the termination is a measure of link vulnerability (link conductance) or resiliency (link resistance)

An Example of Computing the weight metric for a link

- **Link S-A is secured by keys {2,3}**

1. **Link conductance W(s, A) =0**
2. **Form node sets as:**
   > {s, A} ; 3 -> {s, A, C, t}
3. **Keys 2&3 share nodes (s, A)**
   w(s, A) =2;
4. **Updated node sets are: 2 -> { }; 3 -> {C,t} and algorithm terminates**

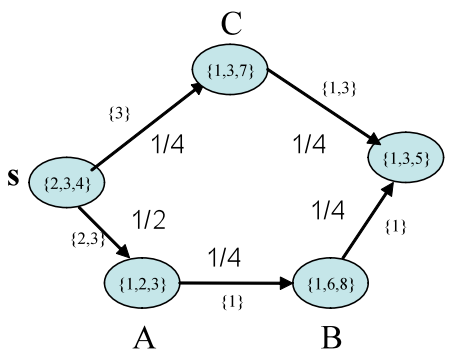


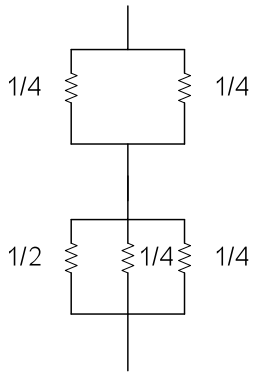
- **The conductance through link s-A is 2 and the resistance is ½=0.5**

From Link Resistance to Security Metric

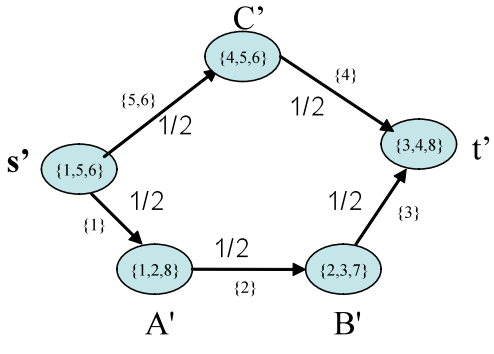
- Data flow from node s to t that passes through paths P_1, P_2,…, P_r.
- The resistances in each path are scaled by the fraction of flow passing through that path
- We put the resistances of the links comprising each path in parallel
  - This ensures that the security of the path is dominated by the security of the weakest link
- Independent paths are put in series
- The resiliency of the traffic from s to t is the effective resistance between s and t in this graph

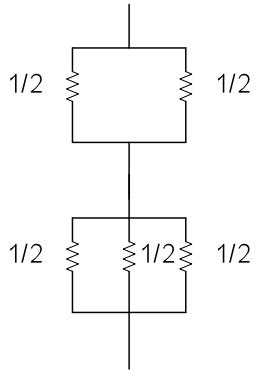
Example of metric computation





R = 0.225





R = 0.417

Since 0.417 > 0.225, our metric classifies the graph on the right as more secure , agreeing with intuition

From Link Metric to Global Properties: Spectral Properties

- The effective resistance between two nodes in a graph can be expressed in terms of the ***Laplacian*** *matrix* L, defined as follows:

L_{ij}
$= -1/r_{ij}$ **if (i,j) is an edge**
$=$ **0 if (i,j) is not an edge**
$= \sum 1/r_{ik}$, **i=j**

summation is taken over all edges k incident on i.

- One can show that the effective resistance between two points (s,t) in a network is given b

$R(s,t) = \sum_k (1/\lambda_k)(u_{ki} - u_{kj})^2$

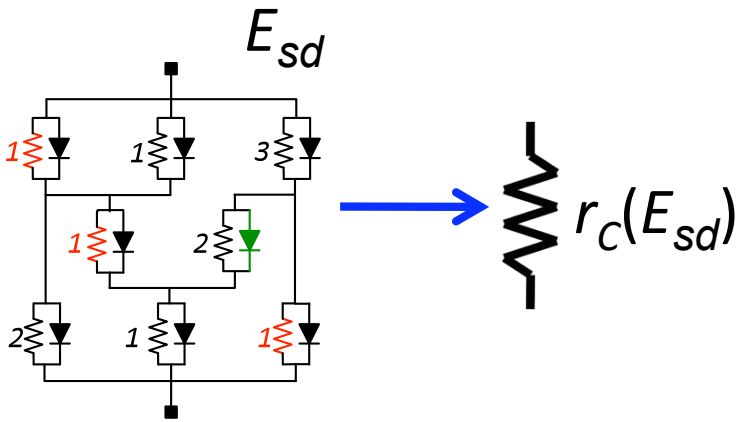
NSL

From Link Metric to Global Properties: Spectral Properties

- Note that $R(s,t) = \sum_k (1/\lambda_k)(u_{ki} - u_{kj})^2$ allows resistance to be –ve as well
- Can define the link resistance in more than one way but needs work

A Circuit Theory Based Vulnerability Metric

- Equivalent resistance of circuit measures overall data security
  - Construct electric circuit E_{sd}
  - Equiv. resistance $r_C(E_{sd})$



- **Circuit-theoretic vulnerability metric** $h_C(s,d)$ from $r_C(E_{sd})$
  - Def: $h_C(s,d) \sim r_C(E_{sd})^{-1}$, normalized w.r.t. $C = \emptyset$

$$h_C(s,d) = \frac{1}{r_\emptyset(E_{sd})}\left(\frac{r_\emptyset(E_{sd}) + 1}{r_C(E_{sd}) + 1}\right)1$$

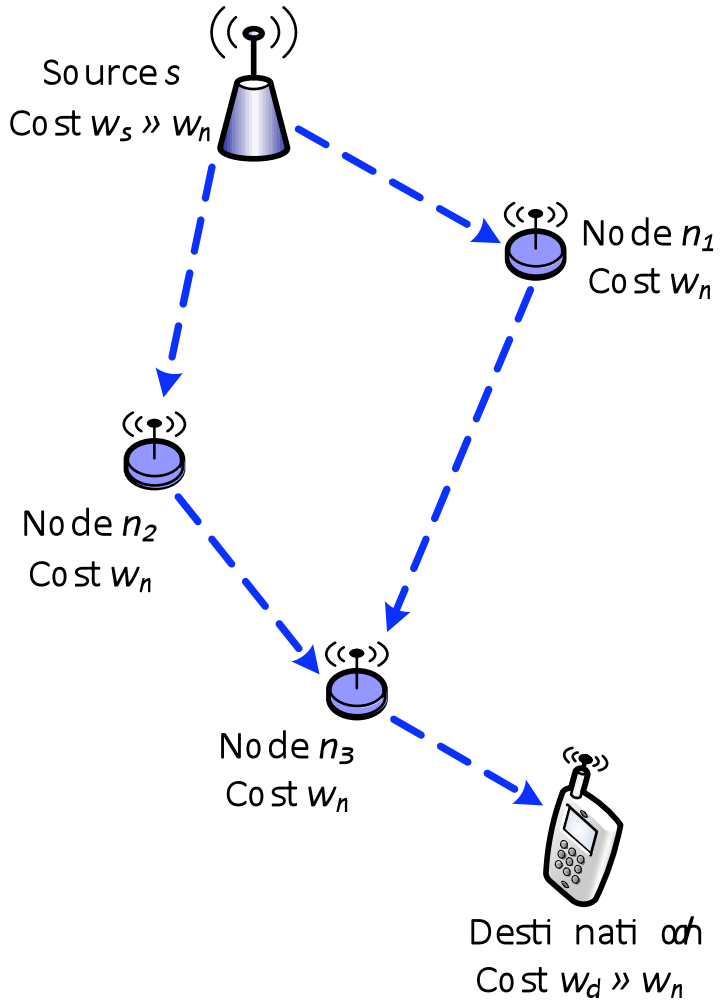
NSL

Work with NRL

- Using the metric to understand the attack progress
- Downloadable at NSL
  www.ee.washington.edu/research/nsl/

Adversary Model

- Adversary
  - Capture/ compromise nodes in set C
  - Recover keys K_C from memory
  - Resource cost w_i to capture node i



NSL

Optimized Node Capture Attacks

Network Element	Condition for Compromise
Link (i,j)	K_{ij} contained in K_C
Path π	At least one link (i,j) in π compromised
Route R_{sd}	All paths π in R_{sd}, and potential end-to-end link (s,d) compromised

Optimal attack: capture the set of nodes C that compromises the target routes with minimum total cost

Nonlinear integer programming problem

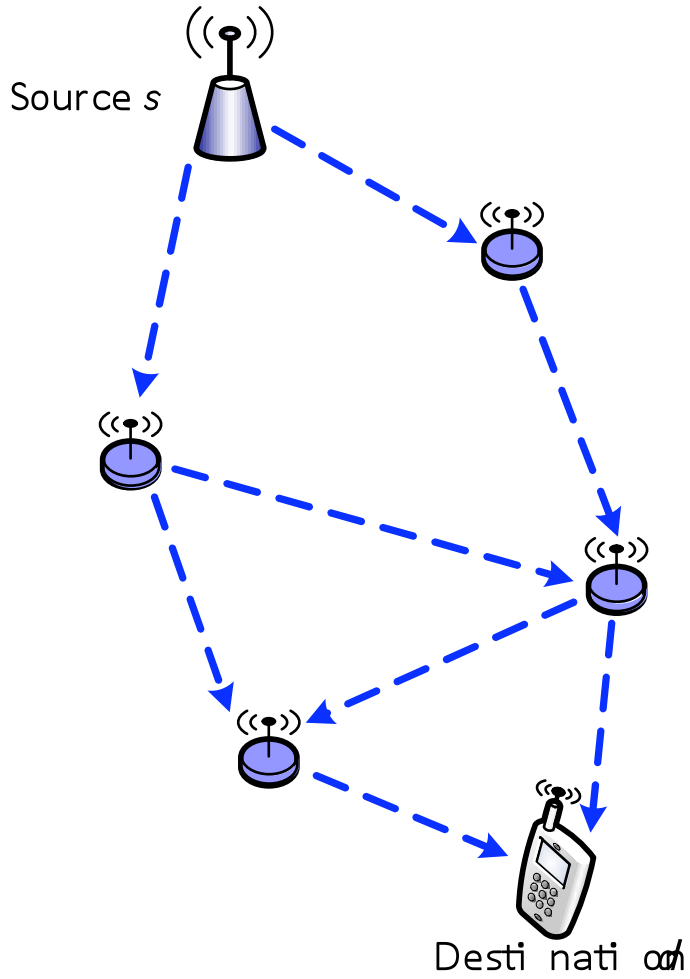
NSL

Heuristic Node Capture Attacks

Heuristic: iteratively capture the single node *n* yielding the *maximum incremental attack progress per unit cost*

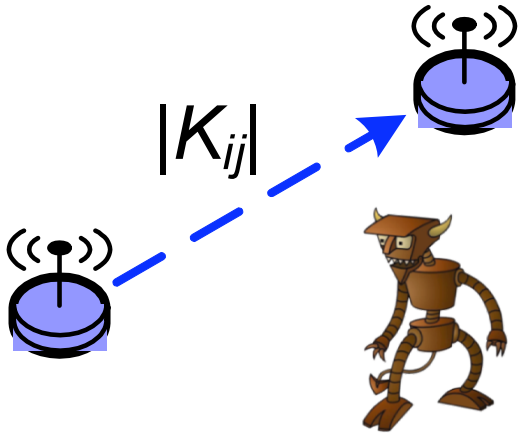
Multiple-Path Routing or Network Coding?

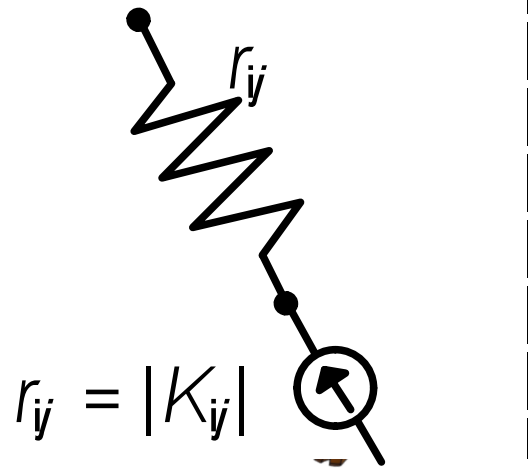
- Without coding
  - Decompose flow into multiple single-path flows
- With Coding
  - Graph composition required to evaluate data security



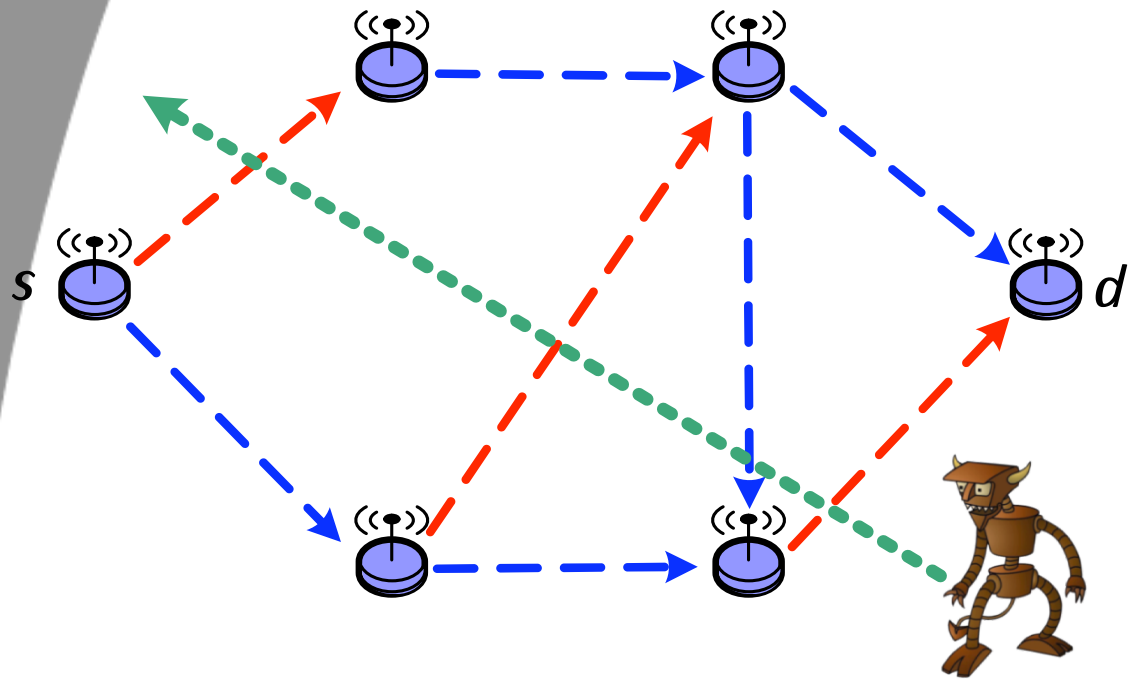
Second Approach: Graph Composition

- Mapping to circuit theory problem
  - Links in series **degrade** security
  - Links in parallel **improve** security
  - Measure the ***overall data security*** in terms of ***effective resistance*** to electric current





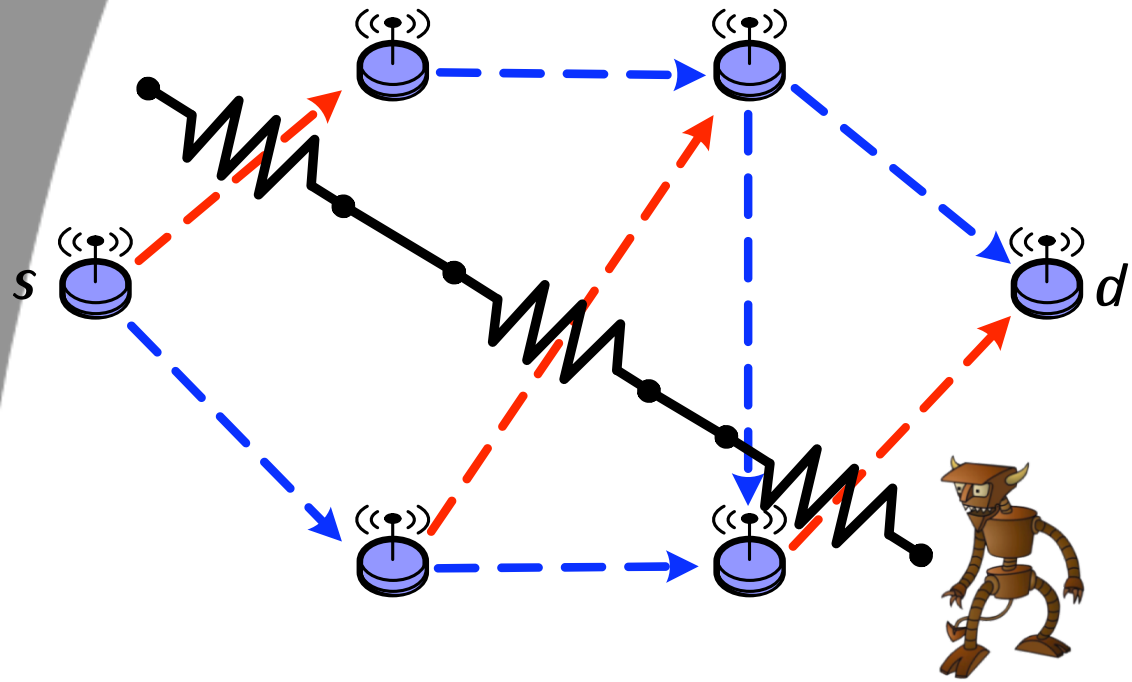
Graph Composition: Step 1



Claim: All traffic in a route is compromised if and only if an **edge cut** of links is compromised.

Graph Composition – Step 1: Map the routing topology to a collection of edge cuts (noting forward- vs. reverse-flow edges).

Graph Composition: Step 2

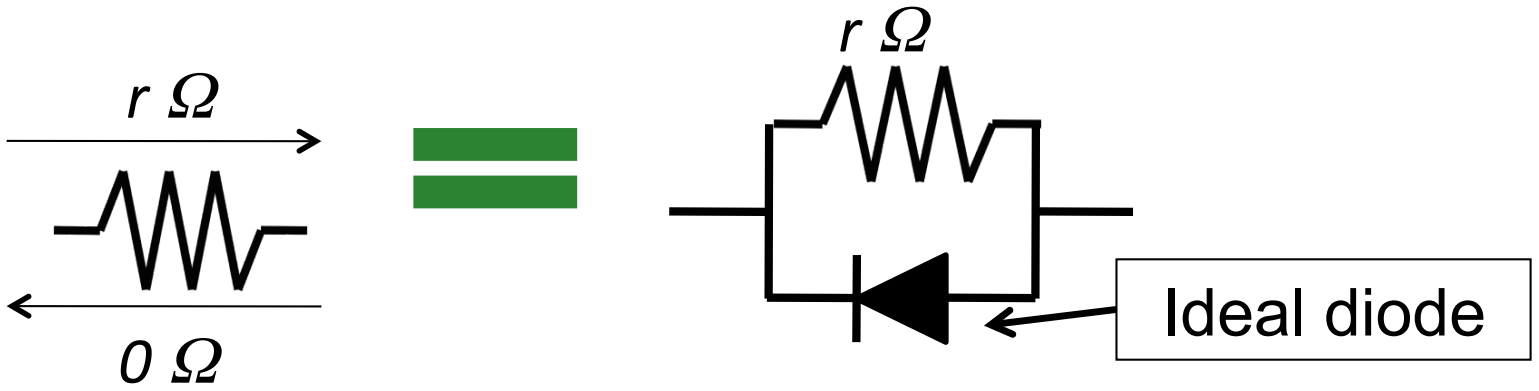


Circuit Analogy: Measure resilience to attack as resistance to electric current.

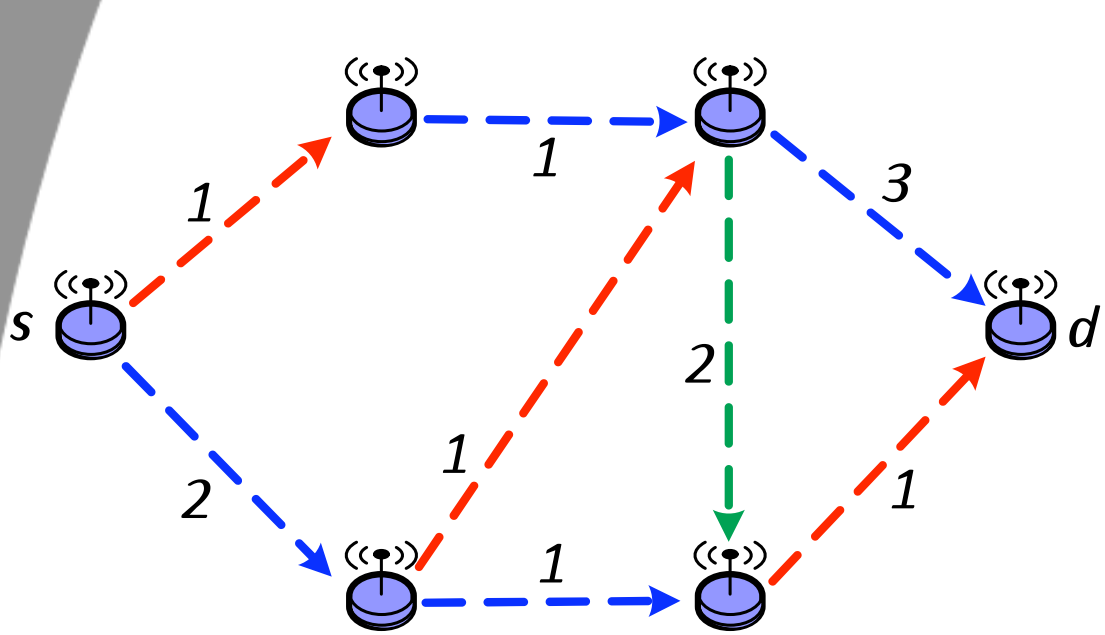
Graph Composition – Step 2: Map each edge cut to a resistive current path, ignoring reverse-flow edges.

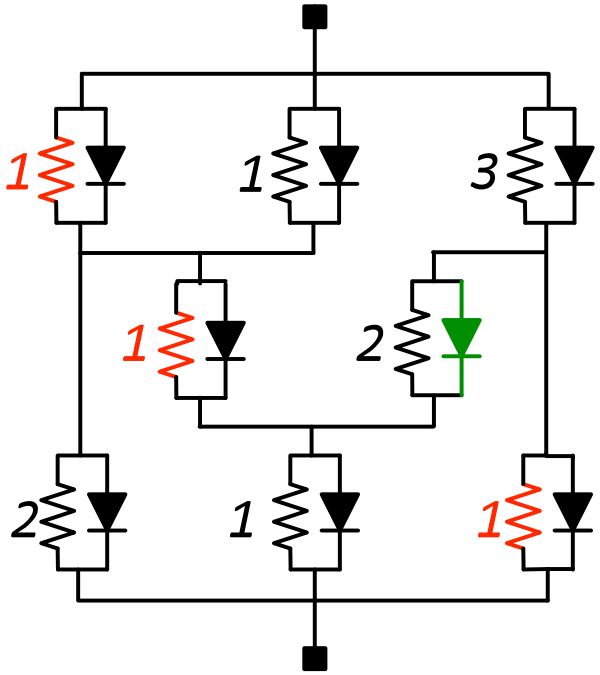
Graph Composition: Step 3

- Combine circuit elements using superposition
  - Must compensate for reverse-flow edges, i.e. constraint due to edge directionality
  - Solution: **Directed resistors**



Graph Composition: Step 3





Graph Composition – Step 3: Combine directed resistor paths into an electric circuit *E* using superposition. Effective resistance of *E* yields overall route security.

NSL

Comments on Composition

- Given composition assumes a planar graph
  - Effectively uses a graph duality property
  - Alternate construction for non-planar graphs, using circuit duality instead of graph duality
    - *Resistance to attack* becomes *conductance of secure data flow*

- Efficiency
  - Edge cut decomposition is not explicitly required, only a dual graph/circuit construction

Heuristic Node Capture Attack Algorithm

- GNAVE Algorithm
  - **G**reedy **N**ode capture **A**pproximation using **V**ulnerability **E**valuation

At each iteration, given node set C already captured/ chosen, add n' given by:

$$n' = \arg\max_n \sum_{(s,d)} v_{sd}\left(h_{C+\{n\}}(s,d) - h_C(s,d)\right) / w_n$$

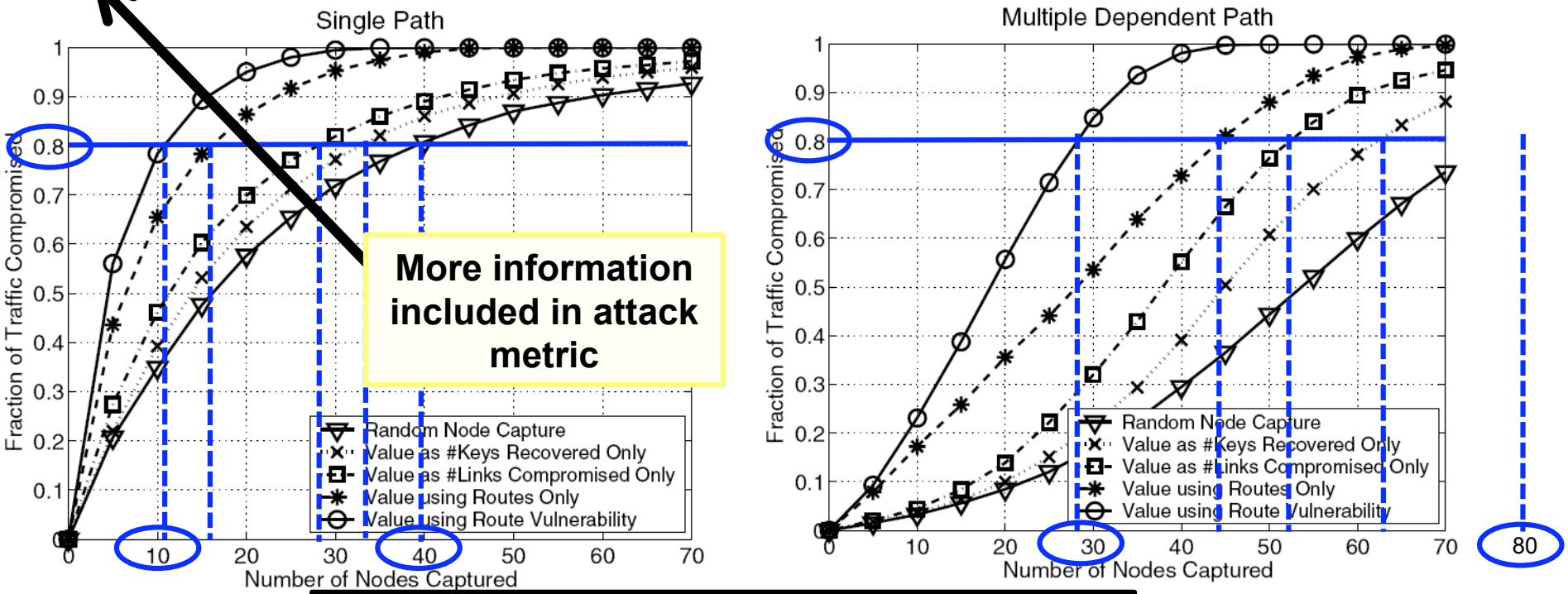
Weight of preference for route data

Incremental value of node *n* toward attack goal

Cost to capture node *n*

Simulation Results

- Evaluate data compromise for 5 attacks
  - Compare single path routing to multiple-path routing with coding in network of 500 nodes



¼ - ⅓ resources needed for same 80% attack impact

NSL

Ongoing work on Network Vulnerability Metric

- Quantifies network resilience to failures and/or attacks
- Uses node and link level information to produce network level value
- Computes the impact of a node or link removal on the overall network
- Classify networks based on effective resistance

NSL

Reading List

1. Patrick Tague, David Slater, Jason Rogers, and Radha Poovendran, Vulnerability of Network Traffic under Node Capture Attacks using Circuit Theoretic Analysis, ***INFOCOM* 2008.**

2. Patrick Tague and Radha Poovendran, Modeling Node Capture Attacks in Wireless Sensor Networks, ***Allerton* 2008**.

3. Patrick Tague and Radha Poovendran, Modeling Adaptive Node Capture Attacks in Multi-hop Wireless Networks, ***Ad Hoc Networks***, Aug 2007.

4. Patrick Tague, David Slater, Jason Rogers, and Radha Poovendran, Evaluating the Vulnerability of Network Traffic Using Joint Security and Routing Analysis, ***IEEE Transactions on Dependable and Secure Computing***, Apr-Jun 2009, **featured article**, http://www.computer.org/tdsc/.

NSL

Acknowledgments

- Professor Wade Trappe, Rutgers University
- NSL Students: Patrick Tague, Andrew Clark, Basel Aloimar
- www.ee.washington.edu/research/nsl
- Also the software tools from NSL